JUNIPER NETWORKS | Engineering Simplicity

Junos® OS

MPLS Applications User Guide

Published
2025-12-16

JUNOS

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Juniper Networks, the Juniper Networks logo, Juniper, and Junos are registered trademarks of Juniper Networks, Inc. in the United States and other countries. All other trademarks, service marks, registered marks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Junos® OS MPLS Applications User Guide
Copyright © 2025 Juniper Networks, Inc. All rights reserved.

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at https://support.juniper.net/support/eula/. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

About This Guide | xxvii

1 **Overview**

Understanding MPLS | 2

MPLS Overview | 2

- MPLS Overview | 2
- TTL Processing on Incoming MPLS Packets | 7
- Link-Layer Support in MPLS | 10
- MPLS Overview for ACX Series Universal Metro Routers | 10
- MPLS for EX Series Switches Overview | 11
- MPLS Feature Support on QFX Series and EX4600 Switches | 13
- MPLS Limitations on QFX Series and EX4600 Switches | 30

Supported Standards | 36

Supported MPLS Standards | 36

- Supported MPLS Standards | 36
- Supported RSVP Standards | 39
- Supported LDP Standards | 41
- DiffServ-Aware Traffic Engineering Standards | 42
- Supported GMPLS Standards | 43
- Supported PCEP Standards | 44

2 **MPLS Configuration**

Configuring MPLS | 46

Basic MPLS Configuration | 46

- MPLS Configuration Overview | 46
- MPLS Configuration Guidelines | 47
- Configuring MPLS | 48
- Example: Enabling MPLS | 48
  - Requirements | 49
  - Overview | 49
  - Configuration | 49
  - Verification | 51

Example: Configuring MPLS on EX8200 and EX4500 Switches | 52

Requirements | 53

Overview and Topology | 53

Configuring the Local PE Switch | 59

Configuring the Remote PE Switch | 63

Configuring the Provider Switch | 67

Verification | 71

MPLS on Provider and Provider Edge Devices Configuration | 76

Configuring MPLS on Provider Switches | 76

Configuring MPLS on Provider Edge Switches | 78

Configuring the Ingress PE Switch | 78

Configuring the Egress PE Switch | 80

Configuring MPLS on Provider Edge Switches Using IP-Over-MPLS | 83

Configuring the Ingress PE Switch | 83

Configuring the Egress PE Switch | 86

Configuring MPLS on Provider Edge EX8200 and EX4500 Switches Using Circuit Cross-Connect | 89

Configuring MPLS on EX8200 and EX4500 Provider Switches | 93

MPLS Configuration on IRB Interfaces | 95

MPLS Support on IRB Interfaces | 95

Configure MPLS on IRB Interfaces | 97

Configuring MPLS Tunnels | 99

IPv6-over-Ipv4 Tunnels | 99

Configuring IPv6 Tunneling for MPLS | 99

Example: Tunneling IPv6 Traffic over MPLS IPv4 Networks | 101

Requirements | 101

Overview | 101

Configuration | 104

Verification | 112

Next-Hop-Based Dynamic Tunnels | 113

Example: Configuring Next-Hop-Based MPLS-Over-UDP Dynamic Tunnels | 114

Requirements | 114

Overview | 115

Configuration | 119

Verification | **126**

Troubleshooting | **131**

Anti-Spoofing Protection for Next-Hop-Based Dynamic Tunnels Overview | **132**

Example: Configuring Anti-Spoofing Protection for Next-Hop-Based Dynamic Tunnels | **135**

Requirements | **135**

Overview | **136**

Configuration | **138**

Verification | **146**

Next-Hop-Based Dynamic Tunnel Localization Overview | **149**

Overview of Next-Hop-Based Dynamic Tunneling Using IP-Over-IP Encapsulation | **155**

Example: Configuring Next-Hop-Based IP-Over-IP Dynamic Tunnels | **157**

Requirements | **157**

Overview | **158**

Configuring IP-over-IP Dynamic Tunnels with a Protocol Next Hop | **159**

Example: Configure an IPoIP Tunnel in an MPLS Environment with LDP tunnel, Resolved Through inetcolor.0 Using Static Configuration | **172**

Example: Configure an IPoIP Tunnel with LDP tunnel in an MPLS Cloud, Resolved through inetcolor.0 Using BGP Signaling | **188**

Verification | **199**

3 **MPLS Traffic**

Managing MPLS Traffic | 205

Bidirectional Forwarding Detection (BFD) for MPLS | **205**

Configuring Bidirectional Forwarding Detection for MPLS (CLI Procedure) | **205**

Configuring BFD on Provider Edge and Provider Switches for an LDP-Based LSP | **206**

Configuring BFD on Provider Edge and Provider Switches for an RSVP-Based LSP | **209**

BFD-Triggered Local Repair for Rapid Convergence | **210**

Understanding BFD-Triggered Local Protection | **211**

Configuring BFD for MPLS IPv4 LSPs | **213**

Platform-Specific BFD for MPLS Behavior | **217**

Firewall Filters for MPLS | **218**

Configuring MPLS Firewall Filters and Policers on Routers | **218**

Overview of MPLS Firewall Filters on Loopback Interface | **227**

Configuring MPLS Firewall Filters and Policers on Switches | **229**

Configuring an MPLS Firewall Filter | **230**

Applying an MPLS Firewall Filter to an MPLS Interface | **230**

Applying an MPLS Firewall Filter to a Loopback Interface | **231**

Configuring Policers for LSPs | **231**

System Log Messages and SNMP Traps for MPLS | **232**

Load Balancing MPLS Traffic | **234**

Configuring Load Balancing Based on MPLS Labels | **235**

Example: Load-Balanced MPLS Network | **241**

Router Configurations for the Load-Balanced MPLS Network | **242**

Configuring Load Balancing Based on MPLS Labels on ACX Series Routers | **257**

MPLS Encapsulated Payload Load-balancing Overview | **261**

Configuring MPLS Encapsulated Payload for Load Balancing | **262**

Policy-Based Multipath Routes Overview | **263**

Understanding IP-Based Filtering and Selective Port Mirroring of MPLS Traffic | **269**

Shared Risk Link Groups for MPLS | **275**

SRLG Overview | **276**

Example: Configuring SRLG | **276**

Requirements | **277**

Overview | **277**

Configuration | **279**

Verification | **286**

Example: Excluding SRLG Links Completely for the Secondary LSP | **289**

Requirements | **290**

Overview | **290**

Configuration | **291**

Verification | **296**

Example: Configuring SRLG with Link Protection | **298**

Requirements | **298**

Overview | **298**

Configuration | **299**

Verification | **327**

Example: Configuring SRLG with Link Protection with the exclude-srlg Option | **329**

Requirements | **329**

Overview | **329**

Configuration | **330**

Verification | **358**

Protecting MPLS Traffic | 360

Node and Path Protection for MPLS LSPs | **360**

- MPLS and Traffic Protection | **360**
- Node-Link Protection Overview | **361**
- Path Protection Overview | **363**
- Configuring Path Protection in an MPLS Network (CLI Procedure) | **364**
  - Configuring the Primary Path | **365**
  - Configuring the Secondary Path | **366**
  - Configuring the Revert Timer | **367**
- Preventing Use of a Path That Previously Failed | **367**
- Configuring MPLS Inter-AS Link-Node Protection with Labeled BGP | **368**
  - Understanding MPLS Inter-AS Link Protection | **368**
  - Example: Configuring MPLS Inter-AS Link-Node Protection | **370**
- Configuring Egress Protection Service Mirroring for BGP Signaled Layer 2 Services | **389**
- Example: Configuring MPLS Egress Protection Service Mirroring for BGP Signaled Layer 2 Services | **394**
  - Requirements | **395**
  - Overview | **395**
  - Configuration | **397**
  - Verification | **414**
- Example: Configuring Layer 3 VPN Egress Protection with PLR as Protector | **419**
  - Requirements | **419**
  - Overview | **419**
  - Configuration | **420**
  - Verification | **443**
- Understanding MPLS and Path Protection on EX Series Switches | **453**
- Verifying Path Protection in an MPLS Network | **454**
  - Verifying the Primary Path | **454**
  - Verifying the RSVP-Enabled Interfaces | **456**
  - Verifying a Secondary Path | **456**

Link Protection for MPLS LSPs | **459**

- Link Protection | **459**
- Multiple Bypass LSPs for Link Protection | **460**
- Node Protection | **461**
- Bandwidth Protection for Links and Nodes (Dynamic and Static) | **462**

Fast Reroute, Node Protection, and Link Protection | **463**

Configuring Link Protection on Interfaces Used by LSPs | **468**

Configuring Node Protection or Link Protection for LSPs | **478**

Configuring Inter-AS Node and Link Protection | **478**

Configuring Constraint Aware Bypass LSPs | **479**

Measuring MPLS Traffic | 484

Gather Statistics on MPLS Sessions | **484**

Configuring MPLS to Gather Statistics | **484**

On-Demand Packet Loss and Delay Measurement for UHP LSPs Overview | **486**

Example: Configuring On-Demand Loss and Delay Measurement | **493**

Requirements | **493**

Overview | **494**

Configuration | **495**

Verification | **500**

Example: Configuring Pro-active Loss and Delay Measurements for Bidirectional MPLS LSPs | **506**

Requirements | **506**

Overview | **507**

Configuration | **508**

Verification | **514**

Configuring On-Demand Loss and Delay Measurement | **516**

Configuring Pro-Active Loss and Delay Measurements | **517**

4 **MPLS LSPs**

Understanding MPLS LSPs | 521

LSP Overview | **521**

How a Packet Travels Along an LSP | **521**

Types of LSPs | **522**

Scope of LSPs | **523**

LSP Labels | **523**

MPLS Label Overview | **524**

MPLS Label Allocation | **524**

Operations on MPLS Labels | **525**

Understanding MPLS Label Operations | **526**

Understanding MPLS Label Manager | **529**

Special MPLS Labels | **530**

Entropy Label Support in Mixed Mode Overview | 531

Abstract Hops for MPLS LSPs Overview | 531

Example: Configuring Abstract Hops for MPLS LSPs | 545

Requirements | 545

Overview | 546

Configuration | 548

Verification | 565

Configuring the Maximum Number of MPLS Labels | 567

Configuring MPLS to Pop the Label on the Ultimate-Hop Router | 569

Advertising Explicit Null Labels to BGP Peers | 570

Understanding MPLS Label Operations on EX Series Switches | 571

LSP Routes | 575

MPLS and Routing Tables | 575

Fast Reroute Overview | 577

Configuring Fast Reroute | 580

Detour Merging Process | 581

Detour Computations | 582

Fast Reroute Path Optimization | 582

Configuring the Optimization Interval for Fast Reroute Paths | 583

Adding LSP-Related Routes to the inet.3 or inet6.3 Routing Table | 583

LSP Computation | 585

Constrained-Path LSP Computation | 585

How CSPF Selects a Path | 586

CSPF Path Selection Tie-Breaking | 587

Computing CSPF Paths Offline | 588

Configuring CSPF Tie Breaking | 588

Disabling Constrained-Path LSP Computation | 589

LSP Routers | 590

Routers in an LSP | 591

Configuring the Ingress and Egress Router Addresses for LSPs | 591

Configuring the Ingress Router for MPLS-Signaled LSPs | 594

Configuring the Intermediate and Egress Routers for MPLS-Signaled LSPs | 599

Configuring the Connection Between Ingress and Egress Routers | 599

Pinging LSPs | 600

Configuring MPLS LSPs | 603

Basic LSP Configuration | **603**

- Configuring RSVP-signaled MPLS Label-Switched Path (LSP) Metrics | **604**
- Configuring a Text Description for RSVP-signaled MPLS Label-Switched Paths (LSP) and Static LSPs | **609**
- Configuring RSVP-signaled MPLS Label-Switched Path Soft Preemption | **611**
- Configuring Priority and Preemption for RSVP-signaled MPLS Label-Switched Paths | **612**
- Configuring Administrative Groups for LSPs | **613**
- Configuring Extended Administrative Groups for LSPs | **616**
- Configuring the Junos Routing Preference Value for RSVP-signaled MPLS LSPs | **618**
- Disabling Path Route Recording by RSVP-signaled MPLS LSPs | **618**
- Achieving a Make-Before-Break, Hitless Switchover for RSVP-signaled MPLS LSPs | **618**
  - Specifying the Amount of Time the Router Waits to Switch Over to New Paths | **620**
  - Specifying the Amount of Time to Delay the Tear Down of Old Paths | **620**
  - Achieving a Hitless, MBB Switchover Without Artificial Delays | **621**
- Optimizing Signaled LSPs | **622**
- Configuring the Smart Optimize Timer for LSPs | **625**
- Limiting the Number of Hops in the Path of an RSVP-signaled MPLS LSP | **627**
- Configuring a Manually-Defined Bandwidth Value for RSVP-signaled MPLS LSPs | **627**
- Automatic Bandwidth Allocation for LSPs | **627**
- Configuring Automatic Bandwidth Allocation for LSPs | **628**
- Configuring Optimized Auto-bandwidth Adjustments for MPLS LSPs | **629**
- Configuring Reporting of Automatic Bandwidth Allocation Statistics for LSPs | **633**
- Using Polling Profiles for Automatic Bandwidth Allocation for RSVP LSPs | **637**
- Configuring an RSVP-signaled MPLS LSP to Calculate a Path Across Autonomous Systems | **639**
- Damping IGP Advertisements of RSVP-signaled MPLS LSP State Changes | **641**
- Configuring Corouted Bidirectional RSVP-signaled MPLS LSPs | **641**
- Configuring the Entropy Label for LSPs | **644**
- Example: Configuring an Entropy Label for a BGP Labeled Unicast LSP | **646**
  - Requirements | **646**
  - Overview | **647**
  - Configuration | **648**
  - Verification | **663**
- Configuring Ultimate-Hop Popping for RSVP-signaled MPLS LSPs | **671**
- Configuring Explicit-Path LSPs | **675**
- Example: Configuring an Explicit-Path LSP | **676**

LSP Bandwidth Oversubscription Overview | **677**

LSP Size Oversubscription | **678**

LSP Link Size Oversubscription | **678**

Class Type Oversubscription and Local Oversubscription Multipliers | **679**

Configuring the Bandwidth Subscription Percentage for LSPs | **679**

Detecting MPLS MTU Exceed Errors | **681**

Primary, Secondary, and Static LSP Configuration | **683**

Configuring Primary and Secondary LSPs | **683**

Configuring Hot Standby of Secondary Paths for LSPs | **689**

Configuring Static LSPs | **690**

Configuring Static Label Switched Paths for MPLS (CLI Procedure) | **703**

Configuring the Ingress PE Switch | **703**

Configuring the Provider and the Egress PE Switch | **705**

Configuring Static Label Switched Paths for MPLS | **706**

Configuring the Ingress PE Switch | **707**

Configuring the Provider and the Egress PE Switch | **708**

Adaptive LSP Configuration | **709**

Container LSP Configuration | **710**

Dynamic Bandwidth Management Using Container LSP Overview | **710**

Example: Configuring Dynamic Bandwidth Management Using Container LSP | **742**

Requirements | **742**

Overview | **743**

Configuration | **744**

Verification | **756**

Configuring Dynamic Bandwidth Management Using Container LSP | **774**

Multiclass LSP Configuration | **778**

Multiclass LSP Overview | **778**

Multiclass LSPs | **779**

Establishing a Multiclass LSP on the Differentiated Services Domain | **779**

Point-to-Multipoint LSP Configuration | **780**

Point-to-Multipoint LSPs Overview | **781**

Understanding Point-to-Multipoint LSPs | **782**

Point-to-Multipoint LSP Configuration Overview | **784**

Example: Configuring a Collection of Paths to Create an RSVP-Signaled Point-to-Multipoint LSP | **784**

Requirements | **784**

Overview | **784**

Configuration | **786**

Verification | **809**

Configuring Primary and Branch LSPs for Point-to-Multipoint LSPs | **811**

Configuring Inter-Domain Point-to-Multipoint LSPs | **813**

Configuring Link Protection for Point-to-Multipoint LSPs | **815**

Configuring Graceful Restart for Point-to-Multipoint LSPs | **815**

Configuring a Multicast RPF Check Policy for Point-to-Multipoint LSPs | **816**

Configuring Ingress PE Router Redundancy for Point-to-Multipoint LSPs | **817**

Configuring a Service to Correlate Point-to-Multipoint sub-LSPs with FPCs | **818**

Enabling Point-to-Point LSPs to Monitor Egress PE Routers | **822**

Preserving Point-to-Multipoint LSP Functioning with Different Junos OS Releases | **823**

Re-merge Behavior on Point-to-Multipoint LSP Overview | **823**

Pop-and-Forward LSP Configuration | **826**

Segment Routing LSP Configuration | **833**

Enabling Distributed CSPF for Segment Routing LSPs | **834**

Static Segment Routing Label Switched Path | **841**

Static Segment Routing LSP in MPLS Networks | **841**

Example: Configuring Static Segment Routing Label Switched Path | **860**

Routing Engine-based S-BFD for Segment-Routing Traffic Engineering with First-Hop Label Resolution | **880**

Understanding RE-based S-BFD for Segment-Routing Traffic Engineering with First-Hop Label Resolution | **881**

S-BFD for SRv6 TE Paths | **882**

Configuring RE-based S-BFD for Segment-Routing Traffic Engineering with First-Hop Label Resolution | **883**

Auto Derivation of Remote Discriminator (RD) for S-BFD Session | **886**

Example: Configuring Seamless Bidirectional Forwarding Detection (S-BFD) for SR-TE | **887**

Configuring Static Adjacency Segment Identifier for Aggregate Ethernet Member Links Using Single-Hop Static LSP | **911**

Non-Router-ID Endpoints in Segment Routing Traffic Engineering | **913**

Computing Delay Optimized Intradomain and Interdomain Segment Routing Paths | **914**

Understanding SRv6 TE Tunnel Local Path Computation | **919**

Configuring SRv6 TE Tunnel Local Path Computation | **920**

Understand BGP VPN Option C MPLS over SRv6 | **921**

Platform-Specific Segment Routing LSP Behavior | **925**

Express Segment LSP Configuration | **926**

Establish End-to-End Segment Routing Path Using Express Segments | **926**

Example: Inter-domain SR-TE Connectivity Using Express Segments Through RSVP-TE Underlay | **935**

Requirements | **935**

Overview | **936**

Configuration | **940**

Verification | **1031**

Example: Inter-domain SR-TE Connectivity Using Express Segments Through SR-TE Underlay | **1048**

Requirements | **1048**

Overview | **1049**

Configuration | **1053**

Verification | **1199**

5 **MPLS Signalling Protocols**

RSVP | 1224

RSVP Overview | **1224**

RSVP Overview | **1225**

RSVP Operation Overview | **1225**

Understanding the RSVP Signaling Protocol | **1226**

RSVP-TE protocol extensions for FRR | **1229**

Junos OS RSVP Protocol Implementation | **1231**

RSVP Authentication | **1232**

RSVP and IGP Hello Packets and Timers | **1232**

RSVP Message Types | **1233**

Understanding RSVP Automatic Mesh | **1235**

RSVP Reservation Styles | **1236**

RSVP Refresh Reduction | **1237**

MTU Signaling in RSVP | **1237**

How the Correct MTU Is Signaled in RSVP | **1238**

Determining an Outgoing MTU Value | **1239**

MTU Signaling in RSVP Limitations | **1239**

RSVP Configuration | **1240**

Minimum RSVP Configuration | **1242**

Configuring RSVP and MPLS | **1242**

Configuring RSVP Interfaces | **1244**

Configuring RSVP Node-ID Hellos | **1250**

Example: Configuring RSVP-Signaled LSPs | **1251**

Requirements | **1252**

Overview and Topology | **1252**

Configuration | **1253**

Verification | **1255**

Example: Configuring RSVP Automatic Mesh | **1257**

Requirements | **1258**

Overview | **1258**

Configuration | **1259**

Verification | **1261**

Configuring Hello Acknowledgments for Nonsession RSVP Neighbors | **1262**

Switching LSPs Away from a Network Node | **1263**

Configuring RSVP Setup Protection | **1264**

Configuring Load Balancing Across RSVP LSPs | **1265**

Configuring RSVP Automatic Mesh | **1266**

Configuring Timers for RSVP Refresh Messages | **1267**

Preempting RSVP Sessions | **1268**

Configuring MTU Signaling in RSVP | **1269**

Configuring Ultimate-Hop Popping for RSVP-signaled MPLS LSPs | **1270**

Configuring RSVP to Pop the Label on the Ultimate-Hop Router | **1274**

Enabling Ultimate-Hop Popping on Point-to-Multipoint LSPs | **1275**

Tracing RSVP Protocol Traffic | **1276**

RSVP Graceful Restart | **1279**

RSVP Graceful Restart Terminology | **1280**

Restart time (in milliseconds) | **1280**

Recovery time (in milliseconds) | **1280**

RSVP Graceful Restart Operation | **1281**

Processing the Restart Cap Object | **1282**

Configuring RSVP Graceful Restart | **1282**

RSVP LSP Tunnels Overview | **1284**

Example: RSVP LSP Tunnel Configuration | **1286**

Configuring Link Management Protocol Peers | **1311**

Configuring Link Management Protocol Traffic Engineering Links | **1312**

Configuring Peer Interfaces in OSPF and RSVP | **1312**

Defining Label-Switched Paths for the FA-LSP | **1313**

Establishing FA-LSP Path Information | **1313**

Option: Tearing Down RSVP LSPs Gracefully | **1314**

LDP | 1316

LDP Overview | **1316**

LDP Introduction | **1317**

Understanding the LDP Signaling Protocol | **1317**

Example: Configuring LDP-Signaled LSPs | **1318**

Requirements | **1318**

Overview | **1318**

Configuration | **1319**

Junos OS LDP Protocol Implementation | **1322**

LDP Operation | **1322**

LDP Message Types | **1322**

Tunneling LDP LSPs in RSVP LSPs | **1324**

Tunneling LDP LSPs in RSVP LSPs Overview | **1324**

Tunneling LDP over SR-TE | **1325**

Example: Tunneling LDP over SR-TE in IS-IS Network | **1329**

Requirements | **1329**

Overview | **1329**

Configuration | **1330**

Verification | **1351**

Label Operations | **1359**

LDP Session Protection | **1360**

LDP Native IPv6 Support Overview | **1361**

Longest Match Support for LDP Overview | **1362**

LDP Configuration | **1362**

Minimum LDP Configuration | **1364**

Enabling and Disabling LDP | **1364**

Configuring the LDP Timer for Hello Messages | **1365**

Configuring the Delay Before LDP Neighbors Are Considered Down | **1366**

Enabling Strict Targeted Hello Messages for LDP | **1368**

Configuring the Interval for LDP Keepalive Messages | **1369**

Configuring the LDP Keepalive Timeout | **1369**

Configuring Longest Match for LDP | **1369**

Example: Configuring Longest Match for LDP | **1370**

- Requirements | **1370**
- Overview | **1371**
- Configuration | **1372**
- Verification | **1379**

Configuring LDP Route Preferences | **1391**

LDP Graceful Restart | **1391**

Configuring LDP Graceful Restart | **1392**

Filtering Inbound LDP Label Bindings | **1395**

Filtering Outbound LDP Label Bindings | **1398**

Specifying the Transport Address Used by LDP | **1400**

Control Transport Address Used for Targeted-LDP Session | **1401**

Configuring the Prefixes Advertised into LDP from the Routing Table | **1404**

Configuring FEC Deaggregation | **1405**

Configuring Policers for LDP FECs | **1406**

Configuring LDP IPv4 FEC Filtering | **1407**

Configuring BFD for LDP LSPs | **1408**

Configuring ECMP-Aware BFD for LDP LSPs | **1411**

Configuring a Failure Action for the BFD Session on an LDP LSP | **1412**

Configuring the Holddown Interval for the BFD Session | **1413**

Configuring LDP Link Protection | **1413**

Example: Configuring LDP Link Protection | **1415**

- LDP Link Protection Overview | **1415**
- Example: Configuring LDP Link Protection | **1436**

Understanding Multicast-Only Fast Reroute | **1448**

Configuring Multicast-Only Fast Reroute | **1457**

Example: Configuring Multicast-Only Fast Reroute in a Multipoint LDP Domain | **1460**

- Requirements | **1461**
- Overview | **1461**
- CLI Quick Configuration | **1462**
- Configuration | **1471**
- Verification | **1478**

Example: Configuring LDP Downstream on Demand | **1483**

Requirements | **1483**

Overview | **1483**

Configuration | **1484**

Verification | **1488**

Configuring LDP Native IPv6 Support | **1490**

Example: Configuring LDP Native IPv6 Support | **1491**

Requirements | **1491**

Overview | **1492**

Configuration | **1492**

Example: Configuring Multipoint LDP In-Band Signaling for Point-to-Multipoint LSPs | **1510**

Understanding Multipoint LDP Inband Signaling for Point-to-Multipoint LSPs | **1511**

Example: Configuring Multipoint LDP In-Band Signaling for Point-to-Multipoint LSPs | **1522**

Mapping Client and Server for Segment Routing to LDP Interoperability | **1549**

Miscellaneous LDP Properties | **1555**

Configuring LDP LSP Traceroute | **1563**

Collecting LDP Statistics | **1564**

Tracing LDP Protocol Traffic | **1567**

Example: Configuring Multiple-Instance LDP | **1571**

Tunneling LDP over SR-TE | **1598**

Example: Tunneling LDP over SR-TE in IS-IS Network | **1602**

Requirements | **1602**

Overview | **1602**

Configuration | **1603**

Verification | **1624**

Example: Tunneling LDP over SR-TE in OSPF Network | **1632**

Overview | **1633**

Requirements | **1633**

Configuration | **1634**

Verification | **1652**

MPLS TTL Propagation Flexibility for LDP-signaled LSPs | **1661**

Example: Configuring MPLS TTL Propagation for LDP-signaled LSPs | **1663**

Overview | **1663**

Topology | **1663**

Purpose | **1663**

Use Case for the No Propagate TTL and the Propagate TTL behavior at LDP | **1664**

Configuration | **1665**

Understanding Multipoint LDP Recursive FEC | **1668**

Example: Configuring Multipoint LDP Recursive FEC | **1671**

Example Prerequisites | **1671**

Before You Begin | **1672**

Functional Overview | **1672**

Topology Overview | **1672**

Topology Illustration | **1674**

PE1 Configuration Steps | **1674**

Verification | **1678**

Appendix 1: Set Commands on All Devices | **1683**

Appendix 2: Show Configuration Output on PE1 and ASBR2 | **1692**

6 **MPLS Traffic Engineering**

Configuring MPLS Traffic Engineering | 1701

MPLS Traffic Engineering Configuration | **1701**

MPLS and Traffic Engineering | **1702**

MPLS Traffic Engineering and Signaling Protocols Overview | **1702**

Traffic Engineering Capabilities | **1703**

Components of Traffic Engineering | **1703**

Configuring Traffic Engineering for LSPs | **1704**

Enabling Interarea Traffic Engineering | **1708**

Enabling Inter-AS Traffic Engineering for LSPs | **1709**

Packet Forwarding Component | **1712**

Offline Path Planning and Analysis | **1715**

Flexible LSP Calculation and Configuration | **1715**

Link-State Distribution Using BGP Overview | **1716**

Example: Configuring Link State Distribution Using BGP | **1731**

Requirements | **1732**

Overview | **1732**

Configuration | **1733**

Verification | **1749**

Configuring Link State Distribution Using BGP | **1756**

BGP Classful Transport Planes Overview | **1760**

Improving Traffic Engineering Database Accuracy with RSVP PathErr Messages | **1769**

Color-Based Traffic Engineering Configuration | **1772**

BGP Classful Transport Planes Overview | **1772**

Example: Configuring Classful Transport Planes (Intra-Domain) | **1781**

Before You Begin | **1781**

Functional Overview | **1782**

Topology Overview | **1784**

Topology Illustrations | **1786**

PE1 Configuration Steps | **1786**

Verify Classful Transport Planes | **1790**

Appendix 1: Troubleshooting | **1800**

Appendix 2: Set Commands on All Devices | **1809**

Appendix 3: Show Configuration Output on PE1 | **1815**

Color-Based Mapping of VPN Services Overview | **1819**

DiffServ-Aware Traffic Engineering Configuration | **1826**

DiffServ-Aware Traffic Engineering Introduction | **1827**

DiffServ-Aware Traffic Engineering Terminology | **1827**

Bandwidth model | **1828**

CAC | **1828**

Class type | **1828**

Differentiated Services | **1828**

Differentiated Services domain | **1828**

DiffServ-aware traffic engineering | **1828**

Multiclass LSP | **1828**

MAM | **1829**

RDM | **1829**

Traffic engineering class | **1829**

Traffic engineering class map | **1829**

DiffServ-Aware Traffic Engineering Features | **1829**

Configuring Link Down Notification for Optics Options Alarm or Warning | **1830**

DiffServ-Aware Traffic Engineered LSPs Overview | **1830**

DiffServ-Aware Traffic Engineered LSPs Operation | **1831**

Configuring Routers for DiffServ-Aware Traffic Engineering | **1832**

Configuring LSPs for DiffServ-Aware Traffic Engineering | **1837**

7 **MPLS Transport Profile**

Operation, Administration, and Maintenance (OAM) for MPLS | 1842

MPLS OAM Configuration | **1842**

Configuring the MPLS Transport Profile for OAM | **1842**

MPLS Transport Profile Overview | **1842**

Example: Configuring the MPLS Transport Profile for OAM | **1843**

Configuring OAM Ingress Policies for LDP | **1860**

Tracing MPLS and LSP Packets and Operations | **1861**

MPLS Pseudowires | 1863

MPLS Pseudowires Configuration | **1863**

Ethernet Pseudowire Overview | **1863**

Example: Ethernet Pseudowire Base Configuration | **1864**

Requirements | **1864**

Overview of an Ethernet Pseudowire Base Configuration | **1865**

Configuring an Ethernet Pseudowire | **1865**

Pseudowire Overview for ACX Series Universal Metro Routers | **1868**

Understanding Multisegment Pseudowire for FEC 129 | **1869**

Example: Configuring a Multisegment Pseudowire | **1875**

Requirements | **1875**

Overview | **1876**

Configuration | **1882**

Verification | **1910**

Troubleshooting | **1921**

MPLS Stitching For Virtual Machine Connection | **1924**

TDM Pseudowires Overview | **1926**

Example: TDM Pseudowire Base Configuration | **1926**

Requirements | **1927**

Overview of a TDM Pseudowire Base Configuration | **1927**

Configuring an TDM Pseudowire | **1927**

Configuring Load Balancing for Ethernet Pseudowires | **1931**

Configuring Load Balancing Based on MAC Addresses | **1933**

Pseudowire Headend Termination (PWHT) Configuration | **1934**

PWHT Overview | **1935**

PWHT RLT Configuration Modes | **1936**

Configuring PWHT Active-Backup Mode | **1937**

Configuring PWHT Active-Active Mode without Targeting | **1940**

Configuring PWHT Active-Active Mode with Targeting | **1941**

Pseudowire over Aggregated Ethernet Overview | **1946**

Class-of-Service (CoS) for MPLS | 1950

MPLS Class-of-Service Configuration | **1950**

Configure CoS for MPLS LSPs | **1951**

Configuring MPLS Rewrite Rules | **1954**

Configuring CoS Bits for an MPLS Network | **1956**

Configuring CoS on an MPLS Provider Edge Switch Using IP Over MPLS | **1957**

Configuring CoS | **1957**

Configuring an LSP Policer | **1958**

Configuring CoS on an MPLS Provider Edge Switch Using Circuit Cross-Connect | **1959**

Configuring CoS | **1959**

Configuring an LSP Policer | **1960**

Configuring CoS on Provider Switches of an MPLS Network | **1961**

Understanding Using CoS with MPLS Networks on EX Series Switches | **1962**

Example: Combining CoS with MPLS on EX Series Switches | **1966**

Requirements | **1967**

Overview and Topology | **1967**

Configuring the Local PE Switch | **1971**

Configuring the Remote PE Switch | **1974**

Configuring the Provider Switch | **1976**

Verification | **1978**

Understanding CoS MPLS EXP Classifiers and Rewrite Rules | **1984**

Configuring Rewrite Rules for MPLS EXP Classifiers | **1988**

Configuring CoS Bits for an MPLS Network | **1990**

Configuring a Global MPLS EXP Classifier | **1990**

Generalized MPLS (GMPLS) | 1992

GMPLS Configuration | **1992**

Introduction to GMPLS | **1992**

GMPLS Terms and Acronyms | **1994**

Generalized MPLS (GMPLS) | **1994**

Forwarding adjacency | **1994**
GMPLS label | **1994**
GMPLS LSP types | **1994**
Link Management Protocol | **1995**
Traffic engineering link | **1995**
GMPLS Operation | **1995**
GMPLS and OSPF | **1995**
GMPLS and CSPF | **1996**
GMPLS Features | **1996**
Configuring MPLS Paths for GMPLS | **1997**
Tracing LMP Traffic | **1997**
Configuring MPLS LSPs for GMPLS | **1998**
Gracefully Tearing Down GMPLS LSPs | **2002**
GMPLS RSVP-TE VLAN LSP Signaling Overview | **2004**
Example: Configuring GMPLS RSVP-TE VLAN LSP Signaling | **2011**
Requirements | **2011**
Overview | **2012**
Configuration | **2018**
Verification | **2034**

8 **MPLS VPNs and Circuits**

Ethernet over MPLS (L2 Circuit) | 2042

Understanding Ethernet-over-MPLS (L2 Circuit) | **2042**

Configuring Ethernet over MPLS (Layer 2 Circuit) | **2043**
Configuring the Local PE Switch for Port-Based Layer 2 Circuit (Pseudo-wire) | **2044**
Configuring the Remote PE Switch for Port-Based Layer 2 Circuit (Pseudo-wire) | **2046**
Configuring the Local PE Switch for VLAN-Based Layer 2 Circuit | **2046**
Configuring the Remote PE Switch for VLAN-Based Layer 2 Circuit | **2048**

CCC, TCC, and Layer 2.5 Switching | 2050

CCC, TCC, and Ethernet Over MPLS Configuration | **2050**
TCC and Layer 2.5 Switching Overview | **2051**
Configuring VLAN TCC Encapsulation | **2051**
Configuring TCC Interface Switching | **2053**
CCC Overview | **2055**
Understanding Carrier-of-Carriers VPNs | **2056**

Understanding Interprovider and Carrier-of-Carriers VPNs | **2058**

Configuring BGP to Gather Interprovider and Carrier-of-Carriers VPNs Statistics | **2059**

Configuring an MPLS-Based VLAN CCC Using a Layer 2 Circuit | **2060**

VLAN CCC Encapsulation on Transport Side of Pseudowire Client Logical Interfaces Overview | **2064**

Transmitting Nonstandard BPDUs | **2067**

TCC Overview | **2067**

Configuring Layer 2 Switching Cross-Connects Using CCC | **2068**

Configuring MPLS LSP Tunnel Cross-Connects Using CCC | **2079**

Configuring TCC | **2084**

CCC and TCC Graceful Restart | **2090**

Configuring CCC and TCC Graceful Restart | **2091**

Configuring an MPLS-Based VLAN CCC Using the Connection Method (CLI Procedure) | **2092**

Configuring CCC Switching for Point-to-Multipoint LSPs | **2094**

Configuring an MPLS-Based VLAN CCC Using a Layer 2 VPN (CLI Procedure) | **2096**

Understanding Ethernet-over-MPLS (L2 Circuit) | **2101**

Configuring Ethernet over MPLS (Layer 2 Circuit) | **2102**

- Configuring the Local PE Switch for Port-Based Layer 2 Circuit (Pseudo-wire) | **2104**
- Configuring the Remote PE Switch for Port-Based Layer 2 Circuit (Pseudo-wire) | **2105**
- Configuring the Local PE Switch for VLAN-Based Layer 2 Circuit | **2106**
- Configuring the Remote PE Switch for VLAN-Based Layer 2 Circuit | **2107**

9 **MPLS for Software Defined Networking (SDN)**

Path Computation Element Protocol (PCEP) | 2111

PCEP Configuration | **2111**

Path Computation Element Protocol (PCEP) for RSVP LSPs and Segment Routing | **2112**

Support of the Path Computation Element Protocol for RSVP-TE Overview | **2113**

Example: Configuring the Path Computation Element Protocol for MPLS RSVP-TE | **2132**

- Requirements | **2133**
- Overview | **2133**
- Configuration | **2136**
- Verification | **2143**

Example: Configuring Path Computation Element Protocol for MPLS RSVP-TE with Support of PCE-Initiated Point-to-Point LSPs | **2150**

- Requirements | **2150**
- Overview | **2150**
- Configuration | **2153**

Verification | **2158**

Configuring Path Computation Element Protocol for MPLS RSVP-TE with Support of PCE-Initiated Point-to-Point LSPs | **2162**

Example: Configuring Path Computation Element Protocol for MPLS RSVP-TE with Support for PCE-Controlled Point-to-Multipoint LSPs | **2166**

Requirements | **2167**

Overview | **2167**

Configuration | **2169**

Verification | **2183**

Understanding Path Computation Element Protocol for MPLS RSVP-TE with Support for PCE-Initiated Point-to-Multipoint LSPs | **2187**

Enable Segment Routing for the Path Computation Element Protocol | **2192**

Segment Routing for the Path Computation Element Protocol Overview | **2192**

Example: Configure Segment Routing for the Path Computation Element Protocol | **2200**

Static Segment Routing Label Switched Path | **2234**

Static Segment Routing LSP in MPLS Networks | **2235**

Example: Configuring Static Segment Routing Label Switched Path | **2253**

Enabling Distributed CSPF for Segment Routing LSPs | **2273**

Example: Configuring CoS-Based Forwarding and Policy-Based Routing For SR-TE LSPs | **2281**

CoS-Based Forwarding and Policy-Based Routing For SR-TE LSPs Overview | **2281**

Configure CoS-Based Forwarding and Policy-Based Routing for SR-TE LSPs | **2282**

Enable Multiple Paths for SR-TE LSPs in PCEP | **2291**

Enable Transport Layer Security for PCEP Sessions | **2295**

Reporting Path Optimization and Computed Metrics in PCEP | **2301**

SRv6-TE Tunnels with micro-SIDs in PCEP | **2308**

10 **MPLS Troubleshooting**

Troubleshooting MPLS | 2312

Troubleshooting MPLS | **2312**

Verify MPLS Interfaces | **2313**

Verify Protocol Families | **2316**

Verify the MPLS Configuration | **2320**

Checking the MPLS Layer | **2323**

Verify the LSP | **2326**

Verify the LSP Route on the Transit Router | **2330**

Verify the LSP Route on the Ingress Router | **2332**

Verify MPLS Labels with the traceroute Command | **2334**

- Verify MPLS Labels with the ping Command | **2336**
- Take Appropriate Action | **2338**
- Verify the LSP Again | **2340**

Verify That Node-Link Protection Is Up | **2344**

Verify That Link Protection Is Up | **2352**

Verify One-to-One Backup | **2358**

Verify That the Primary Path Is Operational | **2366**

Verify That the Secondary Path Is Established | **2368**

Verifying the Physical Layer | **2371**

- Verify the LSP | **2374**
- Verify Router Connection | **2376**
- Verify Interfaces | **2377**
- Take Appropriate Action | **2379**
- Verify the LSP Again | **2380**

Checking the Data Link Layer | **2382**

- Verify the LSP | **2385**
- Verify Interfaces | **2387**
- Take Appropriate Action | **2392**
- Verify the LSP Again | **2394**

Verifying the IP and IGP Layers | **2399**

Verifying the IP Layer | **2402**

- Verify the LSP | **2403**
- Verify IP Addressing | **2404**
- Verify Neighbors or Adjacencies at the IP Layer | **2407**
- Take Appropriate Action | **2411**
- Verify the LSP Again | **2413**

Verify the LSP Again | **2417**

Checking the RSVP Layer | **2421**

- Verify the LSP | **2424**
- Verify RSVP Sessions | **2426**
- Verify RSVP Neighbors | **2429**
- Verify RSVP Interfaces | **2431**
- Verify the RSVP Protocol Configuration | **2433**
- Take Appropriate Action | **2434**
- Verify the LSP Again | **2436**

Determining LSP Statistics | **2441**

Verifying LSP Use in Your Network | **2443**

- Verifying an LSP on the Ingress Router | **2444**
- Verifying an LSP on a Transit Router | **2446**

Verify That Load Balancing Is Working | **2448**

Verify the Operation of Uneven Bandwidth Load Balancing | **2453**

Use the traceroute Command to Verify MPLS Labels | **2455**

Troubleshooting GMPLS and GRE Tunnel | **2457**

Determining LSP Status | **2482**

- Check the Status of the LSP | **2483**
- Display Extensive Status About the LSP | **2484**

Checking That RSVP Path Messages Are Sent and Received | **2489**

11 **Configuration Statements and Operational Commands**

Junos CLI Reference Overview | 2494

About This Guide

Use this guide to understand the MPLS technology and MPLS applications functions, and to configure MPLS and other feature modules deploying the MPLS applications.

RELATED DOCUMENTATION

- Day One: Deploying MPLS
- Day One: MPLS for Enterprise Engineers

1

PART

Overview

- Understanding MPLS | **2**
- Supported Standards | **36**

CHAPTER 1

Understanding MPLS

IN THIS CHAPTER

- MPLS Overview | 2

MPLS Overview

IN THIS SECTION

- MPLS Overview | 2
- TTL Processing on Incoming MPLS Packets | 7
- Link-Layer Support in MPLS | 10
- MPLS Overview for ACX Series Universal Metro Routers | 10
- MPLS for EX Series Switches Overview | 11
- MPLS Feature Support on QFX Series and EX4600 Switches | 13
- MPLS Limitations on QFX Series and EX4600 Switches | 30

MPLS Overview

IN THIS SECTION

- Why Use MPLS? | 3
- Why Not Use MPLS? | 4
- How Do I Configure MPLS? | 4
- What Does the MPLS Protocol Do? | 5

How Does MPLS Interface to Other Protocols? | 6
If I Have Used Cisco MPLS, What Do I Need to Know? | 7

Multiprotocol Label Switching (MPLS) is a protocol that uses labels to route packets instead of using IP addresses. In a traditional network, each switch performs an IP routing lookup, determines a next-hop based on its routing table, and then forwards a packet to that next-hop. With MPLS, only the first device does a routing lookup, and, instead of finding the next-hop, finds the ultimate destination along with a path to that destination. The path of an MPLS packet is called a label-switched path (LSP).

MPLS applies one or more labels to a packet so it can follow the LSP to the destination. Each switch pops off its label and sends the packet to the next switch label in the sequence.

The Junos OS includes everything you need to configure MPLS. You do not need to install any additional programs or protocols. MPLS is supported on switches with a subset of the commands supported on routers. The Junos MPLS-configured switches can interact with each other and with Junos MPLS-configured routers.

MPLS has the following advantages over conventional packet forwarding:

- Packets arriving on different ports can be assigned different labels.
- A packet arriving at a particular provider edge (PE) switch can be assigned a label that is different from that of the same packet entering the network at a different PE switch. As a result, forwarding decisions that depend on the ingress PE switch can be easily made.
- Sometimes it is desirable to force a packet to follow a particular route that is explicitly chosen at or before the time the packet enters the network, rather than letting it follow the route chosen by the normal dynamic routing algorithm as the packet travels through the network. In MPLS, a label can be used to represent the route so that the packet need not carry the identity of the explicit route.

This topic describes:

Why Use MPLS?

MPLS reduces the use of the forwarding table by using labels instead of the forwarding table. The size of forwarding tables on a switch are limited by silicon and using exact matching for forwarding to destination devices is cheaper than buying more sophisticated hardware. In addition, MPLS allows you to control where and how traffic is routed on your network – this is called traffic engineering.

Some reasons to use MPLS instead of another switching solution are:

- MPLS can connect different technologies that would not otherwise be compatible---service providers have this compatibility issue when connecting clients with different autonomous systems

in their networks. In addition, MPLS has a feature called Fast Reroute that provides alternate backups for paths – this prevents network degradation in case of a switch failure.

- Other IP-based encapsulations such as Generic Route Encapsulation (GRE) or Virtual Extensible Local Area Networks (VXLAN) support only two levels of hierarchy, one for the transport tunnel and one piece of metadata. Using virtual servers means that you need multiple hierarchy levels. For example, one label is needed for top-of-rack (ToR), one label for the egress port that identifies the server, and one for the virtual server.

Why Not Use MPLS?

There are no protocols to auto-discover MPLS enabled nodes. MPLS protocol just exchanges label values for an LSP. They do not create the LSPs.

You must build the MPLS mesh, switch by switch. We recommend using scripts for this repetitive process.

MPLS hides suboptimal topologies from BGP where multiple exits may exist for the same route.

Large LSPs are limited by the circuits they traverse. You can work around this by creating multiple, parallel LSPs.

How Do I Configure MPLS?

There are three types of switches you must set up for MPLS:

- Label Edge Router/Switch (LER) or ingress node to the MPLS network. This switch encapsulates the packets.
- Label Switching Routers/Switches (LSR). One or more switches that transfer MPLS packets in the MPLS network.
- Egress router/switch is the final MPLS device that removes the last label before packets leave the MPLS network.

Service providers (SP) use the term provider router (P) for a backbone router/switch doing label switching only. The customer-facing router at the SP is called a provider edge router (PE). Each customer needs a customer edge router (CE) to communicate with the PE. Customer facing routers typically can terminate IP addresses, L3VPNs, L2VPNs/ pseudowires, and VPLS before packets are transferred to the CE.

Configure the MPLS LER (Ingress) Switch and the Egress Switch

To configure MPLS, you must first create one or more named paths on the ingress and egress routers. For each path, you can specify some or all transit routers in the path, or you can leave it empty. See

"Configuring the Ingress and Egress Router Addresses for LSPs" on page 591 and "Configuring the Connection Between Ingress and Egress Routers" on page 599.

Configure LSRs for MPLS

Configure one or more MPLS LSRs by following these steps:

1. Configure interfaces on each switch to transmit and receive MPLS packets using the usual interface command with MPLS appended. For example:

```
[edit interfaces ge-0/0/0 unit 0] family mpls;
```

2. Add those same interfaces under [edit protocols mpls]. For example:

```
[edit protocols mpls]
interface ge-0/0/0;
```

3. Configure the interfaces on each switch to handle MPLS labels with a protocol. For example, for LDP:

```
[edit protocols ldp]
Interface ge-0/0/0.0;
```

   To watch a demo of these configurations, see https://www.youtube.com/watch?v=xegWBCUJ4tE.

What Does the MPLS Protocol Do?

Multiprotocol Label Switching (MPLS) is an Internet Engineering Task Force (IETF)-specified framework that provides for the designation, routing, forwarding and switching of traffic flows through the network. In addition, MPLS:

- Specifies mechanisms to manage traffic flows of various granularities, such as flows between different hardware, machines, or even flows between different applications.
- Remains independent of the layer-2 and layer-3 protocols.
- Provides a means to map IP addresses to simple, fixed-length labels used by different packet-forwarding and packet-switching technologies.
- Interfaces to existing routing protocols, such as Resource ReSerVation Protocol (RSVP) and Open Shortest PathFirst (OSPF).

- Supports IP, ATM, and Frame Relay layer-2 protocols.
- Uses these additional technologies:
    - FRR: MPLS Fast Reroute improves convergence during a failure by mapping out alternate LSPs in advance.
    - Link Protection/ Next-hop backup: A bypass LSP is created for every possible link failure.
    - Node Protection/ Next-hop backup: A bypass LSP is created for every possible switch (node) failure.
    - VPLS: Creates Ethernet multipoint switching service over MPLS and emulates functions of an L2 switch.
    - L3VPN: IP-based VPN customers get individual virtual routing domains.

How Does MPLS Interface to Other Protocols?

Some of the protocols that work with MPLS are:

- RSVP-TE: Resource Reservation Protocol - Traffic Engineering reserves bandwidth for LSPs.
- LDP: Label Distribution Protocol is the defacto protocol used for distribution of MPLS packets and is usually configured to tunnel inside RSVP-TE.
- IGP: Interior Gateway Protocol is a routing protocol. Edge routers (PE-routers) run BGP between themselves to exchange external (customer) prefixes. Edge and core (P) routers run IGP (usually OSPF or IS-IS) to find optimum path toward BGP next hops. P- and PE-routers use LDP to exchange labels for known IP prefixes (including BGP next hops). LDP indirectly builds end-to-end LSPs across the network core.
- BGP: Border Gateway Protocol (BGP) allows policy-based routing to take place, using TCP as its transport protocol on port 179 to establish connections. The Junos OS routing protocol software includes BGP version 4. You do not configure BGP---configuring interfaces with MPLS and LDP/ RSVP establishes the labels and the ability to transmit packets. BGP automatically determines the routes packets take.
- OSPF and ISIS: These protocols are used for routing between the MPLS PE and CE. Open Shortest Path First (OSPF) is perhaps the most widely used interior gateway protocol (IGP) in large enterprise networks. IS-IS, another link-state dynamic routing protocol, is more common in large service provider networks. Assuming you're running L3VPN to your customers, on the SP edge between the PE and the CE you can run any protocol that your platform supports as a VRF aware instance.

If I Have Used Cisco MPLS, What Do I Need to Know?

Cisco Networks and Juniper Networks use different MPLS terminology.

What Cisco Calls:	Juniper Calls:
affinities	admin-groups
autoroute announce	TE shortcuts
forwarding adjacency	LSP-advertise
tunnel	LSP
make-before-break	adaptive
application-window	adjust-interval
shared risk link groups	fate-sharing

TTL Processing on Incoming MPLS Packets

The flow chart on Figure 1 on page 9 illustrates TTL processing on incoming MPLS packets. On a transit LSR or an egress LER, MPLS pops one or more labels and can push one or more labels. The incoming TTL of the packet is determined by the configured TTL processing tunnel model.

When all of the following conditions are met, the incoming TTL is set to the TTL value found in the immediate inner header:

- The outer label is popped as opposed to being swapped
- The TTL processing model is configured to pipe
- The inner header is MPLS or IP

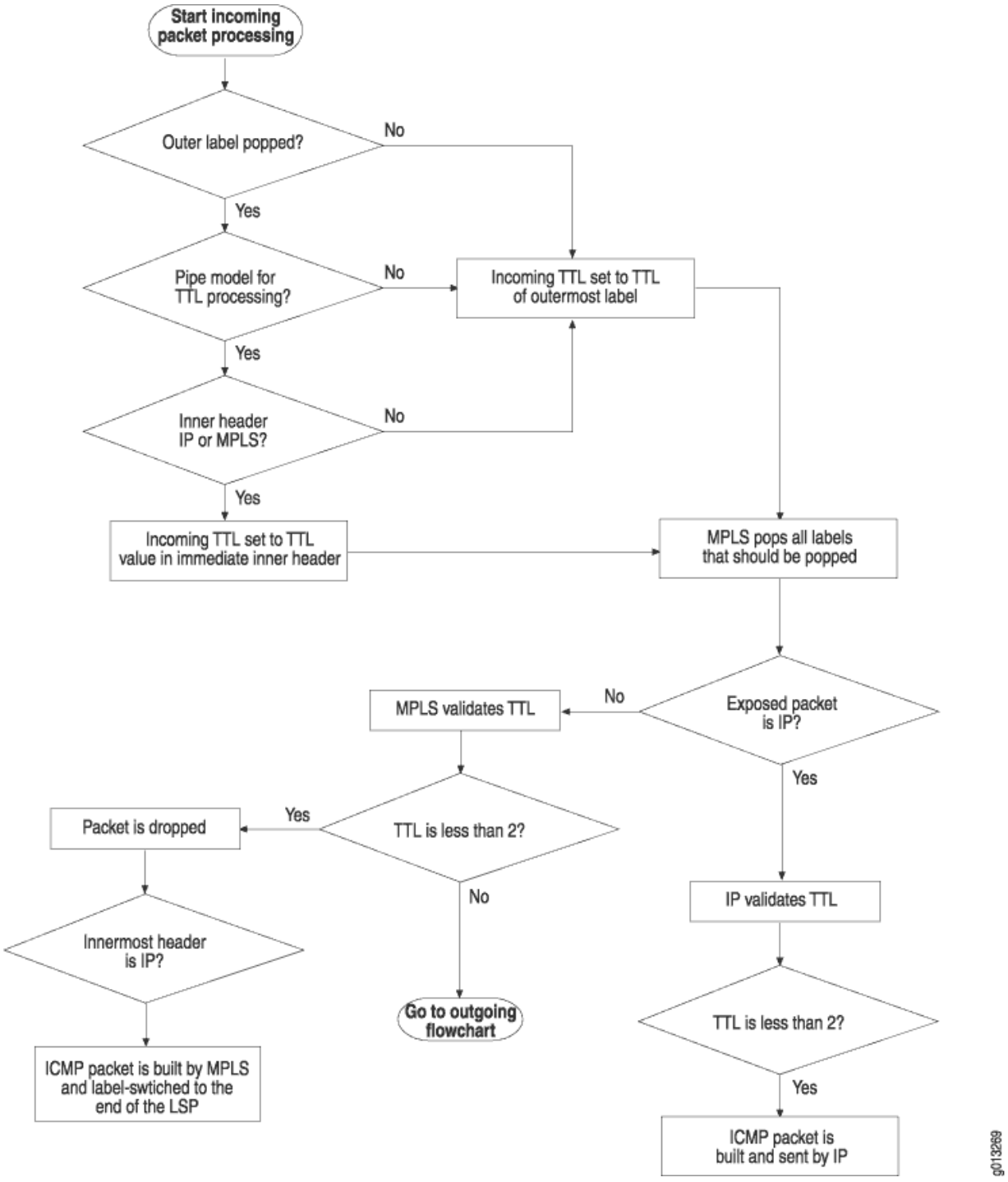
If any of those conditions is not met, then the incoming TTL is set to the TTL value found in the outermost label. In all cases, the TTL values of any further inner labels are ignored.

When an IP packet is exposed after MPLS pops all the labels that should be popped, MPLS passes the packet to IP for further processing, including TTL checking. When the uniform tunnel model for TTL processing is in effect, MPLS sets the TTL value of the IP packet to the incoming TTL value that was just

set. In other words, the TTL value is copied from the outermost label to the IP packet. When the pipe model for TTL processing is in effect, the TTL value in the IP header is left unchanged.

If an IP packet is not exposed by the label popping, then MPLS performs the TTL validation. If the incoming TTL is less than 2, the packet is dropped. If innermost packet is IP, an ICMP packet is built and sent. If the TTL does not expire and the packet needs to be sent out, the outgoing TTL is determined by the rules for outgoing MPLS packets.

Figure 1: TTL Processing on Incoming MPLS Packets



SEE ALSO

> Disabling Normal TTL Decrementing
>
> *no-propagate-ttl*

Link-Layer Support in MPLS

MPLS supports the following link-layer protocols, which are all supported in the Junos OS MPLS implementation:

- Point-to-Point Protocol (PPP)—Protocol ID 0x0281, Network Control Protocol (NCP) protocol ID 0x8281.
- Ethernet/Cisco High-level Data Link Control (HDLC)—Ethernet type 0x8847.
- Asynchronous Transfer Mode (ATM)—Subnetwork attachment point encoded (SNAP-encoded) Ethernet type 0x8847. Support is included for both point-to-point mode or nonbroadcast multiaccess (NBMA) mode. Support is not included for encoding MPLS labels as part of ATM virtual path identifier/virtual circuit identifier (VPI/VCI).
- Frame Relay—SNAP-encoded, Ethernet type 0x8847. Support is not included for encoding MPLS labels as part of Frame Relay data-link connection identifier (DLCI).
- Generic routing encapsulation (GRE) tunnel—Ethernet type 0x8847.

MPLS Overview for ACX Series Universal Metro Routers

Multiprotocol Label Switching (MPLS) provides a mechanism for engineering network traffic patterns that is independent of routing tables by assigning short labels to network packets, which describe how to forward them through the network. MPLS is independent of any routing protocol and can be used for unicast packets. On the ACX Series routers, the following MPLS features are supported:

- The configuration of a *label-switching router* (LSR) for processing of label-switched packets and forwarding of packets based on their labels.
- The configuration of an ingress label edge router (LER) where IP packets are encapsulated within MPLS packets and forwarded to the MPLS domain, and as an egress LER where MPLS packets are decapsulated and the IP packets contained within the MPLS packets are forwarded using information in the IP forwarding table. Configuring MPLS on the LER is the same as configuring an LSR.
- Uniform and pipe mode configuration providing different types of visibility in the MPLS network. Uniform mode makes all the nodes that a label-switched path (LSP) traverses visible to nodes outside the LSP tunnel. Uniform mode is the default. Pipe mode makes only the LSP ingress and egress points visible to nodes outside the LSP tunnel. Pipe mode acts like a circuit and must be enabled with the global `no-propagate-ttl` statement at the [**edit protocols mpls**] hierarchy level on each router that is in the path of the LSP. The `no-propagate-ttl` statement disables time-to-live (TTL) propagation at the

router level and affects all RSVP-signalled or LDP-signalled LSPs. Only the global configuration of TTL propagation is supported.

- Exception packet handling of IP packets not processed by the normal packet flow through the Packet Forwarding Engine. The following types of exception packet handling are supported:
  - Router alert
  - Time-to-live (TTL) expiry value
  - Virtual circuit connection verification (VCCV)
- LSP hot standby for secondary paths configuration to maintain a path in a hot-standby state enabling swift cut over to the secondary path when downstream routers on the current active path indicate connectivity problems.
- Redundancy for a label-switched path (LSP) path with the configuration of fast reroute.
- Configuration of link protection to ensure that traffic traversing a specific interface from one router to another can continue to reach its destination in the event that this interface fails.

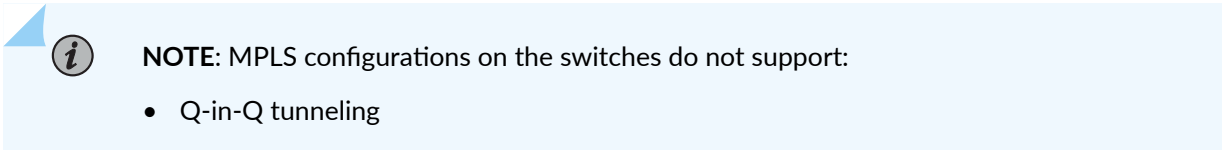
MPLS for EX Series Switches Overview

IN THIS SECTION

- Benefits of MPLS | **12**
- Additional Benefits of MPLS and Traffic Engineering | **12**

You can configure Junos OS MPLS on Juniper Networks EX4650 and EX9200 Ethernet Switches to increase transport efficiency in the network. MPLS services can be used to connect various sites to a backbone network and to ensure better performance for low-latency applications such as voice over IP (VoIP) and other business-critical functions.

NOTE: MPLS configurations on EX4650 and EX9200 switches are compatible with configurations on other Juniper Networks devices that support MPLS and MPLS-based circuit cross-connect (CCC). MPLS features available on the switches depend upon which switch you are using. For information about the software features on the EX Series switches, see Feature Explorer.



NOTE: MPLS configurations on the switches do not support:

- Q-in-Q tunneling

This topic describes:

Benefits of MPLS

MPLS has the following advantages over conventional packet forwarding:

- Packets arriving on different ports can be assigned different labels.
- A packet arriving at a particular provider edge (PE) switch can be assigned a label that is different from that of the same packet entering the network at a different PE switch. As a result, forwarding decisions that depend on the ingress PE switch can be easily made.
- Sometimes it is desirable to force a packet to follow a particular route that is explicitly chosen at or before the time the packet enters the network, rather than letting it follow the route chosen by the normal dynamic routing algorithm as the packet travels through the network. In MPLS, a label can be used to represent the route so that the packet need not carry the identity of the explicit route.

Additional Benefits of MPLS and Traffic Engineering

MPLS is the packet-forwarding component of the Junos OS traffic engineering architecture. Traffic engineering provides the capabilities to do the following:

- Route primary paths around known bottlenecks or points of congestion in the network.
- Provide precise control over how traffic is rerouted when the primary path is faced with single or multiple failures.
- Provide efficient use of available aggregate bandwidth and long-haul fiber by ensuring that certain subsets of the network are not overutilized while other subsets of the network along potential alternate paths are underutilized.
- Maximize operational efficiency.
- Enhance the traffic-oriented performance characteristics of the network by minimizing packet loss, minimizing prolonged periods of congestion, and maximizing throughput.
- Enhance statistically bound performance characteristics of the network (such as loss ratio, delay variation, and transfer delay) required to support a multiservice Internet.

MPLS Feature Support on QFX Series and EX4600 Switches

IN THIS SECTION

- Supported Features | 13

This topic describes the MPLS features that are supported on the QFX Series, EX4600, EX4650 switches. Be sure to check for any exceptions to this support in "MPLS Limitations on QFX Series and EX4600 Switches" on page 30. Configuring unsupported statements on the switch does not affect its operation.

NOTE: EX4600 and EX4650 switches use the same chipset as QFX5100 switches—this is why EX Series switches are included here along with QFX Series switches. Other EX Series switches also support MPLS but with a different feature set.

Supported Features

The tables in this section lists the MPLS features that are supported on the QFX Series, EX4600, EX4650 switches, and the Junos OS release in which they were introduced. Table 1 on page 13 lists the features for QFX10000 switches. Table 2 on page 19 lists the features for QFX3500, QFX5100, QFX5120, QFX5110, QFX5200, QFX5210 switches.Table 3 on page 27 lists the features for EX4600 and EX4650 switches.

Table 1: QFX10000 MPLS Features

Feature	QFX10002	QFX10008	QFX10016
QFX10000 standalone switch as an MPLS provider edge (PE) switch or provider switch	15.1X53-D10	15.1X53-D30	15.1X53-D60

Table 1: QFX10000 MPLS Features *(Continued)*

Feature	QFX10002	QFX10008	QFX10016
Label edge router (LER)	15.1X53-D10	15.1X53-D30	15.1X53-D60
Label-switching router (LSR)	15.1X53-D10	15.1X53-D30	15.1X53-D60
BGP MPLS Ethernet VPN (EVPN)	17.4R1	17.4R1	17.4R1
BGP route reflectors	15.1X53-D10	15.1X53-D30	15.1X53-D60
Automatic bandwidth and dynamic label-switched path (LSP) count sizing	15.1X53-D60	15.1X53-D60, 17.2R1	15.1X53-D60, 17.2R1
BGP labeled unicast	15.1X53-D10	15.1X53-D30	15.1X53-D60
BGP link state distribution	17.1R1	17.1R1	17.1R1
Carrier-of-carriers and interprovider Layer 3 VPNs	17.1R1	17.1R1	17.1R1
Entropy labels	17.2R1	17.2R1	17.2R1

Table 1: QFX10000 MPLS Features *(Continued)*

Feature	QFX10002	QFX10008	QFX10016
Ethernet-over-MPLS (L2 circuit)	15.1X53-D60	15.1X53-D60	15.1X53-D60
Fast reroute, one-to-one local protection and many-to-one local protection	15.1X53-D10	15.1X53-D30	15.1X53-D60
Fast reroute using detours and secondary LSP	15.1X53-D10	15.1X53-D30	15.1X53-D60
Flexible Ethernet services	17.3R1	17.3R1	17.3R1
Firewall filters	15.1X53-D30	15.1X53-D30	15.1X53-D60
RSVP graceful restart for OSPF	15.1X53-D10	15.1X53-D30	15.1X53-D60
IP-over-MPLS LSPs, both static and dynamic links	15.1X53-D10	15.1X53-D30	15.1X53-D60
IPv6 tunneling over an IPv4 network (6PE)	15.1X53-D10	15.1X53-D30	15.1X53-D60

Table 1: QFX10000 MPLS Features *(Continued)*

Feature	QFX10002	QFX10008	QFX10016
LDP tunneling over RSVP	15.1X53-D10	15.1X53-D30	15.1X53-D60
L2 Circuit on aggregated interfaces	17.3R1	17.3R1	17.3R1
L3VPNs for both IPv4 and IPv6	15.1X53-D10	15.1X53-D30	15.1X53-D60
MPLS over integrated bridging and routing (IRB) interfaces	15.1X53-D10	15.1X53-D30	15.1X53-D60
MPLS over UDP	18.3R1	18.3R1	18.3R1
MTU signaling in RSVP	15.1X53-D10	15.1X53-D30	15.1X53-D60
Operation, Administration, and Maintenance (OAM) including ping, traceroute and Bidirectional Forwarding Detection (BFD)	15.1X53-D10	15.1X53-D30	15.1X53-D60
OSPF TE	15.1X53-D10	15.1X53-D30	15.1X53-D60

Table 1: QFX10000 MPLS Features *(Continued)*

Feature	QFX10002	QFX10008	QFX10016
OSPFv2 as an interior gateway protocol (IGP)	15.1X53-D10	15.1X53-D30	15.1X53-D60
Path Computation Element Protocol for RSVP-TE	16.3R1	16.3R1	16.3R1
Pseudowire-over-aggregated Ethernet interfaces (core-facing interface)	15.1X53-D60 (supported only on network-to-network (NNI) interfaces)	15.1X53-D60 (supported only on NNI interfaces)	15.1X53-D60 (supported only on NNI interfaces)
RSVP support, including bandwidth allocation and traffic engineering	15.1X53-D10	15.1X53-D30	15.1X53-D60
RSVP fast reroute (FRR), including link-protection, node-link-protection, fast reroute using detours, and secondary LSP	15.1X53-D10	15.1X53-D30	15.1X53-D60

Table 1: QFX10000 MPLS Features *(Continued)*

Feature	QFX10002	QFX10008	QFX10016
SNMP MIB support	15.1X53-D10	15.1X54-D30	15.1X53-D60
Static and dynamic LSPs	15.1X53-D10	15.1X53-D30	15.1X53-D60
Traffic engineering extensions (OSPF-TE, IS-IS-TE)	15.1X53-D10	15.1X53-D30	15.1X53-D60
Traffic engineering (TE) Automatic bandwidth allocation and RSVP bandwidth Dynamic bandwidth management using ingress LSP splitting and merging	15.1X53-D10	15.1X53-D30	15.1X53-D60
Virtual routing and forwarding (VRF) label support	15.1X53-D10	15.1X53-D30	15.1X53-D60

Table 2: QFX3500, QFX5100, QFX5110, QFX5120, QFX5200, QFX5210 MPLS Features

Feature	QFX3500	QFX5100	QFX5110	QFX5120	QFX5200	QFX5210
QFX Series standalone switches as MPLS provider edge (PE) switches or provider switches	12.2X50-D10	13.2X51-D15 VC/VCF (14.1X53-D30)	15.1X53-D210	18.3R1	15.1X53-D30	18.1R1
Label edge router (LER)	12.2X50-D10	13.2X51-D15 VC/VCF (14.1X53-D30)	15.1X53-D210	18.3R1	15.1X53-D30	18.1R1
Label-switching router (LSR)	12.2X50-D10	13.2X51-D15 VC/VCF (14.1X53-D30)	15.1X53-D210	18.3R1	15.1X53-D30	18.1R1
Automatic bandwidth allocation on LSPs	Not supported	13.2X51-D15 VC/VCF (14.1X53-D30)	15.1X53-D210	18.3R1	15.1X53-D30	18.1R1
BGP labeled unicast	12.2X50-D10	13.2X51-D15 VC/VCF (14.1X53-D30)	15.1X53-D210	18.3R1	15.1X53-D30	18.1R1

Table 2: QFX3500, QFX5100, QFX5110, QFX5120, QFX5200, QFX5210 MPLS Features *(Continued)*

Feature	QFX3500	QFX5100	QFX5110	QFX5120	QFX5200	QFX5210
BGP link state distribution	Not supported	17.1R1	17.1R1	18.3R1	17.1R1	18.1R1
BGP route reflector	15.1X53-D10	15.1X53-D30	15.1X53-D210	18.3R1	15.1X53-D30	18.1R1
Carrier-to-carrier and interprovider BGP Layer 3 VPNs	14.1X53-D15	14.1X53-D15 VC/VCF (14.1X53-D30)	15.1X53-D210	18.3R1	15.1X53-D30	18.1R1
Class of service (CoS or QoS) for MPLS traffic	12.3X50-D10	13.2X51-D15 VC/VCF (14.1X53-D30)	15.1X53-D210	18.3R1	15.1X53-D30	18.1R1
Dynamic label-switched path (LSP) count sizing: TE++	Not supported	17.2R1 VC/VCF 17.2R1	17.2R1 VC/VCF 17.2R1	18.3R1	17.2R1	18.1R1

Table 2: QFX3500, QFX5100, QFX5110, QFX5120, QFX5200, QFX5210 MPLS Features *(Continued)*

Feature	QFX3500	QFX5100	QFX5110	QFX5120	QFX5200	QFX5210
Equal-cost multipath (ECMP) at LSRs: • SWAP • PHP • L3VPN • L2 Circuit	Not supported	14.1X53-D35 (Supported only on label stack. Not supported on flow label, entropy label, or ECMP label)	15.1X53-D210 (Supported only on label stack. Not supported on flow label, entropy label, or ECMP label)	18.3R1 (Supported only on label stack. Not supported on flow label, entropy label, or ECMP label)	15.1X53-D30	18.1R1
Entropy labels	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported
Ethernet-over-MPLS (L2 Circuit)	14.1X53-D10	14.1X53-D10 VC/VCF (14.1X53-D30)	15.1X53-D210	18.3R1	15.1X53-D30	18.1R1
Fast reroute (FRR), one-to-one local protection and many-to-one local protection	14.1X53-D10	14.1X53-D10	15.1X53-D210	18.3R1	15.1X53-D30	18.1R1
FRR using detours and secondary LSP	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported

Table 2: QFX3500, QFX5100, QFX5110, QFX5120, QFX5200, QFX5210 MPLS Features *(Continued)*

Feature	QFX3500	QFX5100	QFX5110	QFX5120	QFX5200	QFX5210
Firewall filters	12.3X50-D10	13.2X51-D15 VC/VCF (14.1X53-D30)	15.1X53-D210	18.3R1	15.1X53-D30	18.1R1
Flow-aware transport of pseudowires (FAT) flow labels	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported
RSVP graceful restart for OSPF	12.2X50-D10	13.2X51-D15 VC/VCF (14.1X53-D30)	15.1X53-D210	18.3R1	15.1X53-D30	18.1R1
Traffic engineering extensions (OSPF-TE, IS-IS-TE)	12.2X50-D10	13.2X51-D15 VC/VCF (14.1X53-D30)	15.1X53-D210	18.3R1	15.1X53-D30	18.1R1
IP-over-MPLS LSPs, both static and dynamic links	12.2X50-D10	13.2X51-D15 VC/VCF (14.1X53-D30)	15.1X53-D210	18.3R1	15.1X53-D30	18.1R1

Table 2: QFX3500, QFX5100, QFX5110, QFX5120, QFX5200, QFX5210 MPLS Features *(Continued)*

Feature	QFX3500	QFX5100	QFX5110	QFX5120	QFX5200	QFX5210
IPv6 tunneling over an MPLS IPv4 network (6PE)	12.3X50-D10	13.2X51-D15 VC/VCF (14.1X53-D30)	15.1X53-D210	18.3R1	15.1X53-D30	18.1R1
IPv6 over an MPLS core network	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported
LDP tunneling over RSVP	12.2X50-D10	13.2X51-D15 VC/VCF (14.1X53-D30)	15.1X53-D210	18.3R1	15.1X53-D30	18.1R1
Layer 3 VPNs for both IPv4 and IPv6	12.3X50-D10	13.2X51-D15 VC/VCF (14.1X53-D30)	15.1X53-D210	18.3R1	15.1X53-D30	18.1R1
Loop-free alternate (LFA)	Not supported	13.2X51-D15 VC/VCF (14.1X53-D30)	15.1X53-D210	18.3R1	18.1R1	18.1R1
MPLS over integrated bridging and routing (IRB) interfaces	Not supported	14.1X53-D40	18.1R1	18.3R1	18.1R1	18.1R1

Table 2: QFX3500, QFX5100, QFX5110, QFX5120, QFX5200, QFX5210 MPLS Features *(Continued)*

Feature	QFX3500	QFX5100	QFX5110	QFX5120	QFX5200	QFX5210
MTU signaling in RSVP	12.3X50-D10	13.2X51-D15 VC/VCF (14.1X53-D30)	15.1X53-D210	18.3R1	15.1X53-D30	18.1R1
Operation, Administration, and Maintenance (OAM) including MPLS ping, traceroute, and BFD	12.3X50-D10	13.2X51-D15 VC/VCF (14.1X53-D30)	15.1X53-D210	18.3R1	15.1X53-D30	18.1R1
OSPF TE	12.3X50-D10	13.2X51-D15	15.1X53-D210	18.3R1	15.1X53-D30	18.1R1
OSPFv2 as an interior gateway protocol	12.2X50-D10	13.2X51-D15 VC/VCF (14.1X53-D30)	15.1X53-D210	18.3R1	15.1X53-D30	18.1R1
Path Computation Element Protocol for RSVP-TE	Not supported	17.4R1	17.4R1	18.3R1	17.4R1	18.1R1

Table 2: QFX3500, QFX5100, QFX5110, QFX5120, QFX5200, QFX5210 MPLS Features *(Continued)*

Feature	QFX3500	QFX5100	QFX5110	QFX5120	QFX5200	QFX5210
Pseudowire-over-aggregated Ethernet interfaces (core-facing interface)	14.1X53-D10	14.1X53-D15 VC/VCF (14.1X53-D30)	15.1X53-D210	18.3R1	15.1X53-D30	18.1R1
RSVP automatic bandwidth	12.2X50-D10	13.2X51-D15 VC/VCF (14.1X53-D30)	15.1X53-D210	18.3R1	15.1X53-D30	18.1R1
RSVP fast reroute (FRR), including link-protection, node-link-protection, fast reroute using detours, and secondary LSP	14.1X53-D15	14.1X53-D15	15.1X53-D210	18.3R1	15.1X53-D30	18.1R1
RSVP-TE extensions (IS-IS and OSPF)	12.2X50-D10	13.2X51-D15 VC/VCF (14.1X53-D30)	15.1X53-D210	18.3R1	15.1X53-D30	18.1R1

Table 2: QFX3500, QFX5100, QFX5110, QFX5120, QFX5200, QFX5210 MPLS Features *(Continued)*

Feature	QFX3500	QFX5100	QFX5110	QFX5120	QFX5200	QFX5210
SNMP MIB support	12.2X50-D10	13.2X51-D15 VC/VCF (14.1X53-D30)	15.1X53-D210	18.3R1	15.1X53-D30	18.1R1
Static and dynamic LSPs	12.2X50-D10	13.2X51-D10 VC/VCF (14.1X53-D30)	15.1X53-D210	18.3R1	15.1X53-D30	18.1R1
Traffic engineering (TE) automatic bandwidth allocation on LSPs	13.1X51-D10	13.1X51-D10 VC/VCF (13.2X51-D10)	15.1X53-D210	18.3R1	15.1X53-D30	18.1R1
Virtual routing and forwarding (VRF) label support	12.2X50-D10	13.2X51-D15 VC/VCF (14.1X53-D30)	15.1X53-D210	18.3R1	15.1X53-D30	18.1R1
VRF support in IRB Interfaces in a Layer 3 VPN	Not supported	17.3R1	17.3R1	18.3R1	17.3R1	18.1R1

Table 3: EX4600 and EX4650 MPLS Features

Feature	EX4600	EX4650
EX4600 and EX4650 standalone switches as MPLS provider edge (PE) switches or provider switches	14.1X53-D15	18.3R1
Label edge router (LER)	14.1X53-D15	18.3R1
Label-switching router (LSR)	14.1X53-D15	18.3R1
Automatic bandwidth allocation on LSPs	Not supported	18.3R1
BGP labeled unicast	14.1X53-D15	18.3R1
BGP link state distribution	Not supported	18.3R1
BGP route reflector	14.1X53-D15	18.3R1
Carrier-to-carrier and interprovider BGP Layer 3 VPNs	14.1X53-D15	18.3R1
Class of service (CoS or QoS) for MPLS traffic	14.1X53-D15	18.3R1
Dynamic label-switched path (LSP) count sizing: TE++	Not supported	18.3R1

Table 3: EX4600 and EX4650 MPLS Features *(Continued)*

Feature	EX4600	EX4650
Equal-cost multipath (ECMP) at LSRs: • SWAP • PHP • L3VPN • L2 Circuit	Not supported	18.3R1 (Supported only on label stack. Not supported on flow label, entropy label, or ECMP label)
Entropy labels	Not supported	Not supported
Ethernet-over-MPLS (L2 Circuit)	14.1X53-D15	18.3R1
Fast reroute (FRR), one-to-one local protection and many-to-one local protection	14.1X53-D15	18.3R1
FRR using detours and secondary LSP	Not supported	Not supported
Firewall filters	14.1X53-D15	18.3R1
Flow-aware transport of pseudowires (FAT) flow labels	Not supported	Not supported
RSVP graceful restart for OSPF	13.2X51-D25	18.3R1
Traffic engineering extensions (OSPF-TE, IS-IS-TE)	14.1X53-D15	18.3R1
IP-over-MPLS LSPs, both static and dynamic links	14.1X53-D15	18.3R1

Table 3: EX4600 and EX4650 MPLS Features *(Continued)*

Feature	EX4600	EX4650
IPv6 tunneling over an MPLS IPv4 network (6PE)	14.1X53-D15	18.3R1
IPv6 over an MPLS core network	Not supported	Not supported
LDP tunneling over RSVP	14.1X53-D15	18.3R1
Layer 3 VPNs for both IPv4 and IPv6	14.1X53-D15	18.3R1
Loop-free alternate (LFA)	Not supported	Not supported
MPLS over integrated bridging and routing (IRB) interfaces	Not supported	18.3R1
MTU signaling in RSVP	14.1X53-D15	18.3R1
Operation, Administration, and Maintenance (OAM) including MPLS ping, traceroute, and BFD	14.1X53-D15	18.3R1
OSPF TE	14.1X53-D15	18.3R1
OSPFv2 as an interior gateway protocol	13.2X51-D25	18.3R1
Path Computation Element Protocol for RSVP-TE	Not supported	18.3R1
Pseudowire-over-aggregated Ethernet interfaces (core-facing interface)	14.1X53-D15	18.3R1

Table 3: EX4600 and EX4650 MPLS Features *(Continued)*

Feature	EX4600	EX4650
RSVP automatic bandwidth	14.1X53-D15	18.3R1
RSVP fast reroute (FRR), including link-protection, node-link-protection, fast reroute using detours, and secondary LSP	14.1X53-D15	18.3R1
RSVP-TE extensions (IS-IS and OSPF)	14.1X53-D15	18.3R1
SNMP MIB support	14.1X53-D15	18.3R1
Static and dynamic LSPs	14.1X53-D15	18.3R1
Traffic engineering (TE)automatic bandwidth allocation on LSPs	14.1X53-D15	18.3R1
Virtual routing and forwarding (VRF) label support	14.1X53-D15	18.3R1
VRF support in IRB Interfaces in a Layer 3 VPN	Not supported	18.3R1

MPLS Limitations on QFX Series and EX4600 Switches

IN THIS SECTION

- MPLS Limitations on QFX10000 Switches | **31**
- MPLS Limitations on EX4600, EX4650, QFX5100, QFX5110, QFX5120, QFX5200, and QFX5210 Switches | **32**
- MPLS Limitations on QFX5100 Virtual Chassis and Virtual Chassis Fabric Switches | **34**
- MPLS Limitations on QFX3500 Switches | **35**

MPLS is a fully implemented protocol on routers, while switches support a subset of the MPLS features. The limitations of each switch are listed in a separate section here, although many of the limitations are duplicates that apply to more than one switch.

MPLS Limitations on QFX10000 Switches

- Configuring an MPLS firewall filter on a switch that is deployed as an egress provider edge (PE) switch has no effect.
- Configuring the `revert-timer` statement at the `[edit protocols mpls]` hierarchy level has no effect.
- These LDP features are not supported on the QFX10000 switches:
  - LDP multipoint
  - LDP link protection
  - LDP Bidirectional Forwarding Detection (BFD)
  - LDP Operation Administration and Management (OAM)
  - LDP multicast-only fast reroute (MoFRR)
- Pseudowire-over-aggregated Ethernet interfaces on UNI are not supported.
- MPLS-over-UDP tunnels are not supported on the following:
  - MPLS TTL propagation
  - IP fragmentation at the tunnel start point
  - CoS rewrite rules and priority propagation for RSVP LSP labels (ingress tunnels only)
  - Plain IPv6
  - Multicast traffic
  - Firewall filters on tunnel start and endpoints
  - CoS tunnel endpoints

NOTE: MPLS-over-UDP tunnels are created only if corresponding RSVP-TE, LDP, or BGP-LU tunnels are not available for the destination route.

MPLS Limitations on EX4600, EX4650, QFX5100, QFX5110, QFX5120, QFX5200, and QFX5210 Switches

- MPLS support differs on the various switches. EX4600 switches support only basic MPLS functionality while the QFX5100, QFX5110, QFX5120, QFX5200, and QFX5210 switches support some of the more advanced features. See "MPLS Feature Support on QFX Series and EX4600 Switches" on page 13 for details.

- On a QFX5100 switch, configuring integrated bridging and routing (IRB) interfaces on the MPLS core is implemented on the switch by using TCAM rules. This is the result of a chip limitation on the switch, which only allows for a limited amount of TCAM space. There is 1K TCAM space is allocated for IRB. If multiple IRBs exist, make sure that you have enough available TCAM space on the switch. To check the TCAM space, see TCAM Filter Space Allocation and Verification in QFX Devices from Junos OS 12.2x50-D20 Onward.

- (QFX5100, QFX5110, QFX5120, QFX5200, QFX5210, EX4600) When `flexible-ethernet-services` encapsulation is configured on an interface and `vlan-bridge` encapsulation is enabled on a CE connected logical interface, the switch drops packets if you also enable VLAN CCC encapsulation on a different logical unit of that same interface. Only one of the below combinations can be configured, not both:

  ```
  set interfaces xe-0/0/18 encapsulation flexible-ethernet-services
  set interfaces xe-0/0/18 unit 0 encapsulation vlan-bridge
  ```

  Or:

  ```
  set interfaces xe-0/0/18 encapsulation vlan-ccc
  set interfaces xe-0/0/18 unit 0 encapsulation vlan-ccc
  ```

- Layer 2 circuits on aggregated Ethernet (AE) interfaces are not supported on QFX5100, QFX5110, QFX5120, QFX5200, and QFX5210 switches.

- Layer 2 circuit local switching is not supported on the EX4600, EX4650, QFX5100, QFX5110, QFX5120, QFX5200, and QFX5210 switches.

- The EX4600, QFX5100, QFX5110, QFX5120, QFX5200, and QFX5210 switches do not depend on the VRF match for loopback filters configured at different routing instances. Loopback filters per routing instance (such as lo0.100, lo0.103, lo0.105) are not supported and may cause unpredictable behavior. We recommend that you only apply the loopback filter (lo0.0) to the master routing instance

- On EX4600 and EX4650 switches, when loopback filters with both accept and deny terms for the same IP address are configured and if RSVP packets have that IP address in either source IP or

destination IP, then those RSVP packets will be dropped even if accept terms have higher priority than deny terms. As per design, if the switch receives an RSVP packet with IP OPTION, the packet is copied to the CPU and then the original packet is dropped. Because RSVP packets are marked for drop, the accept term will not process these packets and the deny term will drop the packets.

- On a link-protected, fast reroute Layer 2 circuit, you might see a traffic convergence delay of 200 to 300 milliseconds.
- If you configure the BGP labeled unicast address family (using the `labeled-unicast` statement at the `[edit protocols bgp family inet]` hierarchy level) on a QFX Series switch or on an EX4600 switch deployed as a route reflector for BGP labeled routes, path selection will occur at the route reflector, and a single best path will be advertised. This will result in loss of BGP multipath informaton.
- Although fast reroute (FRR) on regular interfaces is supported, the `include-all` and `include-any` options for FRR are not supported. See "Fast Reroute Overview" on page 577.
- FRR is not supported on MPLS over IRB interfaces.
- MPLS-based circuit cross-connects (CCC) are not supported—only circuit-based pseudowires are supported.
- Configuring link aggregation groups (LAGs) on user-to-network interface (UNI) ports for L2 circuits is not supported.
- MTU signaling in RSVP and discovery is supported in the control plane. However, this cannot be enforced in the data plane.
- With L2 circuit-based pseudowires, if multiple equal-cost RSVP LSPs are available to reach an L2 circuit neighbor, one LSP is randomly used for forwarding. Use this feature to specify LSPs for specific L2 circuit traffic to load-share the traffic in the MPLS core.
- Configuring an MPLS firewall filter on a switch that is deployed as an egress provider edge (PE) switch has no effect.
- Firewall filters and policers on `family mpls` are only supported on QFX5100 switches that act as pure label-switching routers (LSRs) in an MPLS network. A pure LSR is a transit router that switches paths solely on the incoming label's instructions. Firewall filters and policers on `family mpls` are not supported on QFX5100 ingress and egress provider edge (PE) switches. This includes switches that perform penultimate hop popping (PHP).
- Configuring the `revert-timer` statement at the `[edit protocols mpls]` hierarchy level has no effect.
- These are the hardware limitations for EX4600, EX4650, QFX5100, QFX5110, QFX5120, QFX5200, and QFX5210 switches:
  - Push of a maximum of three labels is supported in the MPLS edge switch if label swap is not done.

    - Push of a maximum of two labels is supported in the MPLS edge switch if label swap is done.
    - Pop at line rate is supported for a maximum of two labels.
    - Global label space is supported but interface-specific label space is not supported.
    - MPLS ECMP on PHY node with BOS=1 is not supported for single labels.
    - QFX Series switches with Broadcom chips do not support separate next hops for the same label with different S bits (S-0 and S-1). This includes the QFX3500, QFX3600, EX4600, QFX5100, and QFX5200 switches.
    - On EX4600, EX4650, QFX5100, QFX5110, QFX5120, QFX5200, and QFX5210 switches, the MPLS MTU command can cause unexpected behavior—this is due to SDK chipset limitations on this platform.
- These LDP features are not supported on the EX4600, EX4650, QFX5100, QFX5110, QFX5120, QFX5200, and QFX5210 switches:
    - LDP multipoint
    - LDP link protection
    - LDP Bidirectional Forwarding Detection (BFD)
    - LDP Operation Administration and Management (OAM)
    - LDP multicast-only fast reroute (MoFRR)
- Configuring unit with `family mpls` and unit with `encapsulation vlan-bridge` on the same physical interface is not supported on EX4600, EX4650, QFX5100, QFX5110, or QFX5120.

MPLS Limitations on QFX5100 Virtual Chassis and Virtual Chassis Fabric Switches

The following MPLS features are not supported by the QFX5100 VC and QFX5100 VCF switches:

- Next-hop LSP
- BFD including BFD triggered FRR
- L2 VPN based on BGP (See RFC 6624)
- VPLS
- Extended VLAN CCC
- Pseudowire protection using Ethernet OAM
- Local switching of pseudo-wire

- Pseudowire fault detection based on VCCV
- QFX Series switches with Broadcom chipsets do not support separate next hops for the same label with different S bits (S-0 and S-1). This includes QFX3500, QFX3600, EX4600, QFX5100, and QFX5200 switches.

MPLS Limitations on QFX3500 Switches

- If you configure the BGP labeled unicast address family (using the `labeled-unicast` statement at the `[edit protocols bgp family inet]` hierarchy level) on a QFX Series switch or on an EX4600 switch deployed as a route reflector for BGP labeled routes, path selection will occur at the route reflector, and a single best path will be advertised. This will result in loss of BGP multipath information.
- Although fast reroute is supported, the `include-all` and `include-any` options for fast reroute are not supported. See "Fast Reroute Overview" on page 577 for details.
- MPLS-based circuit cross-connects (CCC) are not supported—only circuit-based pseudowires are supported.
- MTU signaling in RSVP and discovery is supported in the control plane. However, this cannot be enforced in the data plane.
- With Layer 2 (L2) circuit-based pseudowires, if multiple equal-cost RSVP label-switched paths (LSPs) are available to reach a L2 circuit neighbor, one LSP is randomly used for forwarding. Use this feature to specify LSPs for specific L2 circuit traffic to load-share the traffic in the MPLS core.
- Configuring an MPLS firewall filter on a switch that is deployed as an egress provider edge (PE) switch has no effect.
- Configuring the `revert-timer` statement at the `[edit protocols mpls]` hierarchy level has no effect.

RELATED DOCUMENTATION

Understanding MPLS Label Operations | **526**

MPLS Configuration Guidelines | **47**

FAQ: MPLS on EX Series Switchesphysical interface is not supported on EX4600, EX4650, QFX5100, QFX511

CHAPTER 2

Supported Standards

IN THIS CHAPTER

- Supported MPLS Standards | 36

Supported MPLS Standards

IN THIS SECTION

- Supported MPLS Standards | 36
- Supported RSVP Standards | 39
- Supported LDP Standards | 41
- DiffServ-Aware Traffic Engineering Standards | 42
- Supported GMPLS Standards | 43
- Supported PCEP Standards | 44

Supported MPLS Standards

Junos OS substantially supports the following RFCs and Internet drafts, which define standards for MPLS and traffic engineering.

- RFC 2858, *Multiprotocol Extensions for BGP-4*
- RFC 3031, *Multiprotocol Label Switching Architecture*
- RFC 3032, *MPLS Label Stack Encoding*
- RFC 3140, *Per Hop Behavior Identification Codes*
- RFC 3270, *Multi-Protocol Label Switching (MPLS) Support of Differentiated Services*

Only E-LSPs are supported.

- RFC 3443, *Time To Live (TTL) Processing in Multi-Protocol Label Switching (MPLS) Networks*
- RFC 3478, *Graceful Restart Mechanism for Label Distribution Protocol*
- RFC 3906, *Calculating Interior Gateway Protocol (IGP) Routes Over Traffic Engineering Tunnels*
- RFC 4090, *Fast Reroute Extensions to RSVP-TE for LSP Tunnels*

  Node protection in facility backup is not supported.
- RFC 4124, *Protocol Extensions for Support of Diffserv-aware MPLS Traffic Engineering*
- RFC 4182, *Removing a Restriction on the use of MPLS Explicit NULL*
- RFC 4364, *BGP/MPLS IP Virtual Private Networks (VPNs)*
- RFC 4379, *Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures*
- RFC 4385, *Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN*.

  Supported on MX Series routers with the Channelized OC3/STM1 (Multi-Rate) Circuit Emulation MIC with SFP.
- RFC 4875, *Extensions to RSVP-TE for Point-to-Multipoint TE LSPs*
- RFC 4950, *ICMP Extensions for Multiprotocol Label Switching*
- RFC 5317, *Joint Working Team (JWT) Report on MPLS Architectural Considerations for a Transport Profile*
- RFC 5586, *MPLS Generic Associated Channel*
- RFC 5654, *Requirements of an MPLS Transport Profile*

  The following capabilities are supported in the Junos OS implementation of MPLS Transport Profile (MPLS-TP):

  - MPLS-TP OAM can send and receive packets with GAL and G-Ach, without IP encapsulation.
  - Two unidirectional RSVP LSPs between a pair of routers can be associated with each other to create an associated bidirectional LSP for binding a path for the GAL and G-Ach OAM messages. A single Bidirectional Forwarding Detection (BFD) session is established for the associated bidirectional LSP.
- RFC 5712, *MPLS Traffic Engineering Soft Preemption*
- RFC 5718, *An In-Band Data Communication Network For the MPLS Transport Profile*

- RFC 5860, *Requirements for Operations, Administration, and Maintenance (OAM) in MPLS Transport Networks*
- RFC 5884, *Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)*
- RFC 5921, *A Framework for MPLS in Transport Networks*
- RFC 5950, *Network Management Framework for MPLS-based Transport Networks*
- RFC 5951, *Network Management Requirements for MPLS-based Transport Networks*
- RFC 5960, *MPLS Transport Profile Data Plane Architecture*
- RFC 6215, *MPLS Transport Profile User-to-Network and Network-to-Network Interfaces*
- RFC 6291, *Guidelines for the Use of the "OAM" Acronym in the IETF.*
- RFC 6370, *MPLS Transport Profile (MPLS-TP) Identifiers*
- RFC 6371, *Operations, Administration, and Maintenance Framework for MPLS-Based Transport Networks*.
- RFC 6372, *MPLS Transport Profile (MPLS-TP) Survivability Framework*
- RFC 6373, *MPLS-TP Control Plane Framework*
- RFC 6388, *Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths*

  Only Point-to-Multipoint LSPs are supported.
- RFC 6424, *Mechanism for Performing Label Switched Path Ping (LSP Ping) over MPLS Tunnels*
- RFC 6425, *Detecting Data-Plane Failures in Point-to-Multipoint MPLS - Extensions to LSP Ping*
- RFC 6426, *MPLS On-Demand Connectivity Verification and Route Tracing*
- RFC 6428, *Proactive Connectivity Verification, Continuity Check, and Remote Defect Indication for the MPLS Transport Profile*
- RFC 6510, *Resource Reservation Protocol (RSVP) Message Formats for Label Switched Path (LSP) Attributes Objects*
- RFC 6790, *The Use of Entropy Labels in MPLS Forwarding*
- RFC 7746, *Label Switched Path (LSP) Self-Ping*
- Internet draft draft-ietf-mpls-rsvp-te-no-php-oob-mapping-01.txt, *Non PHP behavior and Out-of-Band Mapping for RSVP-TE LSPs*

The following RFCs and Internet drafts do not define standards, but provide information about MPLS, traffic engineering, and related technologies. The IETF classifies them variously as “Experimental,” “Historic,” or “Informational.”

- RFC 2547, *BGP/MPLS VPNs*
- RFC 2702, *Requirements for Traffic Engineering Over MPLS*
- RFC 2917, *A Core MPLS IP VPN Architecture*
- RFC 3063, *MPLS Loop Prevention Mechanism*
- RFC 3208, *PGM Reliable Transport Protocol Specification*

  Only the network element is supported.
- RFC 3469, *Framework for Multi-Protocol Label Switching (MPLS)-based Recovery*
- RFC 3564, *Requirements for Support of Differentiated Services-aware MPLS Traffic Engineering*
- RFC 4125, *Maximum Allocation Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering*
- RFC 4127, *Russian Dolls Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering*
- Internet draft draft-martini-l2circuit-encap-mpls-11.txt, *Encapsulation Methods for Transport of Layer 2 Frames Over IP and MPLS Networks*

  Junos OS differs from the Internet draft in the following ways:

  - A packet with a sequence number of 0 is treated as out of sequence.
  - Any packet that does not have the next incremental sequence number is considered out of sequence.
  - When out-of-sequence packets arrive, the expected sequence number for the neighbor is set to the sequence number in the Layer 2 circuit control word.
- Internet draft draft-martini-l2circuit-trans-mpls-19.txt, *Transport of Layer 2 Frames Over MPLS*
- RFC 4875, *Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs) (Support one path per S2L mode of signaling)*

Supported RSVP Standards

Junos OS substantially supports the following RFCs and Internet drafts, which define standards for RSVP.

- RFC 2205, *Resource ReSerVation Protocol (RSVP)—Version 1 Functional Specification*

- RFC 2210, *The Use of RSVP with IETF Integrated Services*
- RFC 2211, *Specification of the Controlled-Load Network Element Service*
- RFC 2212, *Specification of Guaranteed Quality of Service*
- RFC 2215, *General Characterization Parameters for Integrated Service Network Elements*
- RFC 2745, *RSVP Diagnostic Messages*
- RFC 2747, *RSVP Cryptographic Authentication* (updated by RFC 3097)
- RFC 2750, *RSVP Extensions for Policy Control* (RFC is not supported. Fully compliant with devices that support this RFC).
- RFC 2961, *RSVP Refresh Overhead Reduction Extensions*
- RFC 3097, *RSVP Cryptographic Authentication—Updated Message Type Value*
- RFC 3209, *RSVP-TE: Extensions to RSVP for LSP Tunnels*

  The Null Service Object for maximum transmission unit (MTU) signaling in RSVP is not supported.
- RFC 3210, *Applicability Statement for Extensions to RSVP for LSP-Tunnels*
- RFC 3473, *Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions*

  Only Section 9, “Fault Handling,” is supported.
- RFC 3477, *Signalling Unnumbered Links in Resource ReSerVation Protocol - Traffic Engineering (RSVP-TE)*
- RFC 4090, *Fast Reroute Extensions to RSVP-TE for LSP Tunnels*
- RFC 4203, *OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)*

  (OSPF extensions can carry traffic engineering information over unnumbered links.)
- RFC 4558, *Node-ID Based Resource Reservation Protocol (RSVP) Hello: A Clarification Statement*
- RFC 4561, *Definition of a Record Route Object (RRO) Node-Id Sub-Object*

  The RRO node ID subobject is for use in inter-AS link and node protection configurations.
- RFC 4875, *Extensions to RSVP-TE for Point-to-Multipoint TE LSPs*
- RFC 5151, *Inter-Domain MPLS and GMPLS Traffic Engineering -- Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Extensions*

- RFC 5420, *Encoding of Attributes for MPLS LSP Establishment Using Resource Reservation Protocol Traffic Engineering (RSVP-TE)*

  Only the LSP_ATTRIBUTES object is supported.

- RFC 6437, *IPv6 Flow Label Specification*
- RFC 6510, *Resource Reservation Protocol (RSVP) Message Formats for Label Switched Path (LSP) Attributes Objects*
- RFC 7570, *Label Switched Path (LSP) Attribute in the Explicit Route Object (ERO)*
- RFC 8370, *Techniques to Improve the Scalability of RSVP-TE Deployments*
- RFC 8577, *Signaling RSVP-TE Tunnels on a Shared MPLS Forwarding Plane*
- draft-ietf-mpls-ri-rsvp-frr-05, *Refresh Interval Independent FRR Facility Protection*

The following RFCs do not define standards, but provide information about RSVP and related technologies. The IETF classifies them variously as “Experimental” or “Informational.”

- RFC 2209, *Resource ReSerVation Protocol (RSVP)—Version 1 Message Processing Rules*
- RFC 2216, *Network Element Service Specification Template*
- RFC 4125, *Maximum Allocation Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering*
- RFC 4127, *Russian Dolls Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering*
- RFC 8577, *Signaling RSVP-TE Tunnels on a Shared MPLS Forwarding Plane (Fully compliant)*

Supported LDP Standards

Junos OS substantially supports the following RFCs and Internet drafts, which define standards for LDP.

- RFC 3212, *Constraint-Based LSP Setup using LDP*
- RFC 3478, *Graceful Restart Mechanism for Label Distribution Protocol*
- RFC 7060, *Using LDP Multipoint Extensions on Targeted LDP Sessions*
- RFC 8661, *Segment Routing MPLS Interworking with LDP*
- RFC 8077, *Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)*
- Internet draft draft-napierala-mpls-targeted-mldp-01.txt, *Using LDP Multipoint Extensions on Targeted LDP Sessions*

The following RFCs do not define standards, but provide information about LDP. The IETF classifies them as “Informational.”

- RFC 3215, *LDP State Machine*
- RFC 5036, *LDP Specification*

  For the following features described in the indicated sections of the RFC, Junos OS supports one of the possible modes but not the others:

  - Label distribution control (section 2.6.1): Ordered mode is supported, but not Independent mode.
  - Label retention (section 2.6.2): Liberal mode is supported, but not Conservative mode.
  - Label advertisement (section 2.6.3): Both Downstream Unsolicited mode and Downstream on Demand mode are supported.
- RFC 5283, *LDP Extension for Inter-Area Label Switched Paths (LSPs)*
- RFC 5443, *LDP IGP Synchronization*
- RFC 5561, *LDP Capabilities*
- RFC 6512, *Using Multipoint LDP When the Backbone Has No Route to the Root*

  Only the Recursive Opaque Value is supported.
- RFC 6826, *Multipoint LDP In-Band Signaling for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths*

  Junos OS support limited to point-to-multipoint extensions for LDP.

DiffServ-Aware Traffic Engineering Standards

The following RFCs provide information on DiffServ-aware traffic engineering and multiclass LSPs:

- RFC 3270, *Multi-Protocol Label Switching (MPLS) Support of Differentiated Services*
- RFC 3564, *Requirements for Support of Differentiated Services-aware MPLS Traffic Engineering*
- RFC 4124, *Protocol Extensions for Support of Differentiated-Service-Aware MPLS Traffic Engineering*
- RFC 4125, *Maximum Allocation Bandwidth Constraints Model for Diff-Serv-aware MPLS Traffic Engineering*
- RFC 4127, *Russian Dolls Bandwidth Constraints Model for Diff-Serv-aware MPLS*

These RFCs are available on the IETF website at http://www.ietf.org/.

Supported GMPLS Standards

Junos OS substantially supports the following RFCs and Internet drafts, which define standards for Generalized MPLS (GMPLS).

- RFC 3471, *Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description*

  Only the following features are supported:

  - Bidirectional LSPs (upstream label only)
  - Control channel separation
  - Generalized label (suggested label only)
  - Generalized label request (bandwidth encoding only)

- RFC 3473, *Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions*

  Only Section 9, “Fault Handling,” is supported.

- RFC 4202, *Routing Extensions in Support of Generalized Multi-Protocol Label Switching*

  Only interface switching is supported.

- RFC 4206, *Label Switched Paths (LSP) Hierarchy with Generalized Multi-Protocol Label Switching (GMPLS) Traffic Engineering (TE)*

- Internet draft draft-ietf-ccamp-gmpls-rsvp-te-ason-02.txt, *Generalized MPLS (GMPLS) RSVP-TE Signalling in support of Automatically Switched Optical Network (ASON)* (expires January 2005)

- Internet draft draft-ietf-ccamp-gmpls-sonet-sdh-08.txt, *Generalized Multi-Protocol Label Switching Extensions for SONET and SDH Control*

  Only S,U,K,L,M-format labels and SONET traffic parameters are supported.

- Internet draft draft-ietf-ccamp-lmp-10.txt, *Link Management Protocol (LMP)*

- Internet draft draft-ietf-ccamp-ospf-gmpls-extensions-12.txt, *OSPF Extensions in Support of Generalized Multi-Protocol Label Switching*

  The following sub-TLV types for the Link type, link, value (TLV) are not supported:

  - Link Local/Remote Identifiers (type 11)
  - Link Protection Type (type 14)
  - Shared Risk Link Group (SRLG) (type 16)

The features described in Section 2 of the draft, “Implications on Graceful Restart,” are also not supported.

The Interface Switching Capability Descriptor (type 15) sub-TLV type is implemented, but only for packet switching.

- Internet draft draft-ietf-mpls-bundle-04.txt, *Link Bundling in MPLS Traffic Engineering*

Supported PCEP Standards

Junos OS substantially supports the following RFCs and Internet drafts, which define standards for PCEP.

- RFC 5440, *Path Computation Element (PCE) Communication Protocol (PCEP)—Stateful PCE*
- RFC 8231, *Path Computation Element Communication Protocol (PCEP)—Extensions for Stateful PCE*
- RFC 8281, *Path Computation Element Communication Protocol (PCEP)—Extensions PCE-Initiated LSP Setup in a Stateful PCE Model*
- Internet draft-ietf-pce-stateful-pce-07.txt, *PCEP Extensions for Stateful PCE*
- Internet draft-crabbe-pce-pce-initiated-lsp-03.txt, *PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model*
- Internet draft-ietf-pce-segment-routing-06.txt, *PCEP Extensions for Segment Routing*
- Internet draft-ietf-pce-stateful-pce-p2mp-02.txt, *Path Computation Element (PCE) Protocol Extensions for Stateful PCE usage for Point-to-Multipoint Traffic Engineering Label Switched Paths*
- Internet draft draft-cbrt-pce-stateful-local-protection-01, *PCEP Extensions for RSVP-TE Local-Protection with PCE-Stateful* (excluding support for bypass LSP mapping)
- Internet draft draft-ietf-pce-pcep-flowspec-05, *PCEP Extension for Flow Specification*

  The current implementation of this feature does not implement the following sections of the draft:

  - Section 3.1.2—Advertising PCE capabilties in IGP
  - Section 3.2—PCReq and PCRep message
  - Section 7—Most of the flow specifications, except route distinguisher and IPv4 Multicast Flow specifications, are not supported.

RELATED DOCUMENTATION

Accessing Standards Documents on the Internet

2

PART

MPLS Configuration

- Configuring MPLS | **46**
- Configuring MPLS Tunnels | **99**

CHAPTER 3

Configuring MPLS

IN THIS CHAPTER

- Basic MPLS Configuration | 46
- MPLS on Provider and Provider Edge Devices Configuration | 76
- MPLS Configuration on IRB Interfaces | 95

Basic MPLS Configuration

IN THIS SECTION

- MPLS Configuration Overview | 46
- MPLS Configuration Guidelines | 47
- Configuring MPLS | 48
- Example: Enabling MPLS | 48
- Example: Configuring MPLS on EX8200 and EX4500 Switches | 52

MPLS Configuration Overview

When you first install Junos OS on your device, MPLS is disabled by default. You must explicitly configure your device to allow MPLS traffic to pass through. Complete the following steps for all devices in your MPLS network that are running Junos OS.

To enable MPLS:

1. Delete all configured security services from the device. If you do not complete this step, you will get a commit failure. See Example: Deleting Security Services.
2. Enable MPLS on the device. See "Example: Enabling MPLS" on page 48.
3. Commit the configuration.

4. Reboot the device.
5. Configure MPLS features such as traffic engineering, VPNs, and VPLS. See:
   - "MPLS Traffic Engineering and Signaling Protocols Overview" on page 1702
   - MPLS VPN Overview
   - CLNS Overview
   - VPLS Overview

> **CAUTION**: When packet forwarding mode is changed to MPLS, all flow-based security features are deactivated, and the device performs packet-based processing only. Flow-based services such as security policies, zones, NAT, ALGs, chassis clustering, screens, firewall authentication, and IPsec VPNs are unavailable on the device. However, MPLS can be enabled in flow-based packet forwarding mode for selected traffic using firewall filters.

MPLS Configuration Guidelines

When configuring MPLS on QFX Series devices or on EX4600, note that the number of IP prefixes supported depends on the specific platform being used. See the scale specifications in the data sheet of your device for additional information.

- We recommend the following:
  - If your ingress provider edge (PE) switch needs to support more than 8000 external IP prefixes, use a larger capacity device as an ingress PE switch.
  - If you use a switch as a route reflector for BGP labeled routes, use it as a dedicated route reflector (that is, the switch must not participate in managing data traffic).
  - If you use a switch as a PE switch or as a route reflector for BGP labeled routes, configure routing policies on the PE switch and the route reflector to filter external IP routes from the routing table.

    The configuration example for a routing policy named fib_policy (at the `[edit policy-options` and `[edit routing-options` hierarchy levels) to filter BGP labeled routes from the inet.0 routing table is given below:

```
user@switch# show policy-options
policy-statement fib_policy {
    from {
        protocol bgp;
        rib inet.0;
    }
```

```
    then reject;
}
```

```
user@switch# show routing-options
forwarding-table {
    export fib_policy;
}
```

- Packet fragmentation using the `allow-fragmentation` statement at the `[edit protocols mpls path-mtu]` hierarchy level is not supported on QFX Series devices or on the EX4600 switch. Therefore, you must ensure that the maximum transmission unit (MTU) values configured on every MPLS interface is sufficient to handle MPLS packets. The packets whose size exceeds the MTU value of an interface will be dropped.

Configuring MPLS

You must also configure MPLS for a Layer 2 cross-connect to work. The following is a minimal MPLS configuration:

```
[edit]
interfaces {
                                    interface-name              {
        unit                logical-unit-number;
    }
}
protocols {
    mpls {
        interface all;
    }
}
```

Example: Enabling MPLS

IN THIS SECTION

- Requirements | 49
- Overview | 49

Configuration | 49
Verification | 51

This example shows how to enable MPLS for packet-based processing. It also shows how to enable the MPLS family and MPLS process on all of the transit interfaces in the network.

Requirements

Before you begin, delete all configured security services. See Example: Deleting Security Services.

Overview

The instructions in this topic describe how to enable MPLS on the device. You must enable MPLS on the device before including a device running Junos OS in an MPLS network.

Configuration

IN THIS SECTION

Procedure | 49

Procedure

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-1/0/0 unit 0 family mpls
set protocols mpls ge-1/0/0 unit 0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To enable MPLS:

1. Enable the MPLS family on each transit interface that you want to include in the MPLS network.

```
[edit interfaces]
user@host# set interfaces ge-1/0/0 unit 0 family mpls
```

2. Enable the MPLS process on all of the transit interfaces in the MPLS network.

```
[edit protocols mpls]
user@host# set interface ge-1/0/0 unit 0
```

3. Additionally, for security devices, enable MPLS for **packet-based** processing. Skip this step for routing and switching devices.

```
[edit security forwarding-options]
user@host# set family mpls mode packet-based
```

NOTE: When MPLS is enabled, all flow-based security features are deactivated and the device performs packet-based processing. Flow-based services such as security policies, zones, NAT, ALGs, chassis clustering, screens, firewall authentication, IP packets, and IPsec VPNs are unavailable on the device.

Before changing from flow mode to packet mode, you must remove all security policies remaining under flow mode. To prevent management connection loss, you must bind the management interface to zones and enable host-inbound traffic to prevent the device from losing connectivity.

For information about configuring zones, see Security Policies User Guide for Security Devices.

Results

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying MPLS Is Enabled at the Protocols Level | 51
- Verifying MPLS Is Enabled at the Interfaces Level | 51
- Verifying Packet-based Processing Is Enabled | 51

Confirm that the configuration is working properly.

Verifying MPLS Is Enabled at the Protocols Level

Purpose

Verify that MPLS is enabled at the protocols level.

Action

From operational mode, enter the **show protocols** command.

Verifying MPLS Is Enabled at the Interfaces Level

Purpose

Verify that MPLS is enabled at the interfaces level.

Action

From operational mode, enter the **show interfaces** command.

Verifying Packet-based Processing Is Enabled

Purpose

Specific to security devices, verify that packet-based processing is enabled.

Action

From operational mode, enter the **show security forwarding-options** command.

```
user@host> show security forwarding-options
family {
    mpls {
        mode packet-based;
    }
}
```

NOTE: If you enable MPLS for packet-based processing by using the command `set security forward-option family mpls mode packet`, the mode will not change immediately and the system will display the following messages:

warning: Reboot may required when try reset flow inet mode

warning: Reboot may required when try reset mpls flow mode please check security flow status for detail.

You need to reboot your device for the configuration to take effect.

CAUTION: If you disable MPLS and switch back to using the security services (flow-based processing), the mode will not change immediately and the system will display warning messages instructing you to restart your device. You must reboot your device for the configuration to take effect. This will also result in management sessions being reset and transit traffic getting interrupted.

Example: Configuring MPLS on EX8200 and EX4500 Switches

IN THIS SECTION

- Requirements | **53**
- Overview and Topology | **53**
- Configuring the Local PE Switch | **59**
- Configuring the Remote PE Switch | **63**
- Configuring the Provider Switch | **67**
- Verification | **71**

You can configure MPLS on switches to increase transport efficiency in your network. MPLS services can be used to connect various sites to a backbone network and to ensure better performance for low-latency applications such as voice over IP (VoIP) and other business-critical functions.

To implement MPLS on the switches, you must configure two provider edge (PE) switches—an ingress PE switch and an egress PE switch— and at least one provider (transit) switch. You can configure the customer edge (CE) interfaces on the PE switches of the MPLS network as either circuit cross-connect (CCC) or IP (**family inet**) interfaces.

This example shows how to configure an MPLS tunnel using a simple interface as a CCC:

NOTE: This example shows how to configure MPLS using a simple interface as a CCC. For information on configuring a tagged VLAN interface as a CCC, see "Configuring an MPLS-Based VLAN CCC Using a Layer 2 VPN (CLI Procedure)" on page 2096 or "Configuring an MPLS-Based VLAN CCC Using a Layer 2 Circuit" on page 2060.

Requirements

This example uses the following hardware and software components:

- Junos OS Release 10.1 or later for switches
- Three EX Series switches

Before you begin configuring MPLS, ensure that you have configured the routing protocol (OSPF or IS-IS) on the core interface and the loopback interface on all the switches. This example includes the configuration of OSPF on all the switches. For information on configuring IS-IS as the routing protocol, see the *Junos OS Routing Protocols Configuration Guide*.

Overview and Topology

This example includes an ingress or local PE switch, an egress or remote PE switch, and one provider switch. It includes CCCs that tie the customer edge interface of the local PE switch (PE-1) to the customer edge interface of the remote PE switch (PE-2). It also describes how to configure the core interfaces of the PE switches and the provider switch to support the transmission of the MPLS packets. In this example, the core interfaces that connect the local PE switch and the provider switch are individual interfaces, while the core interfaces that connect the remote PE switch and the provider switch are aggregated Ethernet interfaces.

NOTE:

- Core interfaces cannot be tagged VLAN interfaces.
- Core interfaces can be aggregated Ethernet interfaces. This example includes a LAG between the provider switch and the remote PE switch because this type of configuration is another option you can implement. For information on configuring LAGs, see Configuring Aggregated Ethernet Links (CLI Procedure).

Figure 2 on page 54 shows the topology used in this example.

Figure 2: Configuring MPLS on EX Series Switches

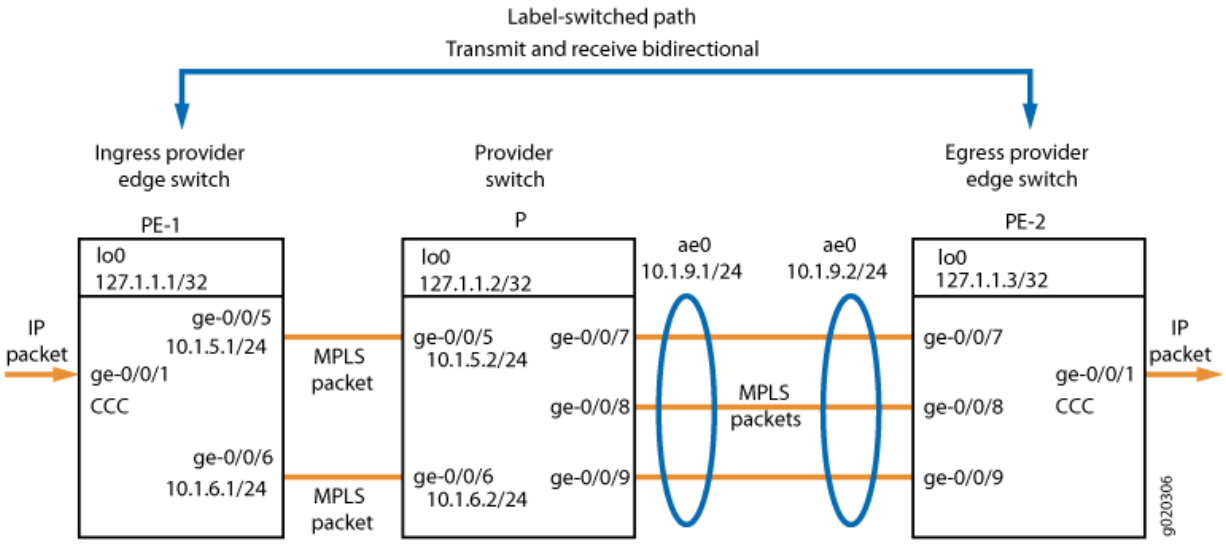


Table 4 on page 54 shows the MPLS configuration components used for the ingress PE switch in this example.

Table 4: Components of the Ingress PE Switch in the Topology for MPLS with Interface-Based CCC

Property	Settings	Description
Local PE switch hardware	EX Series switch	PE-1
Loopback address	**lo0 127.1.1.1/32**	Identifies PE-1 for interswitch communications.

Table 4: Components of the Ingress PE Switch in the Topology for MPLS with Interface-Based CCC *(Continued)*

Property	Settings	Description
Routing protocol	**ospf traffic-engineering**	Indicates that this switch is using OSPF as the routing protocol and that traffic engineering is enabled.
MPLS protocol and definition of label-switched path	**mpls** **label-switched-path lsp_to_pe2_ge1 to 127.1.13**	Indicates that this PE switch is using the MPLS protocol with the specified LSP to reach the other PE switch (specified by the loopback address). The statement must also specify the core interfaces to be used for MPLS traffic.
RSVP	**rsvp**	Indicates that this switch is using RSVP. The statement must specify the loopback address and the core interfaces that will be used for the RSVP session.
Interface family	**family inet** **family mpls** **family ccc**	The logical units of the core interfaces are configured to belong to both **family inet** and **family mpls**. The logical unit of the customer edge interface is configured to belong to **family ccc**.
Customer edge interface	**ge-0/0/1**	Interface that connects this network to devices outside the network.
Core interfaces	**ge-0/0/5.0** and **ge-0/0/6.0** with IP addresses **10.1.5.1/24** and **10.1.6.1/24**	Interfaces that connect to other switches within the MPLS network.

Table 4: Components of the Ingress PE Switch in the Topology for MPLS with Interface-Based CCC *(Continued)*

Property	Settings	Description
CCC definition	**connections** **remote-interface-switch ge-1-to-pe2** **interface ge-0/0/1.0** **transmit-lsp lsp_to_pe2_ge1** **receive-lsp lsp_to_pe1_ge1**	Associates the circuit cross-connect (CCC), **ge-0/0/1**, with the LSPs that have been defined on the local and remote PE switches.

Table 5 on page 56 shows the MPLS configuration components used for the egress PE switch in this example.

Table 5: Components of the Egress PE Switch in the Topology for MPLS with Interface-Based CCC

Property	Settings	Description
Remote PE switch hardware	EX Series switch	PE-2
Loopback address	**lo0 127.1.1.3/32**	Identifies PE-2 for interswitch communications.
Routing protocol	**ospf traffic-engineering**	Indicates that this switch is using OSPF as the routing protocol and that traffic engineering is enabled.
MPLS protocol and definition of label-switched path	**mpls** **label-switched-path lsp_to_pe1_ge1** **to 127.1.1.1**	Indicates that this PE switch is using the MPLS protocol with the specified label-switched path (LSP) to reach the other PE switch. The statement must also specify the core interfaces to be used for MPLS traffic.

Table 5: Components of the Egress PE Switch in the Topology for MPLS with Interface-Based CCC *(Continued)*

Property	Settings	Description
RSVP	**rsvp**	Indicates that this switch is using RSVP. The statement must specify the loopback address and the core interfaces that will be used for the RSVP session.
Interface family	**family inet** **family mpls** **family ccc**	The logical unit of the core interface is configured to belong to both **family inet** and **family mpls**. The logical unit of the customer edge interface is configured to belong to **family ccc**.
Customer edge interface	**ge-0/0/1**	Interface that connects this network to devices outside the network.
Core interface	**ae0** with IP address **10.1.9.2/24**	Aggregated Ethernet interface on PE-2 that connects to aggregated Ethernet interface **ae0** of the provider switch and belongs to **family mpls**.
CCC definition	**connections remote-interface-switch ge-1-to-pe1** **interface ge-0/0/1.0** **transmit-lsp lsp_to_pe1_ge1;** **receive-lsp lsp_to_pe2_ge1;**	Associates the CCC, **ge-0/0/1**, with the LSPs that have been defined on the local and remote PE switches.

Table 6 on page 58 shows the MPLS configuration components used for the provider switch in this example.

Table 6: Components of the Provider Switch in the Topology for MPLS with Interface-Based CCC

Property	Settings	Description
Provider switch hardware	EX Series switch	Transit switch within the MPLS network configuration.
Loopback address	**lo0 127.1.1.2/32**	Identifies provider switch for interswitch communications.
Routing protocol	**ospf traffic-engineering**	Indicates that this switch is using OSPF as the routing protocol and that traffic engineering is enabled.
MPLS protocol	**mpls**	Indicates that this switch is using the MPLS protocol. The statement must specify the core interfaces that will be used for MPLS traffic.
RSVP	**rsvp**	Indicates that this switch is using RSVP. The statement must specify the loopback and the core interfaces that will be used for the RSVP session.
Interface family	**family inet** **family mpls**	The logical units for the loopback interface and the core interfaces belong to **family inet**. The logical units of the core interfaces are also configured to belong to **family mpls**.
Core interfaces	**ge-0/0/5.0** and **ge-0/0/6.0** with IP addresses **10.1.5.1/24** and **10.1.6.1/24** and **ae0** with IP address **10.1.9.1/24**	Interfaces that connect the provider switch (P) to PE-1. Aggregated Ethernet interface on P that connects to aggregated Ethernet interface **ae0** of PE-2.

Configuring the Local PE Switch

IN THIS SECTION

- Procedure | 59

Procedure

CLI Quick Configuration

To quickly configure the local ingress PE switch, copy the following commands and paste them into the switch terminal window of PE-1:

```
[edit]
                        set protocols ospf traffic-engineering
                        set protocols ospf area 0.0.0.0 interface lo0.0
                        set protocols ospf area 0.0.0.0 interface ge-0/0/5.0
                        set protocols ospf area 0.0.0.0 interface ge-0/0/6.0
                        set protocols mpls label-switched-path lsp_to_pe2_ge1 to 127.1.1.3
                        set protocols mpls interface ge-0/0/5.0
                        set protocols mpls interface ge-0/0/6.0
                        set protocols rsvp interface lo0.0
                        set protocols rsvp interface ge-0/0/5.0
                        set protocols rsvp interface ge-0/0/6.0
                        set interfaces lo0 unit 0 family inet address 127.1.1.1/32
                        set interfaces ge-0/0/5 unit 0 family inet address 10.1.5.1/24
                        set interfaces ge-0/0/6 unit 0 family inet address 10.1.6.1/24
                        set interfaces ge-0/0/5 unit 0 family mpls
                        set interfaces ge-0/0/6 unit 0 family mpls
                        set interfaces ge-0/0/1 unit 0 family ccc
                        set protocols connections remote-interface-switch ge-1-to-pe2 interface
ge-0/0/1.0
                        set protocols connections remote-interface-switch ge-1-to-pe2 transmit-
lsp lsp_to_pe2_ge1
                        set protocols connections remote-interface-switch ge-1-to-pe2 receive-
lsp lsp_to_pe1_ge1
```

Step-by-Step Procedure

To configure the local ingress PE switch:

1. Configure OSPF with traffic engineering enabled:

```
[edit protocols]
user@switchPE-1# set ospf traffic-
engineering
```

2. Configure OSPF on the loopback address and the core interfaces:

```
[edit protocols]
user@switchPE-1# set ospf area 0.0.0.0 interface lo0.0
user@switchPE-1# set ospf area 0.0.0.0 interface ge-0/0/5.0
user@switchPE-1# set ospf area 0.0.0.0 interface ge-0/0/6.0
```

3. Configure MPLS on this PE switch (PE-1) with a label-switched path (LSP) to the other PE switch (PE-2):

```
[edit protocols]
user@switchPE-1# set mpls                                    label-switched-path lsp_to_pe2_ge1
to 127.1.1.3
```

4. Configure MPLS on the core interfaces:

```
[edit protocols]
user@switchPE-1# set mpls interface ge-0/0/5.0
user@switchPE-1# set mpls interface ge-0/0/6.0
```

5. Configure RSVP on the loopback interface and the core interfaces:

```
[edit protocols]
user@switchPE-1# set rsvp interface lo0.0
user@switchPE-1# set rsvp interface ge-0/0/5.0
```

```
user@switchPE-1# set rsvp interface ge-0/0/6.0
```

6. Configure IP addresses for the loopback interface and the core interfaces:

```
[edit]
user@switchPE-1# set interfaces lo0 unit 0 family inet address 127.1.1.1/32
user@switchPE-1# set interfaces ge-0/0/5 unit 0 family inet address 10.1.5.1/24
user@switchPE-1# set interfaces ge-0/0/6 unit 0 family inet address
10.1.6.1/24
```

7. Configure **family mpls** on the logical unit of the core interface addresses:

```
[edit]
user@switchPE-1# set interfaces ge-0/0/5 unit 0 family mpls
user@switchPE-1# set interfaces ge-0/0/6 unit 0 family mpls
```

8. Configure the logical unit of the customer edge interface as a CCC:

```
[edit interfaces ge-0/0/1 unit 0]
-user@PE-1# set family ccc
```

9. Configure the interface-based CCC from PE-1 to PE-2:

NOTE: You can also configure a tagged VLAN interface as a CCC. See "Configuring an MPLS-Based VLAN CCC Using a Layer 2 VPN (CLI Procedure)" on page 2096 or "Configuring an MPLS-Based VLAN CCC Using a Layer 2 Circuit" on page 2060.

```
[edit protocols]
user@PE-1# set connections                                    remote-interface-switch ge-1-to-
pe2 interface ge-0/0/1.0
user@PE-1# set connections remote-interface-switch ge-1-to-pe2 transmit-lsp lsp_to_pe2_ge1
user@PE-1# set connections remote-interface-switch ge-1-to-pe2 receive-lsp
lsp_to_pe1_ge1
```

Results

Display the results of the configuration:

```
user@switchPE-1>                                show
configuration
```

```
interfaces {
    ge-0/0/1 {
        unit 0 {
            family ccc;
        }
    }
    ge-0/0/5 {
        unit 0 {
            family inet {
                address 10.1.5.1/24;
            }
            family mpls;
        }
    }
    ge-0/0/6 {
        unit 0 {
            family inet {
                address 10.1.6.1/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 127.1.1.1/32;
            }
        }
    }
protocols {
    rsvp {
        interface lo0.0;
        interface ge-0/0/5.0;
        interface ge-0/0/6.0;
```

```
    }
    mpls {
        label-switched-path lsp_to_pe2_ge1 {
            to 127.1.1.3;
        }
        interface ge-0/0/5.0;
        interface ge-0/0/6.0;
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface lo0.0;
            interface ge-0/0/5.0;
            interface ge-0/0/6.0;
        }
    }
    connections {
        remote-interface-switch ge-1-to-pe2 {
            interface ge-0/0/1.0;
            transmit-lsp lsp_to_pe2_ge1;
            receive-lsp lsp_to_pe1_ge1;
        }
    }
```

Configuring the Remote PE Switch

IN THIS SECTION

- Procedure | 64

Procedure

CLI Quick Configuration

To quickly configure the remote PE switch, copy the following commands and paste them into the switch terminal window of PE-2:

```
[edit]
                        set protocols ospf traffic-engineering
                        set protocols ospf area 0.0.0.0 interface lo0.0
                        set protocols ospf area 0.0.0.0 interface ae0
                        set protocols mpls label-switched-path lsp_to_pe1_ge1 to 127.1.1.1
                        set protocols mpls interface ae0
                        set protocols rsvp interface lo0.0
                        set protocols rsvp interface ae0
                        set interfaces lo0 unit 0 family inet address 127.1.1.3/32
                        set interfaces ae0 unit 0 family inet address 10.1.9.2/24
                        set interfaces ae0 unit 0 family mpls
                        set interfaces ge-0/0/1 unit 0 family ccc
                        set protocols connections remote-interface-switch ge-1-to-pe1 interface
ge-0/0/1.0
                        set protocols connections remote-interface-switch ge-1-to-pe1 transmit-
lsp lsp_to_pe1_ge1
                        set protocols connections remote-interface-switch ge-1-to-pe1 receive-
lsp lsp_to_pe2_ge1
```

Step-by-Step Procedure

To configure the remote PE switch (PE-2):

1. Configure OSPF with traffic engineering enabled:

```
[edit protocols]
user@switchPE-2# set ospf traffic-engineering
```

2. Configure OSPF on the loopback interface and the core interface:

```
[edit protocols]
user@switchPE-2# set ospf area 0.0.0.0 interface lo0.0
```

```
user@switchPE-2# set ospf area 0.0.0.0 interface ae0
```

3. Configure MPLS on this switch (PE-2) with a label-switched path (LSP) to the other PE switch (PE-1):

```
[edit protocols]
user@switchPE-2# set mpls label-switched-path lsp_to_pe1_ge1 to 127.1.1.1
```

4. Configure MPLS on the core interface:

```
[edit protocols]
user@switchPE-2# set mpls interface ae0
```

5. Configure RSVP on the loopback interface and the core interface:

```
[edit protocols]
ser@switchPE-2# set rsvp interface lo0.0
user@switchPE-2# set rsvp interface ae0
```

6. Configure IP addresses for the loopback interface and the core interface:

```
[edit]
user@switchPE-2# set interfaces lo0 unit 0 family inet address 127.1.1.3/32
user@switchPE-2# set interfaces ae0 unit 0 family inet address
10.1.9.2/24
```

7. Configure **family mpls** on the logical unit of the core interface:

```
[edit]
user@switchPE-2# set interfaces ae0 unit 0 family mpls
```

8. Configure the logical unit of the customer edge interface as a CCC:

```
[edit interfaces ge-0/0/1 unit 0]
user@PE-2# set family ccc
```

9. Configure the interface-based CCC from PE-2 to PE-1:

```
[edit protocols]
user@PE-2# set connections remote-interface-switch ge-1-to-pe1 interface ge-0/0/1.0
user@PE-2# set connections remote-interface-switch ge-1-to-pe1 transmit-lsp lsp_to_pe1_ge1
user@PE-2# set connections remote-interface-switch ge-1-to-pe1 receive-lsp
lsp_to_pe2_ge1
```

Results

Display the results of the configuration:

```
user@switchPE-2>                                show
configuration
```

```
interfaces {
    ge-0/0/1 {
        unit 0 {
            family ccc;
        }
    }
    ae0 {
        unit 0 {
            family inet {
                address 10.1.9.2/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 127.1.1.3/32;
```

```
            }
        }
    }
}
protocols {
    rsvp {
        interface lo0.0;
        interface ae0.0;
    }
    mpls {
        label-switched-path lsp_to_pe1_ge1 {
            to 127.1.1.1;
        }
        interface ae0.0;
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface ae0.0;
        }
    }
    connections {
        remote-interface-switch ge-1-to-pe1 {
            interface ge-0/0/1.0;
            transmit-lsp lsp_to_pe1_ge1;
            receive-lsp lsp_to_pe2_ge1;
        }
    }
}
```

Configuring the Provider Switch

IN THIS SECTION

- Procedure | 68

Procedure

CLI Quick Configuration

To quickly configure the provider switch, copy the following commands and paste them into the switch terminal window:

```
[edit]
                    set  protocols ospf traffic-engineering
                    set protocols ospf area 0.0.0.0 interface lo0.0
                    set protocols ospf area 0.0.0.0  interface ge-0/0/5.0
                    set protocols ospf area 0.0.0.0  interface ge-0/0/6.0
                    set protocols ospf area 0.0.0.0  interface ae0
                    set protocols mpls interface ge-0/0/5.0
                    set protocols mpls interface ge-0/0/6.0
                    set protocols mpls interface ae0
                    set protocols rsvp interface lo0.0
                    set protocols rsvp interface ge-0/0/5.0
                    set protocols rsvp interface ge-0/0/6.0
                    set protocols rsvp interface ae0
                    set interfaces lo0 unit 0 family inet address 127.1.1.2/32
                    set interfaces ge-0/0/5 unit 0 family inet address 10.1.5.1/24
                    set interfaces ge-0/0/6 unit 0 family inet address 10.1.6.1/24
                    set interfaces ae0 unit 0 family inet address 10.1.9.1/24
                    set interfaces ge-0/0/5 unit 0 family mpls
                    set interfaces ge-0/0/6 unit 0 family mpls
                    set interfaces ae0 unit 0 family mpls
```

Step-by-Step Procedure

To configure the provider switch:

1. Configure OSPF with traffic engineering enabled:

```
[edit protocols]
user@switchP# set ospf traffic-engineering
```

2. Configure OSPF on the loopback interface and the core interfaces:

```
[edit protocols]
user@switchP# set ospf area 0.0.0.0 interface lo0.0
user@switchP# set ospf area 0.0.0.0 interface ge-0/0/5
user@switchP# set ospf area 0.0.0.0 interface ge-0/0/6
user@switchP# set ospf area 0.0.0.0 interface ae0
```

3. Configure MPLS on the core interfaces on the switch:

```
[edit protocols]
user@switchP# set mpls interface ge-0/0/5
user@switchP# set mpls interface ge-0/0/6
user@switchP# set mpls interface ae0
```

4. Configure RSVP on the loopback interface and the core interfaces:

```
[edit protocols]
user@switchP# set rsvp interface lo0.0
user@switchP# set rsvp interface ge-0/0/5
user@switchP# set rsvp interface ge-0/0/6
user@switchP# set rsvp interface ae0
```

5. Configure IP addresses for the loopback interface and the core interfaces:

```
[edit]
user@switchP# set interfaces lo0 unit 0 family inet address 127.1.1.2/32
user@switchP# set interfaces ge-0/0/5 unit 0 family inet address 10.1.5.1/24
user@switchP# set interfaces ge-0/0/6 unit 0 family inet address 10.1.6.1/24
user@switchP# set interfaces ae0 unit 0 family inet address 10.1.9.1/24
```

6. Configure **family mpls** on the logical unit of the core interface addresses:

```
[edit]
user@switchP# set interfaces ge-0/0/5 unit 0 family mpls
```

```
user@switchP# set interfaces ge-0/0/6 unit 0 family mpls
user@switchP# set interfaces ae0 unit 0 family mpls
```

Results

Display the results of the configuration:

```
user@switchP>                          show
configuration
```

```
interfaces {
    ge-0/0/5 {
        unit 0 {
            family inet {
                address 10.1.5.1/24;
            }
            family mpls;
            }
        }
    ge-0/0/6 {
        unit 0 {
            family inet {
                address 10.1.6.1/24;
            }
            family mpls;
            }
        }
    }
    ae0 {
        unit 0 {
            family inet {
                address 10.1.9.1/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 127.1.1.2/32;
```

```
            }
        }
    }
protocols {
    rsvp {
        interface lo0.0;
        interface ge-0/0/5.0;
        interface ge-0/0/6.0;
        interface ae0.0;
    }
    mpls {
        interface ge-0/0/5.0;
        interface ge-0/0/6.0;
        interface ae0.0;
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface lo0.0;
            interface ge-0/0/5.0;
            interface ge-0/0/6.0;
            interface ae0.0;
        }
    }
```

Verification

IN THIS SECTION

- Verifying the Physical Layer on the Switches | **72**
- Verifying the Routing Protocol | **73**
- Verifying the Core Interfaces Being Used for MPLS Traffic | **73**
- Verifying the Status of the RSVP Sessions | **74**
- Verifying the Assignment of Interfaces for MPLS Label Operations | **74**
- Verifying the Status of the CCC | **75**

To confirm that the configuration is working properly, perform these tasks:

Verifying the Physical Layer on the Switches

Purpose

Verify that the interfaces are up. Perform this verification task on each of the switches.

Action

```
user@switchPE-1> show interfaces terse

Interface               Admin Link Proto    Local                 Remote
ge-0/0/0                up    up
ge-0/0/0.0              up    up  eth-switch
ge-0/0/1                up    up
ge-0/0/1.0              up    up  ccc
ge-0/0/2                up    up
ge-0/0/2.0              up    up  eth-switch
ge-0/0/3                up    up
ge-0/0/3.0              up    up eth-switch
ge-0/0/4                up    up
ge-0/0/4.0              up    up eth-switch
ge-0/0/5                up    up
ge-0/0/5.0              up    up inet     10.1.5.1/24
                                 mpls
ge-0/0/6                up    up
ge-0/0/6.0              up    up inet     10.1.6.1/24
                                 mpls
```

Meaning

The `show interfaces terse` command displays status information about the Gigabit Ethernet interfaces on the switch. This output verifies that the interfaces are **up**. The output for the protocol family (**Proto** column) shows that interface **ge-0/0/1.0** is configured as a circuit cross-connect. The output for the protocol family of the core interfaces (**ge-0/0/5.0** and **ge-0/0/6.0**) shows that these interfaces are configured as both **inet** and **mpls**. The **Local** column for the core interfaces shows the IP address configured for these interfaces.

Verifying the Routing Protocol

Purpose

Verify the state of the configured routing protocol. Perform this verification task on each of the switches. The state must be **Full**.

Action

```
user@switchPE-1> show ospf neighbor

Address          Interface              State     ID               Pri  Dead
127.1.1.2          ge-0/0/5              Full     10.10.10.10      128    39
```

Meaning

The `show ospf neighbor` command displays the status of the routing protocol. This output shows that the state is **Full**, meaning that the routing protocol is operating correctly—that is, hello packets are being exchanged between directly connected neighbors.

Verifying the Core Interfaces Being Used for MPLS Traffic

Purpose

Verify that the state of the MPLS interface is **Up**. Perform this verification task on each of the switches.

Action

```
user@switchPE-1>                              show mpls
interface

Interface        State       Administrative groups
ge-0/0/5            Up          <none>
ge-0/0/6            Up          <none>
```

Meaning

The `show mpls interface` command displays the status of the core interfaces that have been configured to belong to **family mpls**. This output shows that the interface configured to belong to **family mpls** is **Up**.

Verifying the Status of the RSVP Sessions

Purpose

Verify the status of the RSVP sessions. Perform this verification task on each of the switches.

Action

```
user@switchPE-1>   show rsvp session

Ingress RSVP: 1 sessions
To              From            State   Rt Style Labelin Labelout LSPname
127.1.13         127.1.1.1         Up       0  1 FF       -   300064 lsp_to_pe2_ge1
Total 1 displayed, Up 1, Down 0

Egress RSVP: 1 sessions
To              From            State   Rt Style Labelin Labelout LSPname
127.1.1.1        127.1.1.3         Up       0  1 FF       299968     lsp_to_pe1_ge1
Total 1 displayed, Up 1, Down 0

Transit RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Meaning

This output confirms that the RSVP sessions are **Up**.

Verifying the Assignment of Interfaces for MPLS Label Operations

Purpose

Verify which interface is being used as the beginning of the CCC and which interface is being used to push the MPLS packet to the next hop. Perform this task only on the PE switches.

Action

```
user@switchPE-1>                              show route forwarding-table family
mpls
MPLS:
Destination        Type RtRef Next hop           Type Index NhRef Netif
```

```
default            perm     0                    dscd    50     1
0                  user     0                    recv    49     3
1                  user     0                    recv    49     3
2                  user     0                    recv    49     3
299776             user     0                   Pop     541     2     ge-0/0/1.0
ge-0/0/1.0  (CCC)  user     0 2.0.0.1           Push 299792   540 2   ge-0/0/5.0
```

Meaning

This output shows that the CCC has been set up on interface **ge-0/0/1.0**. The switch receives ingress traffic on **ge-0/0/1.0** and pushes label **299792** onto the packet, which goes out through interface **ge-0/0/5.0**. The output also shows when the switch receives an MPLS packet with label 29976, it pops the label and sends the packet out through interface **ge-0/0/1.0**

After you have checked the local PE switch, run the same command on the remote PE switch.

Verifying the Status of the CCC

Purpose

Verify the status of the CCC. Perform this task only on the PE switches.

Action

```
user@switchPE-1>                              show
connections

CCC and TCC connections [Link Monitoring On]
Legend for status (St)              Legend for connection types
UN -- uninitialized                 if-sw:  interface switching
NP -- not present                   rmt-if: remote interface switching
WE -- wrong encapsulation           lsp-sw: LSP switching
DS -- disabled                      tx-p2mp-sw: transmit P2MP switching
Dn -- down                          rx-p2mp-sw: receive P2MP switching
-> -- only outbound conn is up
 <- -- only inbound  conn is up      Legend for circuit types
Up -- operational                   intf -- interface
RmtDn -- remote CCC down            tlsp -- transmit LSP
Restart -- restarting               rlsp -- receive LSP
```

```
Connection/Circuit              Type        St      Time last up     # Up trans
ge1-to-pe2                      rmt-if      Up      Feb 17 05:00:09  1
  ge-0/0/1.0                     intf       Up
  lsp_to_pe1_ge1                   tlsp     Up
  lsp_to_pe2_ge1                   rlsp     Up
```

Meaning

The `show connections` command displays the status of the CCC connections. This output verifies that the CCC interface and its associated transmit and receive LSPs are **Up**. After you have checked the local PE switch, run the same command on the remote PE switch.

RELATED DOCUMENTATION

MPLS Overview | 2

MPLS on Provider and Provider Edge Devices Configuration

IN THIS SECTION

- Configuring MPLS on Provider Switches | 76
- Configuring MPLS on Provider Edge Switches | 78
- Configuring MPLS on Provider Edge Switches Using IP-Over-MPLS | 83
- Configuring MPLS on Provider Edge EX8200 and EX4500 Switches Using Circuit Cross-Connect | 89
- Configuring MPLS on EX8200 and EX4500 Provider Switches | 93

Configuring MPLS on Provider Switches

To implement MPLS, you must configure at least one provider switch as a transit switch for the MPLS packets.

MPLS requires the configuration of an interior gateway protocol (OSPF) and a signaling protocol (RSVP) on the core interfaces and the loopback interface of all the switches. This procedure includes the configuration of OSPF on the provider switch.

To configure the provider switch, complete the following tasks:

1. Configure OSPF on the loopback and core interfaces:

   NOTE: You can use the switch address as an alternative to the loopback interface.

```
[edit protocols ospf]
user@switch# set area 0.0.0.0 interface lo0.0
user@switch# set area 0.0.0.0 interface xe-0/0/5.0
user@switch# set area 0.0.0.0 interface xe-0/0/6.0
user@switch# set area 0.0.0.0 interface ae0
```

2. Configure MPLS on the core interfaces:

```
[edit protocols mpls]
user@switch# set interface xe-0/0/5.0
user@switch# set interface xe-0/0/6.0
user@switch# set interface ae0
```

3. Configure RSVP on the loopback interface and the core interfaces:

```
[edit protocols rsvp]
user@switch# set interface lo0.0
user@switch# set interface xe-0/0/5.0
user@switch# set interface xe-0/0/6.0
user@switch# set interface ae0
```

4. Configure an IP address for the loopback interface and the core interfaces:

```
[edit interfaces]
user@switch# set lo0 unit 0 family inet address 127.1.1.1/32
user@switch# set xe-0/0/5 unit 0 family inet address 10.1.5.1/24
user@switch# set xe-0/0/6 unit 0 family inet address 10.1.6.1/24
user@switch# set ae0 unit 0 family inet address 10.1.9.2/24
```

5. Configure `family mpls` on the logical units of the core interfaces, thereby identifying the interfaces that will be used for forwarding MPLS packets:

```
[edit interfaces]
user@switch# set xe-0/0/5 unit 0 family mpls
user@switch# set xe-0/0/6 unit 0 family mpls
user@switch# set ae0 unit 0 family mpls
```

Configuring MPLS on Provider Edge Switches

IN THIS SECTION

- Configuring the Ingress PE Switch | 78
- Configuring the Egress PE Switch | 80

To implement MPLS, you must configure two provider edge (PE) switches—an ingress PE switch and an egress PE switch—and at least one provider switch. You can configure the customer edge (CE) interfaces on the PE switches of the MPLS network using IP over MPLS.

This topic describes how to configure an ingress PE switch and an egress PE switch using IP over MPLS:

Configuring the Ingress PE Switch

To configure the ingress PE switch:

1. Configure an IP address for the loopback interface and the core interfaces:

```
[edit interfaces]
user@switch# set lo0 unit 0 family inet address 192.168.10.1/32
user@switch# set xe-0/0/5 unit 0 family inet address 10.1.5.1/24
user@switch# set xe-0/0/6 unit 0 family inet address 10.1.6.1/24
```

NOTE: You cannot use routed VLAN interfaces (RVIs) or Layer 3 subinterfaces as core interfaces.

2. Configure OSPF on the loopback interface and the core interfaces:

NOTE: You can use the switch address as an alternative to the loopback interface.

```
[edit protocols ospf]
user@switch# set area 0.0.0.0 interface lo0.0
user@switch# set area 0.0.0.0 interface xe-0/0/5.0
user@switch# set area 0.0.0.0 interface xe-0/0/6.0
```

3. Configure OSPF traffic engineering:

```
[edit protocols ospf]
user@switch# set traffic-engineering
```

4. Configure RSVP on the loopback interface and the core interfaces:

```
[edit protocols rsvp]
user@switch# set interface lo0.0
user@switch# set interface xe-0/0/5.0
user@switch# set interface xe-0/0/6.0
```

5. Configure MPLS traffic engineering.

```
[edit protocols mpls]
user@switch# set traffic-engineering
```

6. Configure MPLS on the core interfaces:

```
[edit protocols mpls]
user@switch# set interface xe-0/0/5.0
user@switch# set interface xe-0/0/6.0
```

7. Configure `family mpls` on the logical units of the core interfaces, thereby identifying the interfaces that will be used for forwarding MPLS packets:

```
[edit interfaces]
user@switch# set xe-0/0/5 unit 0 family mpls
user@switch# set xe-0/0/6 unit 0 family mpls
```

8. Configure a customer edge interface as a Layer 3 routed interface, specifying an IP address:

```
[edit interfaces]
user@switch# set xe-0/0/3 unit 0 family inet address 121.100.10.1/16
```

9. Configure this Layer 3 customer edge interface for the routing protocol:

```
[edit]
user@switch# set protocols ospf area 0.0.0 interface xe-0/0/3.0
```

10. Configure an LSP on the ingress PE switch (192.168.10.1) to send IP packets over MPLS to the egress PE switch (192.168.12.1):

```
[edit protocols mpls]
user@switch# set label-switched-path lsp_1 to 192.168.12.1
```

11. Disable constrained-path LSP computation for this LSP:

```
[edit protocols mpls]
user@switch# set label-switched-path lsp_1 no-cspf
```

12. Configure a static route from the ingress PE switch to the egress PE switch, thereby indicating to the routing protocol that the packets will be forwarded over the MPLS LSP that has been set up to that destination:

```
[edit routing-options]
user@switch# set static route 2.2.2.0/24 next-hop 192.168.10.1
user@switch# set static route 2.2.2.0/24 resolve
```

Configuring the Egress PE Switch

To configure the egress PE switch:

1. Configure an IP address for the loopback interface and the core interfaces:

```
[edit interfaces]
user@switch# set lo0 unit 0 family inet address 192.168.12.1/32
user@switch# set xe-0/0/5 unit 0 family inet address 10.1.20.1/24
user@switch# set xe-0/0/6 unit 0 family inet address 10.1.21.1/24
```

NOTE: You cannot use routed VLAN interfaces (RVIs) or Layer 3 subinterfaces as core interfaces.

2. Configure OSPF on the loopback interface and the core interfaces:

NOTE: You can use the switch address as an alternative to the loopback interface.

```
[edit protocols ospf]
user@switch# set area 0.0.0.0 interface lo0.0
user@switch# set area 0.0.0.0 interface xe-0/0/5.0
user@switch# set area 0.0.0.0 interface xe-0/0/6.0
```

3. Configure RSVP on the loopback interface and the core interfaces:

```
[edit protocols rsvp]
user@switch# set rsvp interface lo0.0
user@switch# set rsvp interface xe-0/0/5.0
user@switch# set rsvp interface xe-0/0/6.0
```

4. Configure MPLS on the core interfaces:

```
[edit protocols mpls]
user@switch# set interface xe-0/0/5.0
user@switch# set interface xe-0/0/6.0
```

5. Configure `family mpls` on the logical units of the core interfaces, thereby identifying the interfaces that will be used for forwarding MPLS packets:

```
[edit interfaces]
user@switch# set xe-0/0/5 unit 0 family mpls
user@switch# set xe-0/0/6 unit 0 family mpls
```

6. Configure a customer edge interface as a Layer 3 routed interface, specifying an IP address:

```
[edit interfaces]
user@switch# set xe-0/0/3 unit 0 family inet address 2.2.2.1/16
```

7. Configure this Layer 3 customer edge interface for the routing protocol:

```
[edit]
user@switch# set protocols ospf area 0.0.0 interface xe-0/0/3
```

8. Configure an LSP on the egress PE switch (192.168.12.1) to send IP packets over MPLS to the ingress PE switch (192.168.10.1):

```
[edit protocols mpls]
user@switch# set label-switched-path lsp_2 to 192.168.10.1
```

9. Disable constrained-path LSP computation for this LSP:

```
[edit protocols mpls]
user@switch# set label-switched-path lsp_2 no-cspf
```

10. Configure a static route from the ingress PE switch to the egress PE switch, thereby indicating to the routing protocol that the packets will be forwarded over the MPLS LSP that has been set up to that destination:

```
[edit routing-options]
user@switch# set static route 121.121.121.0/24 next-hop 192.168.12.1
user@switch# set static route 121.121.121.0/24 resolve
```

Configuring MPLS on Provider Edge Switches Using IP-Over-MPLS

IN THIS SECTION

- Configuring the Ingress PE Switch | 83
- Configuring the Egress PE Switch | 86

You can configure MPLS on EX Series switches to increase transport efficiency in your network. MPLS services can be used to connect various sites to a backbone network or to ensure better performance for low-latency applications such as VoIP and other business-critical functions.

To implement MPLS on switches, you must configure two provider edge (PE) switches—an ingress PE switch and an egress PE switch—and at least one provider switch. You can configure customer edge (CE) interfaces on the PE switches of the MPLS network by using either IP over MPLS or MPLS over circuit cross-connect (CCC).

The main differences between configuring IP over MPLS and configuring MPLS over CCC are that for IP over MLPS you configure the customer edge interfaces to belong to `family inet` (rather than `family ccc`) and you configure a static route for the label-switched path (LSP). The configuration of the provider switch is the same regardless of whether you have used IP over MPLS or MPLS over CCC. See "Configuring MPLS on EX8200 and EX4500 Provider Switches" on page 93.

This topic describes how to configure an ingress PE switch and an egress PE switch using IP over MPLS:

Configuring the Ingress PE Switch

To configure the ingress PE switch:

1. Configure an IP address for the loopback interface and for the core interfaces:

```
[edit]
user@switch# set interfaces lo0 unit 0 family inet address 100.100.100.100/32
user@switch# set interfaces ge-0/0/5 unit 0 family inet address 10.1.5.1/24
user@switch# set interfaces ge-0/0/6 unit 0 family inet address
10.1.6.1/24
```

2. Configure OSPF on the loopback and core interfaces:

```
[edit protocols]
user@switch# set ospf area 0.0.0.0 interface lo0.0
user@switch# set ospf area 0.0.0.0 interface ge-0/0/5.0
user@switch# set ospf area 0.0.0.0 interface ge-0/0/6.0
```

NOTE: If you want to use routed VLAN interfaces (RVIs) or Layer 3 subinterfaces as the core interfaces, replace ge-0/0/5.0 and ge-0/0/6 each with an RVI name (for example, vlan.*logical-interface-number*) or a subinterface name (for example, *interface-name.logical-unit-number*).

RVIs function as logical routers, eliminating the need to have both a switch and a router. Layer 3 subinterfaces allow you to route traffic among multiple VLANs along a single trunk line that connects an EX Series switch to a Layer 2 switch.

3. Enable traffic engineering for the routing protocol:

```
[edit protocols]
user@switch# set ospf traffic-engineering
```

4. Configure RSVP on the loopback interface and the core interfaces:

```
[edit protocols]
user@switch# set rsvp interface lo0.0
user@switch# set rsvp interface ge-0/0/5.0
user@switch# set rsvp interface ge-0/0/6.0
```

5. Configure MPLS traffic engineering:

```
[edit protocols]
user@switch# set protocols mpls traffic-engineering bgp-igp
```

6. Configure MPLS on the core interfaces:

```
[edit protocols]
user@switch# set mpls interface ge-0/0/5.0
user@switch# set mpls interface ge-0/0/6.0
```

7. Configure `family mpls` on the logical units of the core interfaces, thereby identifying the interfaces that will be used for forwarding MPLS packets:

```
[edit]
user@switch# set interfaces ge-0/0/5 unit 0 family mpls
user@switch# set interfaces ge-0/0/6 unit 0 family mpls
```

8. Configure a customer edge interface as a Layer 3 routed interface, specifying an IP address:

```
[edit]
user@switch# set interfaces ge-2/0/3 unit 0 family inet
121.121.121.1/16
```

9. Configure this Layer 3 customer edge interface for the routing protocol:

```
[edit]
user@switch# set protocols ospf area 0.0.0 interface
ge-2/0/3.0
```

10. Configure an LSP on the ingress PE switch (`100.100.100.100`) to send IP packets over MPLS to the egress PE switch (`208.208.208.208`):

```
[edit protocols mpls]
user@switch# set label-switched-path ip_lspjavae_29 from 100.100.100.100
user@switch# set label-switched-path ip_lspjavae_29 to 208.208.208.208
```

11. Disable constrained-path LSP computation for this LSP:

```
[edit protocols mpls]
user@switch# set label-switched-path ip_lspjavae_29 no-cspf
```

12. Configure a static route from the ingress PE switch to the egress PE switch, thereby indicating to the routing protocol that the packets will be forwarded over the MPLS LSP that has been set up to that destination:

NOTE: Do not configure a static route if you are using this procedure to configure an MPLS-based Layer 3 VPN.

```
[edit]
user@switch# set routing-options static route 2.2.2.0/24 next-hop 100.100.100.100
user@switch# set routing-options static route 2.2.2.0/24 resolve
```

Configuring the Egress PE Switch

To configure the egress PE switch:

1. Configure an IP address for the loopback interface and for the core interfaces:

```
[edit]
user@switch# set interfaces lo0 unit 0 family inet address 208.208.208.208/32
user@switch# set interfaces ge-0/0/5 unit 0 family inet address 10.1.20.1/24
user@switch# set interfaces ge-0/0/6 unit 0 family inet address 10.1.21.1/24
```

2. Configure OSPF on the loopback interface (or switch address) and core interfaces:

```
[edit protocols]
user@switch# set ospf area 0.0.0.0 interface lo0.0
user@switch# set ospf area 0.0.0.0 interface ge-0/0/5.0
user@switch# set ospf area 0.0.0.0 interface ge-0/0/6.0
```

NOTE: If you want to use routed VLAN interfaces (RVIs) or Layer 3 subinterfaces as the core interfaces, replace ge-0/0/5.0 and ge-0/0/6 each with an RVI name (for example, vlan.*logical-interface-number*) or a subinterface name (for example, *interface-name.logical-unit-number*).

RVIs function as logical routers, eliminating the need to have both a switch and a router. Layer 3 subinterfaces allow you to route traffic among multiple VLANs along a single trunk line that connects an EX Series switch to a Layer 2 switch.

3. Enable traffic engineering for the routing protocol:

```
[edit protocols]
user@switch# set ospf traffic-engineering
```

4. Configure RSVP on the loopback interface and the core interfaces:

```
[edit protocols]
user@switch# set rsvp interface lo0.0
user@switch# set rsvp interface ge-0/0/5.0
user@switch# set rsvp interface ge-0/0/6.0
```

5. Configure MPLS traffic engineering on both BGP and IGP destinations:

```
[edit protocols]
user@switch# set protocols mpls traffic-engineering bgp-igp
```

6. Configure MPLS on the core interfaces:

```
[edit protocols]
user@switch# set mpls interface ge-0/0/5.0
user@switch# set mpls interface ge-0/0/6.0
```

7. Configure `family mpls` on the logical units of the core interfaces, thereby identifying the interfaces that will be used for forwarding MPLS packets:

```
[edit]
user@switch# set interfaces ge-0/0/5 unit 0 family mpls
user@switch# set interfaces ge-0/0/6 unit 0 family mpls
```

8. Configure a customer edge interface as a Layer 3 routed interface, specifying an IP address:

```
[edit]
user@switch# set interfaces ge-2/0/3 unit 0 family inet address
2.2.2.1/16
```

9. Configure this Layer 3 customer edge interface for the routing protocol:

```
[edit]
user@switch# set protocols ospf area 0.0.0 interface ge-2/0/3
```

10. Configure an LSP on the egress PE switch (208.208.208.208) to send IP packets over MPLS to the ingress PE switch (100.100.100.100):

```
[edit protocols mpls]
user@switch# set label-switched-path ip_lsp29_javae from 208.208.208.208
user@switch# set label-switched-path ip_lsp29_javae to 100.100.100.100
```

11. Disable constrained-path LSP computation for this LSP:

```
[edit protocols mpls]
user@switch# set label-switched-path ip_lsp29_javae no-cspf
```

12. Configure a static route from the ingress PE switch to the egress PE switch, thereby indicating to the routing protocol that the packets will be forwarded over the MPLS LSP that has been set up to that destination:

NOTE: Do not configure a static route if you are using this procedure to configure an MPLS-based Layer 3 VPN.

```
[edit]
user@switch# set routing-options static route 121.121.121.0/24 next-hop 208.208.208.208
user@switch# set routing-options static route 121.121.121.0/24 resolve
```

Configuring MPLS on Provider Edge EX8200 and EX4500 Switches Using Circuit Cross-Connect

Junos OS MPLS for EX8200 and EX4500 switches supports Layer 2 protocols and Layer 2 virtual private networks (VPNs). You can configure MPLS on switches to increase transport efficiency in your network. MPLS services can be used to connect various sites to a backbone network and to ensure better performance for low-latency applications such as VoIP and other business-critical functions.

This topic describes configuring provider edge (PE) switches in an MPLS network using a circuit cross-connect (CCC). The customer edge interface can be either a simple interface or a tagged VLAN interface.

NOTE: If you are configuring a CCC on a tagged VLAN interface, you do not specify **family ccc**. See Configuring an MPLS-Based VLAN CCC Using a Layer 2 VPN and Configuring an MPLS-Based VLAN CCC Using a Layer 2 Circuit.

NOTE: If you are going through this procedure in preparation for configuring an MPLS-based Layer 2 VPN, you do not need to configure the association of the label-switched path (LSP) with the customer edge interface. The BGP signaling automates the connections, so manual configuration of the **connections** is not required.

The following guidelines apply to CCC configurations:

- When an interface is configured to belong to **family ccc**, it cannot belong to any other family.
- You can send any kind of traffic over a CCC, including nonstandard bridge protocol data units (BPDUs) generated by other vendors' equipment.
- If you are configuring a CCC on a tagged VLAN interface, you must explicitly enable VLAN tagging and specify a VLAN ID. The VLAN ID cannot be configured on logical interface unit **0**. The logical unit number must be **1** or higher. See Configuring an MPLS-Based VLAN CCC Using a Layer 2 VPN and Configuring an MPLS-Based VLAN CCC Using a Layer 2 Circuit.

This procedure shows how to set up two CCCs:

- If you are configuring a CCC on a simple interface (**ge-0/0/1**), you do not need to enable VLAN tagging or specify a VLAN ID, so you skip those steps.
- If you are configuring a CCC on a tagged VLAN interface (**ge-0/0/2**), include all the steps in this procedure.

To configure a PE switch with a CCC:

1. Configure OSPF (or IS-IS) on the loopback (or switch address) and core interfaces:

```
[edit protocols]
user@switch# set ospf area 0.0.0.0 interface lo0.0
user@switch# set ospf area 0.0.0.0 interface ge-0/0/5.0
user@switch# set ospf area 0.0.0.0 interface ge-0/0/6.0
user@switch# set ospf area 0.0.0.0 interface ae0
```

2. Enable traffic engineering for the routing protocol:

```
[edit protocols]
user@switch# set ospf traffic-engineering
```

3. Configure an IP address for the loopback interface and for the core interfaces:

```
[edit]
user@switch# set interfaces lo0 unit 0 family inet address 127.1.1.1/32
user@switch# set interfaces ge-0/0/5 unit 0 family inet address 10.1.5.1/24
user@switch# set interfaces ge-0/0/6 unit 0 family inet address 10.1.6.1/24
user@switch# set interfaces ae0 unit 0 family inet address
10.1.9.1/24
```

4. Enable MPLS and define the LSP:

```
[edit protocols]
user@switch# set mpls                                label-switched-path lsp_to_pe2_ge1 to
127.1.1.3
```

TIP: **lsp_to_pe2_ge1** is the LSP name. You will need to use the specified name again when configuring the CCC.

5. Configure MPLS on the core interfaces:

```
[edit protocols]
user@switch# set mpls interface ge-0/0/5.0
user@switch# set mpls interface ge-0/0/6.0
```

```
user@switch# set mpls interface ae0
```

6. Configure RSVP on the loopback interface and the core interfaces:

```
[edit protocols]
user@switch# set rsvp interface lo0.0
user@switch# set rsvp interface ge-0/0/5.0
user@switch# set rsvp interface ge-0/0/6.0
user@switch# set rsvp interface ae0
```

7. Configure **family mpls** on the logical units of the core interfaces:

```
[edit]
user@switch# set interfaces ge-0/0/5 unit 0 family mpls
user@switch# set interfaces ge-0/0/6 unit 0 family mpls
user@switch# set interfaces ae0 unit 0 family mpls
```

NOTE: You can enable **family mpls** on either individual interfaces or aggregated Ethernet interfaces. You cannot enable it on tagged VLAN interfaces.

8. If you are configuring a CCC on a tagged VLAN interface, enable VLAN tagging on the customer edge interface **ge-0/0/2** of the local PE switch:

```
[edit interfaces ge-0/0/2]
user@switch# set vlan-tagging
```

If you are configuring a CCC on a simple interface (**ge-0/0/1**), omit this step.

9. If you are configuring a CCC on a tagged VLAN interface, configure the logical unit of the customer edge interface with a VLAN ID:

```
[edit interfaces ge-0/0/2 unit 1]
user@switch# set vlan-id 100
```

If you are configuring a CCC on a simple interface (**ge-0/0/1**), omit this step.

10. Configure the logical unit of the customer edge interface to belong to **family ccc**:

- On a simple interface:

```
[edit interfaces ge-0/0/1 unit 0]
user@switch# set family ccc
```

- On a tagged VLAN interface:

```
[edit interfaces ge-0/0/2 unit 1]
user@switch# set family ccc
```

11. Associate the CCC interface with two LSPs, one for transmitting MPLS packets and the other for receiving MPLS packets:

NOTE: If you are configuring a Layer 2 VPN, omit this step. The BGP signaling automates the connections, so manual configuration of the **connections** is not required.

- On a simple interface:

```
[edit protocols]
user@switch# set connections                                remote-interface-switch
ge-1-to-pe2 interface ge-0/0/1.0
user@switch# set connections remote-interface-switch ge-1-to-pe2 transmit-lsp
lsp_to_pe2_ge1
user@switch# set connections remote-interface-switch ge-1-to-pe2 receive-lsp
lsp_to_pe1_ge1
```

- On a tagged VLAN interface:

```
[edit protocols]
user@switch# set connections remote-interface-switch ge-1-to-pe2 interface ge-0/0/2.1
user@switch# set connections remote-interface-switch ge-1-to-pe2 transmit-lsp
lsp_to_pe2_ge1
user@switch# set connections remote-interface-switch ge-1-to-pe2 receive-lsp
lsp_to_pe1_ge1
```

TIP: The **transmit-lsp** option specifies the LSP name that was configured on PE-1 (the local PE switch) by the `label-switched-path` statement within the `[edit protocols mpls]` hierarchy.
The **receive-lsp** option specifies the LSP name that was configured on PE-2 (the remote PE switch) by the `label-switched-path` statement within the `[edit protocols mpls]` hierarchy.

When you have completed configuring one PE switch, follow the same procedures to configure the other PE switch.

Configuring MPLS on EX8200 and EX4500 Provider Switches

You can configure MPLS on EX8200 and EX4500 switches to increase transport efficiency in your network. MPLS services can be used to connect various sites to a backbone network and to ensure better performance for low-latency applications such as VoIP and other business-critical functions.

To implement MPLS on EX Series switches, you must configure at least one provider switch as a transit switch for the MPLS packets. The configuration of all the provider switches remains the same regardless of whether the provider edge (PE) switches are using circuit cross-connect (CCC) or using MPLS over IP for the customer edge interfaces. Likewise, you do not need to change the configuration of the provider switches if you implement an MPLS-based Layer 2 VPN, Layer 3 VPN, or a Layer 2 circuit configuration.

MPLS requires the configuration of a routing protocol (OSPF or IS-IS) on the core interfaces and the loopback interface of all the switches. This procedure includes the configuration of OSPF on the provider switch. For information on configuring IS-IS as the routing protocol, see *Junos OS Routing Protocols Configuration Guide*.

To configure the provider switch, complete the following tasks:

1. Enable the routing protocol (OSPF or IS-IS) on the loopback interface and on the core interfaces:

NOTE: You can use the switch address as an alternative to the loopback interface.

```
[edit protocols]
user@switch# set ospf area 0.0.0.0  interface lo0.0
user@switch# set ospf area 0.0.0.0  interface ge-0/0/5.0
user@switch# set ospf area 0.0.0.0 interface ge-0/0/6.0
user@switch# set ospf area 0.0.0.0  interface ae0
```

2. Enable traffic engineering for the routing protocol (traffic engineering must be explicitly enabled for OSPF):

```
[edit protocols]
user@switch# set ospf traffic-engineering
```

3. Enable MPLS within the **protocols** stanza and apply it to the core interfaces:

```
[edit protocols]
user@switch# set mpls interface ge-0/0/5.0
user@switch# set mpls interface ge-0/0/6.0
user@switch# set mpls interface ae0
```

4. Configure RSVP on the loopback interface and the core interfaces:

```
[edit protocols]
user@switch# set rsvp interface lo0.0
user@switch# set rsvp interface ge-0/0/5.0
user@switch# set rsvp interface ge-0/0/6.0
user@switch# set rsvp interface ae0
```

5. Configure an IP address for the loopback interface and for the core interfaces:

```
[edit]
user@switch# set interfaces lo0 unit 0 family inet address 127.1.1.1/32
user@switch# set interfaces ge-0/0/5 unit 0 family inet address 10.1.5.1/24
user@switch# set interfaces ge-0/0/6 unit 0 family inet address 10.1.6.1/24
user@switch# set interfaces ae0 unit 0 family inet address 10.1.9.2/24
```

6. Configure **family mpls** on the logical units of the core interfaces:

```
[edit]
user@switch# set interfaces ge-0/0/5 unit 0 family mpls
user@switch# set interfaces ge-0/0/6 unit 0 family mpls
user@switch# set interfaces ae0 unit 0 family mpls
```

NOTE: You can enable **family mpls** on either individual interfaces or aggregated Ethernet interfaces. You cannot enable it on tagged VLAN interfaces.

RELATED DOCUMENTATION

MPLS Overview | 2

MPLS Configuration on IRB Interfaces

IN THIS SECTION

- MPLS Support on IRB Interfaces | 95
- Configure MPLS on IRB Interfaces | 97

MPLS Support on IRB Interfaces

IN THIS SECTION

- MPLS over IRB Interfaces Overview | 95
- Benefits of Configuring MPLS over IRB Interfaces | 96

With the introduction of MPLS support for IRB (Integrated Routing and Bridging) interfaces on the MX240, MX304, MX480, MX960, MX10004, and MX10008 platforms, you can seamlessly integrate routing and switching over an MPLS core. This feature allows for efficient traffic forwarding, supporting VLAN-based routing (IRB) while maintaining MPLS label switching. By leveraging this enhancement, your network can optimize the path selection process, reduce forwarding delays, and ensure compatibility with complex MPLS topologies.

This feature addresses prior limitations where MPLS encapsulation was not supported on IRB interfaces. With this update, IRB interfaces can encapsulate MPLS labels, ensuring interoperability and full MPLS functionality.

MPLS over IRB Interfaces Overview

With MPLS support on IRB interfaces, you can seamlessly integrate Layer 2 and Layer 3 forwarding over an MPLS-enabled network. This enhancement allows IRB interfaces to encapsulate MPLS labels, enabling efficient label switching alongside VLAN-based routing. You can now bridge and route VLAN

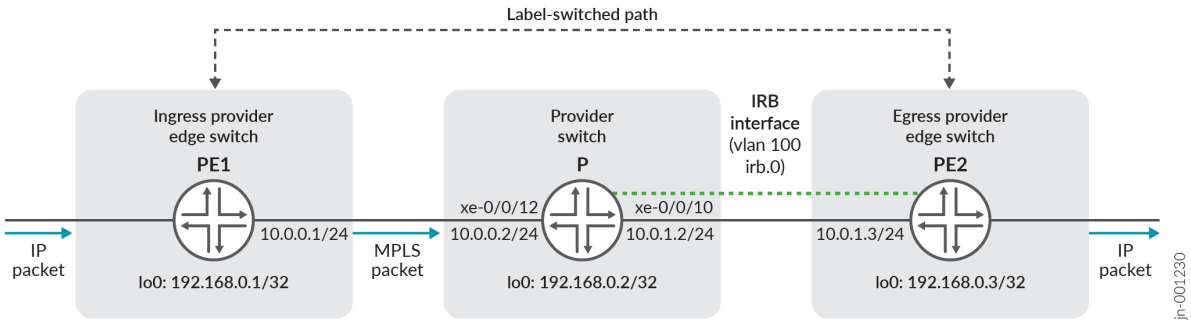
traffic while taking full advantage of MPLS features such as Label Switched Paths (LSPs) and end-to-end traffic engineering.

This feature is particularly valuable in scenarios such as Data Center Interconnect (DCI), where VLANs need to be transported over an MPLS backbone, or when you require a unified approach to bridging and routing. By enabling MPLS on IRB interfaces, you reduce operational complexity, enhance scalability, and ensure compatibility with modern MPLS-based architectures.

Using Junos OS, you can configure IRB interfaces to support MPLS for both IPv4 and IPv6 traffic. This involves defining IRB units for routing VLANs, applying MPLS family settings, and integrating these interfaces with your MPLS configuration. Once configured, the IRB interfaces perform MPLS encapsulation, allowing seamless forwarding across MPLS-enabled devices.

The MPLS over IRB feature enables you to use IRB interfaces for routing between VLANs while simultaneously leveraging MPLS for efficient label switching. This integration is crucial in scenarios requiring VLAN segmentation over MPLS networks, such as Data Center Interconnect (DCI) stitching.

Figure 3: IRB Topology over an MPLS Core Network




```
CE1 --- PE1 (IRB) --- P --- PE2 (IRB) --- CE2
```

- **PE1**: Ingress provider edge with IRB.638 for VLAN 638.
- **P**: Core switch performing MPLS label swaps.
- **PE2**: Egress provider edge.

Benefits of Configuring MPLS over IRB Interfaces

- **Simplified Interconnectivity**: Integrates routing and bridging with MPLS for streamlined operations.
- **Enhanced Flexibility**: Supports both unicast and multicast traffic.
- **Reduced Overhead**: Utilizes MPLS label switching to minimize routing table lookups.

- **Data Center Interconnect (DCI):** Simplifies interconnecting geographically dispersed data centers over MPLS while maintaining VLAN segregation.
- **Multicast and Unicast Traffic Support:** Seamlessly handle both traffic types using MPLS-enabled IRB interfaces.

By implementing this feature, you enhance your network's ability to efficiently route and switch traffic across MPLS cores while preserving the flexibility of VLAN routing.

Configure MPLS on IRB Interfaces

Step-by-Step Procedure

1. Configure IRB Interface:

   Assign MPLS, IPv4, and IPv6 families to the IRB interface.

```
set interfaces irb unit 638 description "IRB Interface for VLAN 638"
set interfaces irb unit 638 family inet address 10.0.0.2/31
set interfaces irb unit 638 family inet6 address fd00:0:200::4e/127
set interfaces irb unit 638 family mpls
```

2. Configure VLAN Bridging:

   Bind the IRB interface to the VLAN.

```
set bridge-domains vlan638 vlan-id 638
set bridge-domains vlan638 routing-interface irb.638
```

3. Enable MPLS:

   Apply MPLS to the core-facing interfaces and the IRB interface.

```
set protocols mpls interface irb.638
set protocols mpls label-switched-path lsp-name to next-hop
```

Verification Commands

1. Verify MPLS Configuration:

```
show mpls interface
show mpls interface detail
```

2. Check IRB Interface:

```
show interfaces irb.638
```

CHAPTER 4

Configuring MPLS Tunnels

IN THIS CHAPTER

- IPv6-over-Ipv4 Tunnels | 99
- Next-Hop-Based Dynamic Tunnels | 113

IPv6-over-Ipv4 Tunnels

IN THIS SECTION

- Configuring IPv6 Tunneling for MPLS | 99
- Example: Tunneling IPv6 Traffic over MPLS IPv4 Networks | 101

Configuring IPv6 Tunneling for MPLS

You can configure the IPv6 tunneling for MPLS to tunnel IPv6 traffic over an MPLS-based IPv4 network. This configuration allows you to interconnect a number of smaller IPv6 networks over an IPv4-based network core, giving you the ability to provide IPv6 service without having to upgrade the switches in your core network. BGP is configured to exchange routes between the IPv6 networks, and data is tunneled between these IPv6 networks by means of IPv4-based MPLS.

To configure IPv6 tunneling for MPLS on your EX Series switch:

1. Configure IPv4 and IPv6 IP addresses for all the core interfaces:

```
[edit]
user@switch# set interfaces interface-name unit logical-unit-number family inet address
address
```

2. Configure the number assigned to you by the Network Information Center (NIC) as the autonomous system (AS) number

```
[edit routing-options]
user@switch# set autonomous-system number
```

3. Advertise label 0 to the egress router of the LSP:

```
[edit protocols]
user@switch# set mpls explicit-null
```

4. Configure the LSP to allow IPv6 routes to be resolved over an MPLS network by converting all routes stored in the inet3 routing table to IPv4-mapped IPv6 addresses and then copying them into the inet6.3 routing table:

```
[edit protocols]
user@switch# set mpls ipv6-tunneling
```

5. Set the local AS number:

```
[edit protocols bgp]
user@switch# set local-as local-autonomous-system-number
```

6. Configure the default import and export policies:

```
[edit protocols bgp]
user@switch# set local-address address
user@switch# set import default-import
user@switch# set family inet6 labeled-unicast explicit-null
user@switch# set export default-export
```

7. Configure a BGP group that recognizes only the specified BGP systems as peers. Define a group name, group type, local end of a BGP session, and a neighbor (peer). To configure multiple BGP peers, include multiple neighbor statements:

```
[edit protocols bgp]
user@switch# set group group-name type internal
user@switch# set group group-name local-address address-of-the-local-end-of-a-bgp-session
user@switch# set group group-name family inet6 labeled-unicast explicit-null
```

```
user@switch# set group group-name peer-as peer-autonomous-system-number
user@switch# set group group-name neighbor address family inet6 labeled-unicast explicit-null
```

8. Configure routing options to accept the default import and export policies:

```
[edit policy-options]
user@switch# set policy-statement default-import then accept
user@switch# set policy-statement default-export then accept
```

Example: Tunneling IPv6 Traffic over MPLS IPv4 Networks

IN THIS SECTION

- Requirements | **101**
- Overview | **101**
- Configuration | **104**
- Verification | **112**

This example shows how to configure the Junos OS to tunnel IPv6 over an MPLS-based IPv4 network. External BGP (EBGP) is used between the customer edge (CE) and provider edge (PE) devices. The remote CE devices have different AS numbers for loop detection.

Requirements

No special configuration beyond device initialization is required before you configure this example.

Overview

Detailed information about the Juniper Networks implementation of IPv6 over MPLS is described in the following Internet drafts:

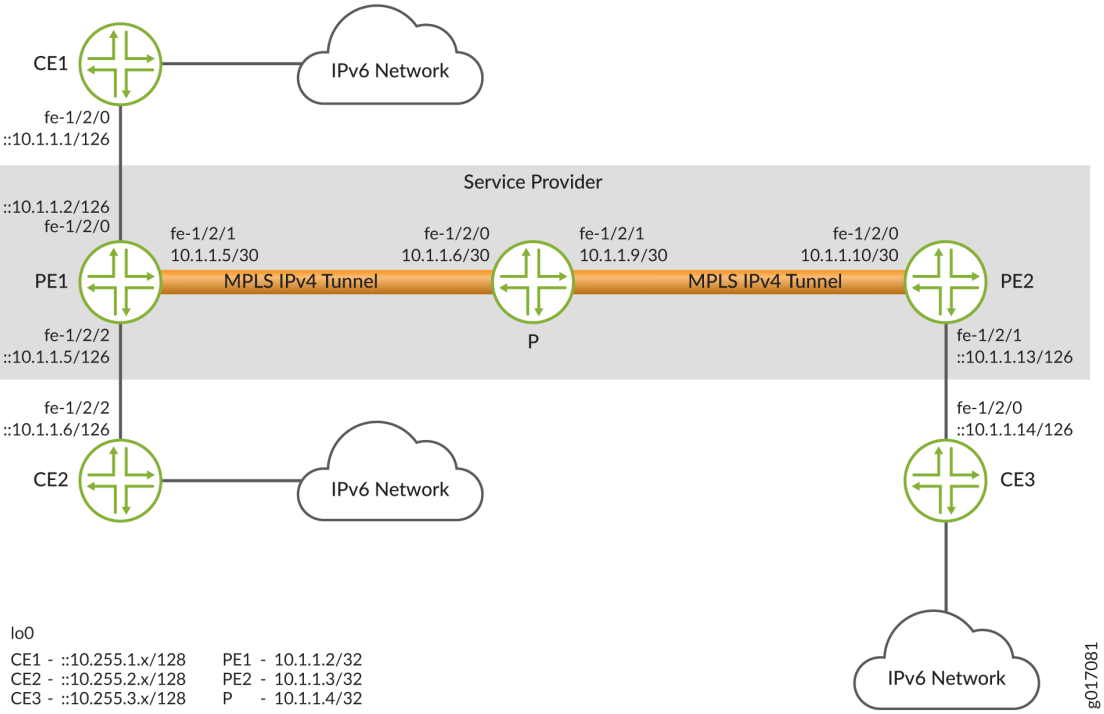
- Internet draft draft-ietf-l3vpn-bgp-ipv6-07.txt, *BGP-MPLS IP VPN extension for IPv6 VPN* (expires January 2006)
- Internet draft draft-ooms-v6ops-bgp-tunnel-06.txt, *Connecting IPv6 Islands over IPv4 MPLS using IPv6 Provider Edge Routers* (expires July 2006)

These Internet drafts are available on the IETF website at http://www.ietf.org/.

This example shows you how to interconnect a two IPv6 networks over an IPv4-based network core, giving you the ability to provide IPv6 service without having to upgrade the routers in your core network. Multiprotocol Border Gateway Protocol (MP-BGP) is configured to exchange routes between the IPv6 networks, and data is tunneled between these IPv6 networks by means of IPv4-based MPLS.

In Figure 4 on page 102, Routers PE1 and PE2 are dual-stack BGP routers, meaning they have both IPv4 and IPv6 stacks. The PE routers link the IPv6 networks through the customer edge (CE) routers to the IPv4 core network. The CE routers and the PE routers connect through a link layer that can carry IPv6 traffic. The PE routers use IPv6 on the CE router-facing interfaces and use IPv4 and MPLS on the core-facing interfaces. Note that one of the connected IPv6 networks could be the global IPv6 Internet.

Figure 4: IPv6 Networks Linked by MPLS IPv4 Tunnels



The two PE routers are linked through an MP-BGP session using IPv4 addresses. They use the session to exchange IPv6 routes with an IPv6 (value 2) address family indicator (AFI) and a subsequent AFI (SAFI) (value 4). Each PE router sets the next hop for the IPv6 routes advertised on this session to its own IPv4 address. Because MP-BGP requires the BGP next hop to correspond to the same address family as the network layer reachability information (NLRI), this IPv4 address needs to be embedded within an IPv6 format.

The PE routers can learn the IPv6 routes from the CE routers connected to them using routing protocols Routing Information Protocol next generation (RIPng) or MP-BGP, or through static configuration. Note that if BGP is used as the PE-router-to-CE-router protocol, the MP-BGP session between the PE router

and CE router could occur over an IPv4 or IPv6 Transmission Control Protocol (TCP) session. Also, the BGP routes exchanged on that session would have SAFI unicast. You must configure an export policy to pass routes between IBGP and EBGP, and between BGP and any other protocol.

The PE routers have MPLS LSPs routed to each others' IPv4 addresses. IPv4 provides signaling for the LSPs by means of either LDP or RSVP. These LSPs are used to resolve the next-hop addresses of the IPv6 routes learned from MP-BGP. The next hops use IPv4-mapped IPv6 addresses, while the LSPs use IPv4 addresses.

The PE routers always advertise IPv6 routes to each other using a label value of 2, the explicit null label for IPv6 as defined in RFC 3032, *MPLS Label Stack Encoding*. As a consequence, each of the forwarding next hops for the IPv6 routes learned from remote PE routers normally push two labels. The inner label is 2 (this label could be different if the advertising PE router is not a Juniper Networks routing platform), and the outer label is the LSP label. If the LSP is a single-hop LSP, then only Label 2 is pushed.

It is also possible for the PE routers to exchange plain IPv6 routes using SAFI unicast. However, there is one major advantage in exchanging labeled IPv6 routes. The penultimate-hop router for an MPLS LSP can pop the outer label and then send the packet with the inner label as an MPLS packet. Without the inner label, the penultimate-hop router would need to discover whether the packet is an IPv4 or IPv6 packet to set the protocol field in the Layer 2 header correctly.

When the PE1 router in Figure 4 on page 102 receives an IPv6 packet from the CE1 router, it performs a lookup in the IPv6 forwarding table. If the destination matches a prefix learned from the CE2 router, then no labels need to be pushed and the packet is simply sent to the CE2 router. If the destination matches a prefix that was learned from the PE2 router, then the PE1 router pushes two labels onto the packet and sends it to the provider router. The inner label is 2 and the outer label is the LSP label for the PE2 router.

Each provider router in the service provider's network handles the packet as it would any MPLS packet, swapping labels as it passes from provider router to provider router. The penultimate-hop provider router for the LSP pops the outer label and sends the packet to the PE2 router. When the PE2 router receives the packet, it recognizes the IPv6 explicit null label on the packet (Label 2). It pops this label and treats it as an IPv6 packet, performing a lookup in the IPv6 forwarding table and forwarding the packet to the CE3 router.

This example includes the following settings:

- In addition to configuring the `family inet6` statement on all the CE router–facing interfaces, you must also configure the statement on all the core-facing interfaces running MPLS. Both configurations are necessary because the router must be able to process any IPv6 packets it receives on these interfaces. You should not see any regular IPv6 traffic arrive on these interfaces, but you will receive MPLS packets tagged with Label 2. Even though Label 2 MPLS packets are sent in IPv4, these packets are treated as native IPv6 packets.

- You enable IPv6 tunneling by including the `ipv6-tunneling` statement in the configuration for the PE routers. This statement allows IPv6 routes to be resolved over an MPLS network by converting all

routes stored in the inet.3 routing table to IPv4-mapped IPv6 addresses and then copying them into the inet6.3 routing table. This routing table can be used to resolve next hops for both inet6 and inet6-vpn routes.

> **NOTE**: BGP automatically runs its import policy even when copying routes from a primary routing table group to a secondary routing table group. If IPv4 labeled routes arrive from a BGP session (for example, when you have configured the `labeled-unicast` statement at the `[edit protocols bgp family inet]` hierarchy level on the PE router), the BGP neighbor's import policy also accepts IPv6 routes, since the neighbor's import policy is run while doing the copy operation to the inet6.3 routing table.

- When you configure MP-BGP to carry IPv6 traffic, the IPv4 MPLS label is removed at the destination PE router. The remaining IPv6 packet without a label can then be forwarded to the IPv6 network. To enable this, include the `explicit-null` statement in the BGP configuration.

Configuration

IN THIS SECTION

- CLI Quick Configuration | **104**
- Configuring Device PE1 | **107**

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the `[edit]` hierarchy level.

Device PE1

```
set interfaces fe-1/2/0 unit 0 family inet6 address ::10.1.1.2/126
set interfaces fe-1/2/0 unit 0 family mpls
set interfaces fe-1/2/1 unit 0 family inet address 10.1.1.5/30
set interfaces fe-1/2/1 unit 0 family inet6
set interfaces fe-1/2/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.1.1.2/32
set protocols mpls ipv6-tunneling
set protocols mpls interface fe-1/2/0.0
```

```
set protocols mpls interface fe-1/2/1.0
set protocols bgp group toCE1 type external
set protocols bgp group toCE1 local-address ::10.1.1.2
set protocols bgp group toCE1 family inet6 unicast
set protocols bgp group toCE1 export send-bgp6
set protocols bgp group toCE1 peer-as 65001
set protocols bgp group toCE1 neighbor ::10.1.1.1
set protocols bgp group toPE2 type internal
set protocols bgp group toPE2 local-address 10.1.1.2
set protocols bgp group toPE2 family inet6 labeled-unicast explicit-null
set protocols bgp group toPE2 export next-hop-self
set protocols bgp group toPE2 export send-v6
set protocols bgp group toPE2 neighbor 10.1.1.4
set protocols ospf area 0.0.0.0 interface fe-1/2/1.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ldp interface fe-1/2/1.0
set policy-options policy-statement next-hop-self then next-hop self
set policy-options policy-statement send-bgp6 from family inet6
set policy-options policy-statement send-bgp6 from protocol bgp
set policy-options policy-statement send-bgp6 then accept
set policy-options policy-statement send-v6 from family inet6
set policy-options policy-statement send-v6 from protocol bgp
set policy-options policy-statement send-v6 from protocol direct
set policy-options policy-statement send-v6 then accept
set routing-options router-id 10.1.1.2
set routing-options autonomous-system 65002
```

Device PE2

```
set interfaces fe-1/2/0 unit 0 family inet address 10.1.1.10/30
set interfaces fe-1/2/0 unit 0 family inet6
set interfaces fe-1/2/0 unit 0 family mpls
set interfaces fe-1/2/1 unit 0 family inet6 address ::10.1.1.13/126
set interfaces fe-1/2/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.1.1.4/32
set protocols mpls ipv6-tunneling
set protocols mpls interface fe-1/2/0.0
set protocols mpls interface fe-1/2/1.0
set protocols bgp group toPE1 type internal
set protocols bgp group toPE1 local-address 10.1.1.4
set protocols bgp group toPE1 family inet6 labeled-unicast explicit-null
set protocols bgp group toPE1 export next-hop-self
```

```
set protocols bgp group toPE1 export send-v6
set protocols bgp group toPE1 neighbor 10.1.1.2
set protocols bgp group toCE3 type external
set protocols bgp group toCE3 local-address ::10.1.1.13
set protocols bgp group toCE3 family inet6 unicast
set protocols bgp group toCE3 export send-bgp6
set protocols bgp group toCE3 peer-as 65003
set protocols bgp group toCE3 neighbor ::10.1.1.14
set protocols ospf area 0.0.0.0 interface fe-1/2/0.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ldp interface fe-1/2/0.0
set policy-options policy-statement next-hop-self then next-hop self
set policy-options policy-statement send-bgp6 from family inet6
set policy-options policy-statement send-bgp6 from protocol bgp
set policy-options policy-statement send-bgp6 then accept
set policy-options policy-statement send-v6 from family inet6
set policy-options policy-statement send-v6 from protocol bgp
set policy-options policy-statement send-v6 from protocol direct
set policy-options policy-statement send-v6 then accept
set routing-options router-id 10.1.1.4
set routing-options autonomous-system 65002
```

Device P

```
set interfaces fe-1/2/0 unit 0 family inet address 10.1.1.6/30
set interfaces fe-1/2/0 unit 0 family inet6
set interfaces fe-1/2/0 unit 0 family mpls
set interfaces fe-1/2/1 unit 0 family inet address 10.1.1.9/30
set interfaces fe-1/2/1 unit 0 family inet6
set interfaces fe-1/2/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.1.1.3/32
set protocols mpls interface fe-1/2/0.0
set protocols mpls interface fe-1/2/1.0
set protocols ospf area 0.0.0.0 interface fe-1/2/0.0
set protocols ospf area 0.0.0.0 interface fe-1/2/1.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ldp interface fe-1/2/0.0
set protocols ldp interface fe-1/2/1.0
set routing-options router-id 10.1.1.3
set routing-options autonomous-system 65002
```

Device CE1

```
set interfaces fe-1/2/0 unit 0 family inet6 address ::10.1.1.1/126
set interfaces lo0 unit 0 family inet6 address ::10.255.1.1/128
set protocols bgp group toPE1 type external
set protocols bgp group toPE1 local-address ::10.1.1.1
set protocols bgp group toPE1 family inet6 unicast
set protocols bgp group toPE1 export send-v6
set protocols bgp group toPE1 peer-as 65002
set protocols bgp group toPE1 neighbor ::10.1.1.2
set policy-options policy-statement send-v6 from family inet6
set policy-options policy-statement send-v6 from protocol direct
set policy-options policy-statement send-v6 then accept
set routing-options router-id 10.255.1.1
set routing-options autonomous-system 65001
```

Device CE3

```
set interfaces fe-1/2/0 unit 0 family inet6 address ::10.1.1.14/126
set interfaces lo0 unit 0 family inet6 address ::10.255.1.5/128
set protocols bgp group toPE2 type external
set protocols bgp group toPE2 local-address ::10.1.1.14
set protocols bgp group toPE2 family inet6 unicast
set protocols bgp group toPE2 export send-v6
set protocols bgp group toPE2 peer-as 65002
set protocols bgp group toPE2 neighbor ::10.1.1.13
set policy-options policy-statement send-v6 from family inet6
set policy-options policy-statement send-v6 from protocol direct
set policy-options policy-statement send-v6 then accept
set routing-options router-id 10.255.1.5
set routing-options autonomous-system 65003
```

Configuring Device PE1

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the Junos OS CLI User Guide.

To configure Device PE1:

1. Configure the interfaces.

```
[edit interfaces]
user@PE1# set fe-1/2/0 unit 0 family inet6 address ::10.1.1.2/126
user@PE1# set fe-1/2/0 unit 0 family mpls
user@PE1# set fe-1/2/1 unit 0 family inet address 10.1.1.5/30
user@PE1# set fe-1/2/1 unit 0 family inet6
user@PE1# set fe-1/2/1 unit 0 family mpls
user@PE1# set lo0 unit 0 family inet address 10.1.1.2/32
```

2. Configure MPLS on the interfaces.

```
[edit protocols mpls]
user@PE1# set ipv6-tunneling
user@PE1# set interface fe-1/2/0.0
user@PE1# set interface fe-1/2/1.0
```

3. Configure BGP.

```
[edit protocols bgp]
user@PE1# set group toCE1 type external
user@PE1# set group toCE1 local-address ::10.1.1.2
user@PE1# set group toCE1 family inet6 unicast
user@PE1# set group toCE1 export send-bgp6
user@PE1# set group toCE1 peer-as 65001
user@PE1# set group toCE1 neighbor ::10.1.1.1
user@PE1# set group toPE2 type internal
user@PE1# set group toPE2 local-address 10.1.1.2
user@PE1# set group toPE2 family inet6 labeled-unicast explicit-null
user@PE1# set group toPE2 export next-hop-self
user@PE1# set group toPE2 export send-v6
user@PE1# set group toPE2 neighbor 10.1.1.4
```

4. Configure OSPF

```
[edit protocols ospf area 0.0.0.0]
user@PE1# set interface fe-1/2/1.0
user@PE1# set interface lo0.0 passive
```

5. Configure a signaling protocol.

```
[edit protocols]
user@PE1# set ldp interface fe-1/2/1.0
```

6. Configure the routing policies.

```
[edit policy-options]
user@PE1# set policy-statement next-hop-self then next-hop self
user@PE1# set policy-statement send-bgp6 from family inet6
user@PE1# set policy-statement send-bgp6 from protocol bgp
user@PE1# set policy-statement send-bgp6 then accept
user@PE1# set policy-statement send-v6 from family inet6
user@PE1# set policy-statement send-v6 from protocol bgp
user@PE1# set policy-statement send-v6 from protocol direct
user@PE1# set policy-statement send-v6 then accept
```

7. Configure the router ID and the autonomous system (AS) number.

```
[edit routing-options]
user@PE1# set router-id 10.1.1.2
user@PE1# set autonomous-system 675002
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show policy-options`, `show protocols`, and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R1# show interfaces
fe-1/2/0 {
    unit 0 {
        family inet6 {
            address ::10.1.1.2/126;
        }
        family mpls;
    }
}
```

```
fe-1/2/1 {
    unit 0 {
        family inet {
            address 10.1.1.5/30;
        }
        family inet6;
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.1.1.2/32;
        }
    }
}
```

```
user@R1# show policy-options
policy-statement next-hop-self {
    then {
        next-hop self;
    }
}
policy-statement send-bgp6 {
    from {
        family inet6;
        protocol bgp;
    }
    then accept;
}
policy-statement send-v6 {
    from {
        family inet6;
        protocol [ bgp direct ];
    }
    then accept;
}
```

```
user@R1# show protocols
mpls {
```

```
    ipv6-tunneling;
    interface fe-1/2/0.0;
    interface fe-1/2/1.0;
}
bgp {
    group toCE1 {
        type external;
        local-address ::10.1.1.2;
        family inet6 {
            unicast;
        }
        export send-bgp6;
        peer-as 65001;
        neighbor ::10.1.1.1;
    }
    group toPE2 {
        type internal;
        local-address 10.1.1.2;
        family inet6 {
            labeled-unicast {
                explicit-null;
            }
        }
        export [ next-hop-self send-v6 ];
        neighbor 10.1.1.4;
    }
}
ospf {
    area 0.0.0.0 {
        interface fe-1/2/1.0;
        interface lo0.0 {
            passive;
        }
    }
}
ldp {
```

```
    interface fe-1/2/1.0;
}
```

```
user@R1# show routing-options
router-id 10.1.1.2;
autonomous-system 65002;
```

If you are done configuring the device, enter `commit` from configuration mode.
Configure the other devices in the topology, as shown in "CLI Quick Configuration" on page 104.

Verification

IN THIS SECTION

- Verifying That the CE Devices Have Connectivity | **112**

Confirm that the configuration is working properly.

Verifying That the CE Devices Have Connectivity

Purpose

Make sure that the tunnel is operating.

Action

From operational mode, enter the `ping` command.

```
user@CE1> ping ::10.1.1.14
PING6(56=40+8+8 bytes) ::10.1.1.1 --> ::10.1.1.14
16 bytes from ::10.1.1.14, icmp_seq=0 hlim=61 time=10.687 ms
```

```
16 bytes from ::10.1.1.14, icmp_seq=1 hlim=61 time=9.239 ms
16 bytes from ::10.1.1.14, icmp_seq=2 hlim=61 time=1.842 ms
```

```
user@CE3> ping ::10.1.1.1
PING6(56=40+8+8 bytes) ::10.1.1.14 --> ::10.1.1.1
16 bytes from ::10.1.1.1, icmp_seq=0 hlim=61 time=1.484 ms
16 bytes from ::10.1.1.1, icmp_seq=1 hlim=61 time=1.338 ms
16 bytes from ::10.1.1.1, icmp_seq=2 hlim=61 time=1.351 ms
```

Meaning

The IPv6 CE devices can communicate over the core IPv4 network.

RELATED DOCUMENTATION

Basic MPLS Configuration | 46

Next-Hop-Based Dynamic Tunnels

IN THIS SECTION

- Example: Configuring Next-Hop-Based MPLS-Over-UDP Dynamic Tunnels | 114
- Anti-Spoofing Protection for Next-Hop-Based Dynamic Tunnels Overview | 132
- Example: Configuring Anti-Spoofing Protection for Next-Hop-Based Dynamic Tunnels | 135
- Next-Hop-Based Dynamic Tunnel Localization Overview | 149
- Overview of Next-Hop-Based Dynamic Tunneling Using IP-Over-IP Encapsulation | 155
- Example: Configuring Next-Hop-Based IP-Over-IP Dynamic Tunnels | 157

Example: Configuring Next-Hop-Based MPLS-Over-UDP Dynamic Tunnels

IN THIS SECTION

- Requirements | **114**
- Overview | **115**
- Configuration | **119**
- Verification | **126**
- Troubleshooting | **131**

This example shows how to configure a dynamic MPLS-over-UDP tunnel that includes a tunnel composite next hop. The MPLS-over-UDP feature provides a scaling advantage on the number of IP tunnels supported on a device.

Starting in Junos OS Release 18.3R1, MPLS-over-UDP tunnels are supported on PTX Series routers and QFX Series switches. For every dynamic tunnel configured on a PTX router or a QFX switch, a tunnel composite next hop, an indirect next hop, and a forwarding next hop is created to resolve the tunnel destination route. You can also use policy control to resolve the dynamic tunnel over select prefixes by including the forwarding-rib configuration statement at the `[edit routing-options dynamic-tunnels]` hierarchy level.

Requirements

This example uses the following hardware and software components:

- Five MX Series routers with MPCs and MICs.
- Junos OS Release 16.2 or later running on the provider edge (PE) routers.

Before you begin:

1. Configure the device interfaces, including the loopback interface.
2. Configure the router ID and autonmous system number for the device.
3. Establish an internal BGP (IBGP) session with the remote PE device.
4. Establish OSPF peering among the devices.

Overview

IN THIS SECTION

- Topology | 118

Starting with Junos OS Release 16.2, a dynamic UDP tunnel supports the creation of a tunnel composite next hop for every UDP tunnel configured. These next-hop-based dynamic UDP tunnels are referred to as MPLS-over-UDP tunnels. The tunnel composite next hop are enabled by default for the MPLS-over-UDP tunnels.

MPLS-over-UDP tunnels can be bidirectional or unidirectional in nature.

- Bidirectional—When the PE devices are connected over MPLS-over-UDP tunnels in both directions, it is called a bidirectional MPLS-over-UDP tunnel.
- Unidirectional—When two PE devices are connected over MPLS-over-UDP tunnel in one direction, and over MPLS/IGP in the other direction, it is called an unidirectional MPLS-over-UDP tunnel.

  Unidirectional MPLS-over-UDP tunnels are used in migration scenarios, or in cases where two PE devices provide connectivity to each other over two disjoint networks. Because reverse direction tunnel does not exist for unidirectional MPLS-over-UDP tunnels, you must configure a filter-based MPLS-over-UDP decapsulation on the remote PE device for forwarding the traffic.

Starting in Junos OS Release 18.2R1, on PTX series routers and QFX10000 with unidirectional MPLS-over-UDP tunnels, you must configure the remote PE device with an input filter for MPLS-over-UDP packets, and an action for decapsulating the IP and UDP headers for forwarding the packets in the reverse tunnel direction.

For example, on the remote PE device, Device PE2, the following configuration is required for unidirectional MPLS-over-UDP tunnels:

PE2

```
[edit firewall filter]
user@host# set Decap_Filter term udp_decap from protocol udp
user@host# set Decap_Filter term udp_decap from destination-port 6635
user@host# set Decap_Filter term udp_decap then count UDP_PKTS
user@host# set Decap_Filter term udp_decap then decapsulate mpls-in-udp
user@host# set Decap_Filter term def then count def_pkt
user@host# set Decap_Filter term def then accept
```

In the above sample configuration, *Decap_Filter* is the name of the firewall filter used for MPLS-over-UDP decapsulation. The term *udp_decap* is the input filter for accepting UDP packets on the core-facing interface of Device PE2, and then decapsulate the MPLS-over-UDP packets to MPLS-over-IP packets for forwarding.

You can use the existing firewall operational mode commands, such as `show firewall filter` to view the filter-based MPLS-over-UDP decapsulation.

For example:

```
user@host >show firewall filter Decap_Filter
Filter: Decap_Filter
Counters:
Name                                                Bytes              Packets
UDP_PKTS                                            16744                  149
def_pkt                                             13049                  136
```

NOTE: For unidirectional MPLS-over-UDP tunnels:

- Only IPv4 address is supported as the outer header. Filter-based MPLS-over-UDP decapsulation does not support IPv6 address in the outer header.
- Only the default routing instance is supported after decapsulation.

Starting in Junos OS Release 17.1, on MX Series routers with MPCs and MICs, the scaling limit of MPLS-over-UDP tunnels is increased.

Starting in Junos Release 19.2R1, on MX Series routers with MPCs and MICs, carrier supporting carrier (CSC) architecture can be deployed with MPLS-over-UDP tunnels carrying MPLS traffic over dynamic IPv4 UDP tunnels that are established between supporting carrier's PE devices. With this enhancement, the scaling advantage that the MPLS-over-UDP tunnels provided is further increased. The CSC support with MPLS-over-UDP tunnel is not supported for IPv6 UDP tunnel.

The existing dynamic tunnel feature requires complete static configuration. Currently, the tunnel information received from peer devices in advertised routes is ignored. Starting in Junos OS Release 17.4R1, on MX Series routers, the next-hop-based dynamic MPLS-over-UDP tunnels are signaled using BGP encapsulation extended community. BGP export policy is used to specify the tunnel types, advertise the sender side tunnel information, and parse and convey the receiver side tunnel information. A tunnel is created according to the received type tunnel community.

Multiple tunnel encapsulations are supported by BGP. On receiving multiple capability, the next-hop-based dynamic tunnel is created based on the configured BGP policy and tunnel preference. The tunnel preference should be consistent across both the tunnel ends for the tunnel to be set up. By default,

MPLS-over-UDP tunnel is preferred over GRE tunnels. If dynamic tunnel configuration exists, it takes precedence over received tunnel community.

When configuring a next-hop-based dynamic MPLS-over-UDP tunnel, be aware of the following considerations:

- An IBGP session must be configured between the PE devices.
- A switchover between the next-hop-based dynamic tunnel encapsulations (UDP and GRE) is allowed, and this can impact network performance in terms of the supported IP tunnel scaling values in each mode.
- Having both GRE and UDP next-hop-based dynamic tunnel encapsulation types for the same tunnel destination leads to a commit failure.
- For unidirectional MPLS-over-UDP tunnels, you must explicitly configure filter-based MPLS-over-UDP decapsulation on the remote PE device for the packets to be forwarded.
- Graceful Routing Engine switchover (GRES) is supported with MPLS-over-UDP, and the MPLS-over-UDP tunnel type flags are unified ISSU and NSR compliant.
- MPLS-over-UDP tunnels are supported on virtual MX (vMX) in Lite mode.
- MPLS-over-UDP tunnels support dynamic GRE tunnel creation based upon new IPv4-mapped-IPv6 next hops.
- MPLS-over-UDP tunnel are supported in interoperability with contrail, wherein the MPLS-over-UDP tunnels are created from the contrail vRouter to an MX gateway. To enable this, the following community is required to be advertised in the route from the MX Series router to the contrail vRouter:

  ```
  [edit policy-options community]
  udp members 0x030c:64512:13;
  ```

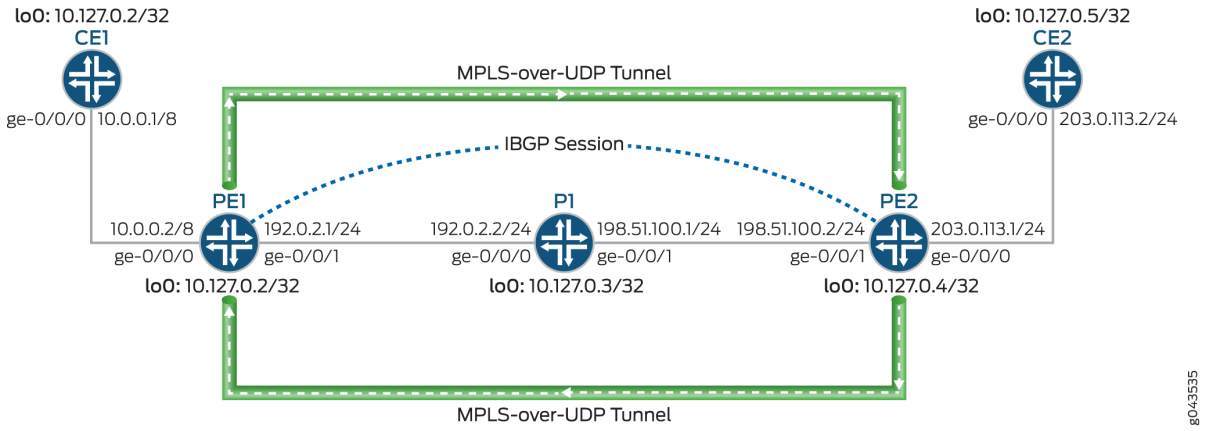
  At a given point in time, only one tunnel type is supported on the contrail vRouter—next-hop-based dynamic GRE tunnels, MPLS-over-UDP tunnels, or VXLAN.

- The following features are not supported with the next-hop-based dynamic MPLS-over-UDP tunnel configuration:
  - RSVP automatic mesh
  - Plain IPV6 GRE and UDP tunnel configuration
  - Logical systems

Topology

Figure 5 on page 118 illustrates a Layer 3 VPN scenario over dynamic MPLS-over-UDP tunnels. The customer edge (CE) devices, CE1 and CE2, connect to provider edge (PE) devices, PE1 and PE2, respectively. The PE devices are connected to a provider device (Device P1), and an internal BGP (IBGP) session interconnects the two PE devices. A dynamic next-hop-based bidirectional MPL-over-UDP tunnel is configured between the PE devices.

Figure 5: Dynamic MPLS-over-UDP Tunnels



The MPLS-over-UDP tunnel is handled as follows:

1. After a MPLS-over-UDP tunnel is configured, a tunnel destination mask route with a tunnel composite next hop is created for the tunnel in the inet.3 routing table. This IP tunnel route is withdrawn only when the dynamic tunnel configuration is deleted.

   The tunnel composite next-hop attributes include the following:

   - When Layer 3 VPN composite next hop is disabled—Source and destination address, encapsulation string, and VPN label.
   - When Layer 3 VPN composite next hop and per-prefix VPN label allocation are enabled—Source address, destination address, and encapsulation string.
   - When Layer 3 VPN composite next hop is enabled and per-prefix VPN label allocation is disabled —Source address, destination address, and encapsulation string. The route in this case is added to the other virtual routing and forwarding instance table with a secondary route.

2. The PE devices are interconnected using an IBGP session. The IBGP route next hop to a remote BGP neighbor is the protocol next hop, which is resolved using the tunnel mask route with the tunnel next hop.

3. After the protocol next hop is resolved over the tunnel composite next hop, indirect next hops with forwarding next hops are created.

4. The tunnel composite next hop is used to forward the next hops of the indirect next hops.

Configuration

IN THIS SECTION

- CLI Quick Configuration | 119
- Procedure | 121
- Results | 124

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

CE1

```
set interfaces ge-0/0/0 unit 0 family inet address 10.0.0.1/8
set interfaces lo0 unit 0 family inet address 10.127.0.1/32
set routing-options router-id 10.127.0.1
set routing-options autonomous-system 65200
set protocols bgp group ce1-pe1 export export-loopback-direct
set protocols bgp group ce1-pe1 peer-as 100
set protocols bgp group ce1-pe1 neighbor 10.0.0.2
set policy-options policy-statement export-loopback-direct term term-1 from interface lo0.0
set policy-options policy-statement export-loopback-direct term term-1 from route-filter
10.127.0.1/32 exact
set policy-options policy-statement export-loopback-direct term term-1 then accept
```

CE2

```
set interfaces ge-0/0/0 unit 0 family inet address 203.0.113.2/24
set interfaces lo0 unit 0 family inet address 10.127.0.5/32
set routing-options router-id 10.127.0.5
```

```
set routing-options autonomous-system 65200
set protocols bgp group ce1-pe1 export export-loopback-direct
set protocols bgp group ce1-pe1 peer-as 65100
set protocols bgp group ce1-pe1 neighbor 203.0.113.1
set policy-options policy-statement export-loopback-direct term term-1 from interface lo0.0
set policy-options policy-statement export-loopback-direct term term-1 from route-filter
10.127.0.5/32 exact
set policy-options policy-statement export-loopback-direct term term-1 then accept
```

PE1

```
set interfaces ge-0/0/0 unit 0 family inet address 10.0.0.2/8
set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.1/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.127.0.2/32
set routing-options static route 10.33.0/16 next-hop 192.0.2.2
set routing-options router-id 10.127.0.2
set routing-options autonomous-system 65100
set routing-options forwarding-table export pplb
set routing-options dynamic-tunnels gre next-hop-based-tunnel
set routing-options dynamic-tunnels udp-dyn-tunnel-to-pe2 source-address 10.127.0.2
set routing-options dynamic-tunnels udp-dyn-tunnel-to-pe2 udp
set routing-options dynamic-tunnels udp-dyn-tunnel-to-pe2 destination-networks 10.127.0.0/24
set protocols bgp group IBGP type internal
set protocols bgp group IBGP local-address 10.127.0.2
set protocols bgp group IBGP family inet-vpn unicast
set protocols bgp group IBGP neighbor 10.127.0.4
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set routing-instances MPLS-over-UDP-PE1 instance-type vrf
set routing-instances MPLS-over-UDP-PE1 interface ge-0/0/0.0
set routing-instances MPLS-over-UDP-PE1 route-distinguisher 10.127.0.2:1
set routing-instances MPLS-over-UDP-PE1 vrf-target target:600:1
set routing-instances MPLS-over-UDP-PE1 protocols bgp group pe1-ce1 peer-as 65200
set routing-instances MPLS-over-UDP-PE1 protocols bgp group pe1-ce1 neighbor 10.0.0.1 as-override
```

P1

```
set interfaces ge-0/0/0 unit 0 family inet address 192.0.2.2/24
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 198.51.100.1/24
```

```
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.127.0.3/32
set routing-options router-id 10.127.0.3
set routing-options autonomous-system 65100
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

PE2

```
set interfaces ge-0/0/0 unit 0 family inet address 203.0.113.1/24
set interfaces ge-0/0/1 unit 0 family inet address 198.51.100.2/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.127.0.4/8
set routing-options nonstop-routing
set routing-options router-id 10.127.0.4
set routing-options autonomous-system 65100
set routing-options forwarding-table export pplb
set routing-options dynamic-tunnels udp-dyn-tunnel-to-pe1 source-address 10.127.0.4
set routing-options dynamic-tunnels udp-dyn-tunnel-to-pe1 udp
set routing-options dynamic-tunnels udp-dyn-tunnel-to-pe1 destination-networks 10.127.0.0/24
set protocols bgp group IBGP type internal
set protocols bgp group IBGP local-address 10.127.0.4
set protocols bgp group IBGP family inet-vpn unicast
set protocols bgp group IBGP neighbor 10.127.0.2
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set routing-instances MPLS-over-UDP-PE2 instance-type vrf
set routing-instances MPLS-over-UDP-PE2 interface ge-0/0/0.0
set routing-instances MPLS-over-UDP-PE2 route-distinguisher 10.127.0.4:1
set routing-instances MPLS-over-UDP-PE2 vrf-target target:600:1
set routing-instances MPLS-over-UDP-PE2 protocols bgp group ebgp peer-as 65200
set routing-instances MPLS-over-UDP-PE2 protocols bgp group ebgp neighbor 203.0.113.2 as-override
```

Procedure

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure Device PE1:

1. Configure the device interfaces including the loopback interface of the device.

```
[edit interfaces]
user@PE1# set ge-0/0/0 unit 0 family inet address 10.0.0.2/8
user@PE1# set ge-0/0/1 unit 0 family inet address 192.0.2.1/24
user@PE1# set ge-0/0/1 unit 0 family mpls
user@PE1# set lo0 unit 0 family inet address 10.127.0.2/8
```

2. Configure a static route for routes from Device PE1 with Device P1 as the next-hop destination.

```
[edit routing-options]
user@PE1# set static route 10.33.0.0/16 next-hop 192.0.2.2
```

3. Configure the router-ID and autonomous system number for Device PE1.

```
[edit routing-options]
user@PE1# set router-id 10.127.0.2
user@PE1# set autonomous-system 65100
```

4. (PTX Series only) Configure policy control to resolve the MPLS-over-UDP dynamic tunnel route over select prefixes.

```
[edit routing-options dynamic-tunnels]
user@PTX-PE1# set forwarding-rib inet.0 inet-import dynamic-tunnel-fwd-route-import
```

5. (PTX Series only) Configure the inet-import policy for resolving dynamic tunnel destination routes over .

```
[edit policy-options]
user@PTX-PE1# set policy-statement dynamic-tunnel-fwd-route-import term 1 from route-filter
10.127.0.4/32 exact
user@PTX-PE1# set policy-statement dynamic-tunnel-fwd-route-import term 1 then accept
user@PTX-PE1# set policy-options policy-statement dynamic-tunnel-fwd-route-import then
reject
```

6. Configure IBGP peering between the PE devices.

```
[edit protocols]
user@PE1# set bgp group IBGP type internal
user@PE1# set bgp group IBGP local-address 10.127.0.2
user@PE1# set bgp group IBGP family inet-vpn unicast
user@PE1# set bgp group IBGP neighbor 10.127.0.4
```

7. Configure OSPF on all the interfaces of Device PE1, excluding the management interface.

```
[edit protocols]
user@PE1# set ospf area 0.0.0.0 interface ge-0/0/1.0
user@PE1# set ospf area 0.0.0.0 interface lo0.0 passive
```

8. Enable next-hop-based dynamic GRE tunnel configuration on Device PE1.

NOTE: This step is required only for illustrating the implementation difference between next-hop-based dynamic GRE tunnels and MPLS-over-UDP tunnels.

```
[edit routing-options]
user@PE1# set dynamic-tunnels gre next-hop-based-tunnel
```

9. Configure the MPLS-over-UDP tunnel parameters from Device PE1 to Device PE2.

```
[edit routing-options]
user@PE1# set dynamic-tunnels udp-dyn-tunnel-to-pe2 source-address 10.127.0.2
user@PE1# set dynamic-tunnels udp-dyn-tunnel-to-pe2 udp
user@PE1# set dynamic-tunnels udp-dyn-tunnel-to-pe2 destination-networks 10.127.0.0/24
```

10. Configure a VRF routing instance on Device PE1 and other routing instance parameters.

```
[edit routing-instances]
user@PE1# set MPLS-over-UDP-PE1 instance-type vrf
user@PE1# set MPLS-over-UDP-PE1 interface ge-0/0/0.0
user@PE1# set MPLS-over-UDP-PE1 route-distinguisher 10.127.0.2:1
user@PE1# set MPLS-over-UDP-PE1 vrf-target target:600:1
```

11. Enable BGP in the routing instance configuration for peering with Device CE1.

```
[edit routing-instances]
user@PE1# set MPLS-over-UDP-PE1 protocols bgp group pe1-ce1 peer-as 65200
user@PE1# set MPLS-over-UDP-PE1 protocols bgp group pe1-ce1 neighbor 10.0.0.1 as-override
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, `show protocols`, and `show routing-instances` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@PE1# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 10.0.0.2/8;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 192.0.2.1/24;
        }
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.127.0.2/32;
        }
    }
}
```

```
user@PE1# show routing-options
static {
    route 10.33.0.0/16 next-hop 192.0.2.2;
```

```
}
router-id 10.127.0.2;
autonomous-system 65100;
forwarding-table {
    export pplb;
}
dynamic-tunnels {
    gre next-hop-based-tunnel;
    udp-dyn-tunnel-to-pe2 {
        source-address 10.127.0.2;
        udp;
        destination-networks {
            10.127.0.0/24;
        }
    }
}
```

```
user@PE1# show protocols
bgp {
    group IBGP {
        type internal;
        local-address 10.127.0.2;
        family inet-vpn {
            unicast;
        }
        neighbor 10.127.0.4;
    }
}
ospf {
    area 0.0.0.0 {
        interface ge-0/0/1.0;
        interface lo0.0 {
            passive;
        }
    }
}
```

```
user@PE1# show routing-instances
MPLS-over-UDP-PE1 {
    instance-type vrf;
```

```
    interface ge-0/0/0.0;
    route-distinguisher 10.127.0.2:1;
    vrf-target target:600:1;
    protocols {
        bgp {
            group pe1-ce1 {
                peer-as 65200;
                neighbor 10.0.0.1 {
                    as-override;
                }
            }
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying the Connection Between PE Devices | **126**
- Verify the Dynamic Tunnel Routes on Device PE1 | **127**
- Verify the Dynamic Tunnel Routes on Device PE2 | **129**
- Verifying That the Routes Have the Expected Indirect-Next-Hop Flag | **130**

Confirm that the configuration is working properly.

Verifying the Connection Between PE Devices

Purpose

Verify the BGP peering status between Device PE1 and Device PE2, and the BGP routes received from Device PE2.

Action

From operational mode, run the **show bgp summary** and **show route receive-protocol bgp *ip-address* table bgp.l3vpn.0** commands.

```
user@PE1> show bgp summary
Groups: 2 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l3vpn.0
                       2          2          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
10.127.0.4             65100        139        136       0       0       58:23 Establ
  bgp.l3vpn.0: 2/2/2/0
  MPLS-over-UDP-PE1.inet.0: 2/2/2/0
10.10.0.1               65200        135        136       0       0       58:53 Establ
  MPLS-over-UDP-PE1.inet.0: 1/1/1/0
```

```
user@PE1> show route receive-protocol bgp 10.127.0.4 table bgp.l3vpn.0
bgp.l3vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
  10.127.0.4:1:127.0.0.5/8
*                         10.127.0.4                 65100        65200 I
```

Meaning

- In the first output, the BGP session state is `Establ`, which means that the session is up and the PE devices are peered.
- In the second output, Device PE1 has learned a BGP route from Device PE2.

Verify the Dynamic Tunnel Routes on Device PE1

Purpose

Verify the routes in the inet.3 routing table and the dynamic tunnel database information on Device PE1.

Action

From operational mode, run the **show route table inet.3**, **show dynamic-tunnels database terse**, **show dynamic-tunnels database**, and **show dynamic-tunnels database summary** commands.

```
user@PE1> show route table inet.3
inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.127.0.0/24     *[Tunnel/300] 00:21:18
                      Tunnel
127.0.0.4/8     *[Tunnel/300] 00:21:18
                      Tunnel Composite
```

```
user@PE1> show dynamic-tunnels database terse
Table: inet.3

Destination-network: 10.127.0.0/24
Destination         Source          Next-hop                Type        Status
10.127.0.4/8      10.127.0.2     0xb395b10 nhid 613        udp         Up
```

```
user@PE1> show dynamic-tunnels database
Table: inet.3
. . .
Tunnel to: 10.127.0.4/32
  Reference count: 2
  Next-hop type: UDP
    Source address: 10.127.0.2 Tunnel Id: 2
    Next hop: tunnel-composite, 0xb395b10, nhid 613
      VPN Label: Push 299776 Reference count: 3
      Traffic Statistics: Packets 0, Bytes 0
      State: Up
```

```
user@PE1> show dynamic-tunnels database summary
Dynamic Tunnels,  Total 1 displayed
GRE Tunnel:
Active Tunnel Mode, Next Hop Base
  IFL Based,  Total 0 displayed, Up 0, Down 0
  Nexthop Based,  Total 0 displayed, Up 0, Down 0
```

```
RSVP Tunnel:
  Total 0 displayed
UDP Tunnel:
  Total 1 displayed, Up 1, Down 0
```

Meaning

- In the first output, because Device PE1 is configured with the MPLS-over-UDP tunnel, a tunnel composite route is created for the inet.3 routing table route entry.
- In the remaining outputs, the MPLS-over-UDP tunnel is displayed with the tunnel encapsulation type, tunnel next hop parameters, and tunnel status.

Verify the Dynamic Tunnel Routes on Device PE2

Purpose

Verify the routes in the inet.3 routing table and the dynamic tunnel database information on Device PE2.

Action

From operational mode, run the **show route table inet.3**, and the **show dynamic-tunnels database terse** commands.

```
user@PE2> show route table inet.3
inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.127.0.0/24      *[Tunnel/300] 00:39:31
                       Tunnel
10.127.0.2/32     *[Tunnel/300] 00:24:53
                       Tunnel Composite
```

```
user@PE1> show dynamic-tunnels database terse
Table: inet.3

Destination-network: 127.0.0.0/8
```

```
Destination         Source          Next-hop                Type        Status
10.127.0.2/32       10.127.0.4      0xb395450 nhid 615      udp         Up
```

Meaning

The outputs show the MPLS-over-UDP tunnel creation and the next-hop ID assigned as the next-hop interface, similar to Device PE1.

Verifying That the Routes Have the Expected Indirect-Next-Hop Flag

Purpose

Verify that Device PE1 and Device PE2 are configured to maintain the indirect next hop to forwarding next-hop binding on the Packet Forwarding Engine forwarding table.

Action

From operational mode, run the **show krt indirect-next-hop** command on Device PE1 and Device PE2.

```
user@PE1> show krt indirect-next-hop
Indirect Nexthop:
Index: 1048574 Protocol next-hop address: 10.127.0.4
  RIB Table: bgp.l3vpn.0
  Label: Push 299776
  Policy Version: 1                     References: 1
  Locks: 3                              0xb2ab630
  Flags: 0x0
  INH Session ID: 0x0
  INH Version ID: 0
  Ref RIB Table: unknown
        Tunnel type: UDP, Reference count: 3, nhid: 613
        Destination address: 10.127.0.4, Source address: 10.127.0.2
        Tunnel id: 2, VPN Label: Push 299776, TTL action: prop-ttl
      IGP FRR Interesting proto count : 1
      Chain IGP FRR Node Num          : 1
         IGP Resolver node(hex)       : 0xb3c70dc
```

```
        IGP Route handle(hex)        : 0xb1ae688      IGP rt_entry protocol        : Tunnel
        IGP Actual Route handle(hex) : 0x0            IGP Actual rt_entry protocol : Any
```

```
user@PE2> show krt indirect-next-hop
Indirect Nexthop:
Index: 1048575 Protocol next-hop address: 10.127.0.2
  RIB Table: bgp.l3vpn.0
  Label: Push 299776
  Policy Version: 1                     References: 2
  Locks: 3                              0xb2ab740
  Flags: 0x0
  INH Session ID: 0x0
  INH Version ID: 0
  Ref RIB Table: unknown
        Tunnel type: UDP, Reference count: 3, nhid: 615
        Destination address: 10.127.0.2, Source address: 10.127.0.4
        Tunnel id: 1, VPN Label: Push 299776, TTL action: prop-ttl
      IGP FRR Interesting proto count : 2
      Chain IGP FRR Node Num          : 1
         IGP Resolver node(hex)       : 0xb3d3a28
         IGP Route handle(hex)        : 0xb1ae634      IGP rt_entry protocol        : Tunnel
         IGP Actual Route handle(hex) : 0x0            IGP Actual rt_entry protocol : Any
```

Meaning

The outputs show that a next-hop-based dynamic MPLS-over-UDP tunnel is created between the PE devices.

Troubleshooting

IN THIS SECTION

- Troubleshooting Commands | 132

To troubleshoot the next-hop-based dynamic tunnels, see:

Troubleshooting Commands

Problem

The next-hop-based dynamic MPLS-over-UDP tunnel configuration is not taking effect.

Solution

To troubleshoot the next-hop-based MPLS-over-UDP tunnel configuration, use the following `traceroute` commands at the `[edit routing-options dynamic-tunnels]` statement hierarchy:

- `traceoptions file` *`file-name`*
- `traceoptions file size` *`file-size`*
- `traceoptions flag all`

For example:

```
[edit routing-options dynamic-tunnels]
traceoptions {
    file udp_dyn_pe1.wri size 4294967295;
    flag all;
}
```

Anti-Spoofing Protection for Next-Hop-Based Dynamic Tunnels Overview

With the rise in deployment of high-scale IP tunnels in data centers, there is a need to add security measures that allow users to limit malicious traffic from compromised virtual machines (VMs). One possible attack is the injecting of traffic into an arbitrary customer VPN from a compromised server through the gateway router. In such cases, anti-spoofing checks on IP tunnels ensure that only legitimate sources are injecting traffic into data centers from their designated IP tunnels.

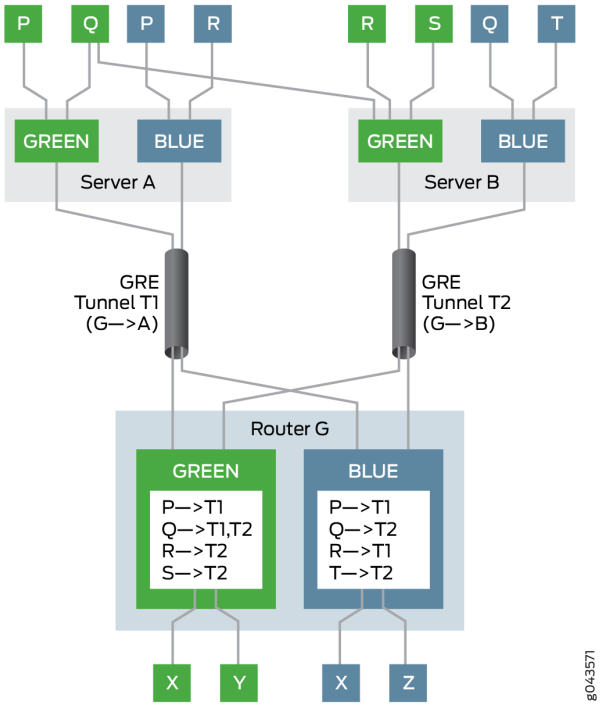
Next-hop-based dynamic IP tunnels create a tunnel composite next hop for every dynamic tunnel created on the device. Because next-hop-based dynamic tunnels remove the dependency on physical interfaces for every new dynamic tunnel configured, configuring next-hop-based dynamic tunnels provides a scaling advantage over the number of dynamic tunnels that can be created on a device. Starting in Junos OS Release 17.1, anti-spoofing capabilities for next-hop-based dynamic IP tunnels is provided for next-hop-based dynamic tunnels. With this enhancement, a security measure is implemented to prevent injecting of traffic into an arbitrary customer VPN from a compromised server through the gateway router.

Anti-spoofing is implemented using reverse path forwarding checks in the Packet Forwarding Engine. The checks are implemented for the traffic coming through the tunnel to the routing instance. Currently,

when the gateway router receives traffic from a tunnel, only the destination lookup us done and the packet is forwarded accordingly. When anti-spoofing protection is enabled, the gateway router also does a source address lookup of the encapsulation packet IP header in the VPN, in addition to the tunnel destination lookup. This ensures that legitimate sources are injecting traffic through their designated IP tunnels. As a result, anti-spoofing protection ensures that the tunnel traffic is received from a legitimate source on the designated tunnels.

Figure 6 on page 133 illustrates a sample topology with the requirements for anti-spoofing protection.

Figure 6: Anti-Spoofing Protection for Next-Hop-Based Dynamic Tunnels



In this example, the gateway router is Router G. Router G has two VPNs—Green and Blue. The two servers, Server A and Server B, can reach the Green and Blue VPNs on Router G through the next-hop-based dynamic tunnels T1 and T2, respectively. Several hosts and virtual machines (P, Q, R, S, and T) connected to the servers can reach the VPNs through the gateway router, Router G. Router G has the virtual routing and forwarding (VRF) tables for Green and Blue VPNs, each populated with the reachability information for the virtual machines in those VPNs.

For example, in VPN Green, Router G uses tunnel T1 to reach host P, tunnel T2 to reach hosts R and S, and load balancing is done between tunnels T1 and T2 to reach the multihomed host Q. In VPN Blue, Router G uses tunnel T1 to reach hosts P and R, and tunnel T2 to reach hosts Q and T.

The check passes for reverse path forwarding when:

- A packet comes from a legitimate source on its designated tunnel.

  Host P in VPN Green sends a packet to host X using tunnel T1. Because Router G can reach host P through tunnel T1, it allows the packet to pass and forwards the packet to host X.

- A packet comes from a multihomed source on its designated tunnels.

  Host Q in VPN Green is multihomed on servers A and B, and can reach Router G through tunnels T1 and T2. Host Q sends a packet to host Y using tunnel T1, and a packet to host X using tunnel T2. Because Router G can reach host Q through tunnels T1 and T2, it allows the packets to pass and forwards them to hosts Y and X, respectively.

Layer 3 VPNs do not have anti-spoofing protection enabled by default. To enable anti-spoofing for next-hop-based dynamic tunnels, include the `ip-tunnel-rpf-check` statement at the `[edit routing-instances` *`routing-instance-name`* `routing-options forwarding-table]` hierarchy level. The reverse path forwarding check is applied to the VRF routing instance only. The default mode is set to `strict`, where the packet that comes from a source on a nondesignated tunnel does not pass the check. The `ip-tunnel-rpf-check` mode can be set as `loose`, where the reverse path forwarding check fails when the packet comes from a nonexistent source. An optional firewall filter can be configured under the `ip-tunnel-rpf-check` statement to count and log the packets that failed the reverse path forwarding check.

The following sample output shows an anti-spoofing configuration:

```
[edit routing-instances routing-instance-name routing-options forwarding-table]
ip-tunnel-rpf-check {
    mode loose;
    fail-filter filter-name;
}
```

Take the following guidelines under consideration when configuring anti-spoofing protection for next-hop-based dynamic tunnels:

- Anti-spoofing protection can be enabled for IPv4 tunnels and IPv4 data traffic only. The anti-spoofing capabilities are not supported on IPv6 tunnels and IPv6 data traffic.

- Anti-spoofing for next-hop-based dynamic tunnels can detect and prevent a compromised virtual machine (inner source reverse path forwarding check) but not a compromised server that is label-spoofing.

- The next-hop-based IP tunnels can originate and terminate on an inet.0 routing table.

- Anti-spoofing protection is effective when the VRF routing instance has label-switched interfaces (LSIs) (using the `vrf-table-label`), or virtual tunnel (VT) interfaces. With `per-next-hop` label on the VRF routing instance, anti-spoofing protection is not supported.

- The `rpf fail-filter` is applicable only to the inner IP packet.
- Enabling anti-spoofing checks does not affect the scaling limit of the next-hop-based dynamic tunnels on a device.
- The system resource utilization with anti-spoofing protection enabled for the VRF routing instance is slightly higher than the utilization of next-hop-based dynamic tunnels without the anti-spoofing protection enabled.
- Anti-spoofing protection requires additional source IP address checks, which has minimal impact on network performance.
- Graceful Routing Engine switchover (GRES) and in-service software upgrade (ISSU) are supported with anti-spoofing protection.

Example: Configuring Anti-Spoofing Protection for Next-Hop-Based Dynamic Tunnels

IN THIS SECTION

- Requirements | **135**
- Overview | **136**
- Configuration | **138**
- Verification | **146**

This example shows how to configure reverse path forwarding checks for the virtual routing and forwarding (VRF) routing instance to enable anti-spoofing protection for next-hop-based dynamic tunnels. The checks ensure that legitimate sources are injecting traffic through their designated IP tunnels.

Requirements

This example uses the following hardware and software components:

- Three MX Series Routers with MICs, each connected to a host device.
- Junos OS Release 17.1 or later running on one or all the routers.

Before you begin:

- Enable tunnel services configuration on the Flexible PIC Concentrator.
- Configure the router interfaces.

- Configure the router-ID and assign an autonomous system number for the router.
- Establish an internal BGP (IBGP) session with the tunnel endpoints.
- Configure RSVP on all the routers.
- Configure OSPF or any other interior gateway protocol on all the routers.
- Configure two dynamic next-hop-based IP tunnels between the two routers.
- Configure a VRF routing instance for every router-to-host connection.

Overview

IN THIS SECTION

Topology | 136

Starting in Junos OS Release 17.1, anti-spoofing capabilities are added to next-hop-based dynamic IP tunnels, where checks are implemented for the traffic coming through the tunnel to the routing instance using reverse path forwarding in the Packet Forwarding Engine.

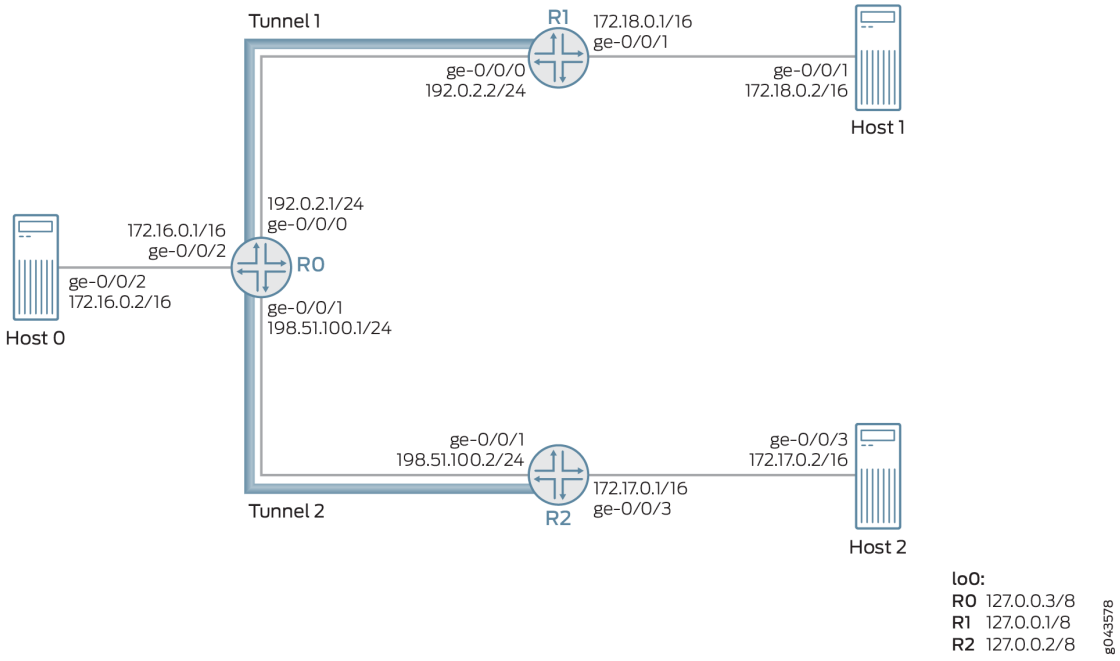
Currently, when the gateway router receives traffic from a tunnel, only the destination address lookup is done before forwarding. With anti-spoofing protection, the gateway router does a source address lookup of the encapsulation packet IP header in the VPN to ensure that legitimate sources are injecting traffic through their designated IP tunnels. This is called the strict mode and is the default behavior of anti-spoofing protection. To pass traffic from nondesignated tunnels, the reverse path forwarding check is enabled in the lose mode. For traffic received from nonexistent sources, the reverse path forwarding check fails for both the strict and loose modes.

Anti-spoofing is supported on VRF routing instances. To enable anti-spoofing for dynamic tunnels, include the `ip-tunnel-rpf-check` statement at the `[edit routing-instances routing-instance-name routing-options forwarding-table]` hierarchy level.

Topology

Figure 7 on page 137 illustrates a sample network topology enabled with anti-spoofing protection. Routers R0, R1 and R2 are each connected to hosts Host0, Host1, and Host2, respectively. Two generic routing encapsulation (GRE) next-hop-based dynamic tunnels, Tunnel 1 and Tunnel 2 – connect Router R0 with Routers R1 and R2, respectively. The VRF routing instance is running between each router and its connected host devices.

Figure 7: Anti-Spoofing Protection for Next-Hop-Based Dynamic Tunnels



Taking as an example, three packets (Packets A, B, and C) are received on Router 0 from Router R2 through the next-hop-based dynamic GRE tunnel (Tunnel 2). The source IP address of these packets are 172.17.0.2 (Packet A), 172.18.0.2 (Packet B), and 172.20.0.2 (Packet C).

The source IP address of Packets A and B belong to Host 2 and Host 1, respectively. Packet C is a nonexistent source tunnel. The designated tunnel in this example is Tunnel 2, and the nondesignated tunnel is Tunnel 1. Therefore, the packets are processed as follows:

- **Packet A**—Because the source is coming from a designated tunnel (Tunnel 2), Packet A passes the reverse path forwarding check and is processed for forwarding through Tunnel 2.
- **Packet B**—Because the source is coming from Tunnel 1, which is a nondesinated tunnel, by default, Packet B fails the reverse path forwarding check in the strict mode. If loose mode is enabled, Packet B is allowed for forwarding.
- **Packet C**—Because the source is a nonexistent tunnel source, Packet C fails the reverse path forwarding check, and the packet is not forwarded.

Configuration

IN THIS SECTION

- CLI Quick Configuration | 138
- Procedure | 140
- Results | 143

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

Router R0

```
set interfaces ge-0/0/0 unit 0 family inet address 192.0.2.1/24
set interfaces ge-0/0/1 unit 0 family inet address 198.51.100.1/24
set interfaces ge-0/0/2 vlan-tagging
set interfaces ge-0/0/2 unit 0 vlan-id 1
set interfaces ge-0/0/2 unit 0 family inet address 172.16.0.1/16
set interfaces lo0 unit 0 family inet address 10.1.1.1/32
set routing-options router-id 10.1.1.1
set routing-options autonomous-system 100
set routing-options dynamic-tunnels gre next-hop-based-tunnel
set routing-options dynamic-tunnels T1 source-address 192.0.2.1
set routing-options dynamic-tunnels T1 gre
set routing-options dynamic-tunnels T1 destination-networks 192.0.2.0/24
set routing-options dynamic-tunnels T2 source-address 198.51.100.1
set routing-options dynamic-tunnels T2 gre
set routing-options dynamic-tunnels T2 destination-networks 198.51.100.0/24
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols bgp group IBGP type internal
set protocols bgp group IBGP local-address 10.1.1.1
set protocols bgp group IBGP family inet-vpn unicast
set protocols bgp group IBGP neighbor 20.1.1.1
set protocols bgp group IBGP neighbor 30.1.1.1
set protocols ospf traffic-engineering
```

```
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface all
set routing-instances VPN1 instance-type vrf
set routing-instances VPN1 interface ge-0/0/2.0
set routing-instances VPN1 route-distinguisher 100:100
set routing-instances VPN1 vrf-target target:100:1
set routing-instances VPN1 vrf-table-label
set routing-instances VPN1 routing-options forwarding-table ip-tunnel-rpf-check mode strict
set routing-instances VPN1 protocols bgp group External type external
set routing-instances VPN1 protocols bgp group External family inet unicast
set routing-instances VPN1 protocols bgp group External peer-as 200
set routing-instances VPN1 protocols bgp group External neighbor 172.16.0.1
```

Router R1

```
set interfaces ge-0/0/0 unit 0 family inet address 192.0.2.2/24
set interfaces ge-0/0/1 vlan-tagging
set interfaces ge-0/0/1 unit 0 vlan-id 2
set interfaces ge-0/0/1 unit 0 family inet address 172.18.0.1/16
set interfaces lo0 unit 0 family inet address 20.1.1.1/32
set routing-options router-id 20.1.1.1
set routing-options autonomous-system 100
set routing-options dynamic-tunnels gre next-hop-based-tunnel
set routing-options dynamic-tunnels T1 source-address 192.0.2.2
set routing-options dynamic-tunnels T1 gre
set routing-options dynamic-tunnels T1 destination-networks 192.0.2.0/24
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols bgp group IBGP type internal
set protocols bgp group IBGP local-address 20.1.1.1
set protocols bgp group IBGP family inet-vpn unicast
set protocols bgp group IBGP neighbor 30.1.1.1
set protocols bgp group IBGP neighbor 10.1.1.1
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface all
set routing-instances VPN2 instance-type vrf
set routing-instances VPN2 interface ge-0/0/1.0
set routing-instances VPN2 route-distinguisher 100:200
set routing-instances VPN2 vrf-target target:200:1
set routing-instances VPN2 vrf-table-label
```

R2

```
set interfaces ge-0/0/1 unit 0 family inet address 198.51.100.2/24
set interfaces ge-0/0/2 vlan-tagging
set interfaces ge-0/0/2 unit 0 vlan-id 3
set interfaces ge-0/0/2 unit 0 family inet address 172.17.0.1/16
set interfaces lo0 unit 0 family inet address 30.1.1.1/32
set routing-options router-id 30.1.1.1
set routing-options autonomous-system 100
set routing-options dynamic-tunnels gre next-hop-based-tunnel
set routing-options dynamic-tunnels T2 source-address 198.51.100.2
set routing-options dynamic-tunnels T2 gre
set routing-options dynamic-tunnels T2 destination-networks 198.51.100.0/24
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols bgp group IBGP type internal
set protocols bgp group IBGP local-address 30.1.1.1
set protocols bgp group IBGP family inet-vpn unicast
set protocols bgp group IBGP neighbor 20.1.1.1
set protocols bgp group IBGP neighbor 10.1.1.1
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface all
set routing-instances VPN3 instance-type vrf
set routing-instances VPN3 interface ge-0/0/2.0
set routing-instances VPN3 route-distinguisher 100:300
set routing-instances VPN3 vrf-target target:300:1
set routing-instances VPN3 vrf-table-label
```

Procedure

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure Router R0:

1. Configure Router R0's interfaces, including the loopback interface.

```
[edit interfaces]
user@R0# set ge-0/0/0 unit 0 family inet address 192.0.2.1/24
user@R0# set ge-0/0/1 unit 0 family inet address 198.51.100.1/24
user@R0# set ge-0/0/2 vlan-tagging
user@R0# set ge-0/0/2 unit 0 vlan-id 1
user@R0# set ge-0/0/2 unit 0 family inet address 172.16.0.1/16
user@R0# set lo0 unit 0 family inet address 10.1.1.1/32
```

2. Assign the router ID and autonomous system number for Router R0.

```
[edit routing-options]
user@R0# set router-id 10.1.1.1
user@R0# set autonomous-system 100
```

3. Configure IBGP peering between the routers.

```
[edit protocols]
user@R0# set bgp group IBGP type internal
user@R0# set bgp group IBGP local-address 10.1.1.1
user@R0# set bgp group IBGP family inet-vpn unicast
user@R0# set bgp group IBGP neighbor 20.1.1.1
user@R0# set bgp group IBGP neighbor 30.1.1.1
```

4. Configure OSPF on all the interfaces of Router R0, excluding the management interface.

```
[edit protocols]
user@R0# set ospf traffic-engineering
user@R0# set ospf area 0.0.0.0 interface lo0.0 passive
user@R0# set ospf area 0.0.0.0 interface all
```

5. Configure RSVP on all the interfaces of Router R0, excluding the management interface.

```
[edit protocols]
user@R0# set rsvp interface all
user@R0# set rsvp interface fxp0.0 disable
```

6. Enable next-hop-based dynamic GRE tunnel configuration on Router R0.

```
[edit routing-options]
user@R0# set dynamic-tunnels gre next-hop-based-tunnel
```

7. Configure the dynamic GRE tunnel parameters from Router R0 to Router R1.

```
[edit routing-options]
user@R0# set dynamic-tunnels T1 source-address 192.0.2.1
user@R0# set dynamic-tunnels T1 gre
user@R0# set dynamic-tunnels T1 destination-networks 192.0.2.0/24
```

8. Configure the dynamic GRE tunnel parameters from Router R0 to Router R2.

```
[edit routing-options]
user@R0# set dynamic-tunnels T2 source-address 198.51.100.1
user@R0# set dynamic-tunnels T2 gre
user@R0# set dynamic-tunnels T2 destination-networks 198.51.100.0/24
```

9. Configure a virtual routing and forwarding (VRF) routing instance on Router R0, and assign the interface connecting to Host 1 to the VRF instance.

```
[edit routing-instances]
user@R0# set VPN1 instance-type vrf
user@R0# set VPN1 route-distinguisher 100:100
user@R0# set VPN1 vrf-target target:100:1
user@R0# set VPN1 vrf-table-label
user@R0# set VPN1 interface ge-0/0/2.0
```

10. Configure an external BGP session with Host 1 for the VRF routing instance.

```
[edit routing-instances]
user@R0# set VPN1 protocols bgp group External type external
user@R0# set VPN1 protocols bgp group External family inet unicast
user@R0# set VPN1 protocols bgp group External peer-as 200
user@R0# set VPN1 protocols bgp group External neighbor 172.16.0.1
```

11. Configure anti-spoofing protection for the VRF routing instance on Router R0. This enables reverse path forwarding check for the next-hop-based dynamic tunnels, T1 and T2, on Router 0.

```
[edit routing-instances]
user@R0# set VPN1 routing-options forwarding-table ip-tunnel-rpf-check mode strict
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, `show protocols`, and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R0# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 192.0.2.1/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 198.51.100.1/24;
        }
    }
}
ge-0/0/2 {
    vlan-tagging;
    unit 0 {
        vlan-id 1;
        family inet {
            address 172.16.0.1/16;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.1.1.1/32;
        }
```

```
    }
}
```

```
user@R0# show routing-options
router-id 10.1.1.1;
autonomous-system 100;
dynamic-tunnels {
    gre next-hop-based-tunnel;
    T1 {
        source-address 192.0.2.1;
        gre;
        destination-networks {
            192.0.2.0/24;
        }
    }
    T2 {
        source-address 198.51.100.1;
        gre;
        destination-networks {
            198.51.100.0/24;
        }
    }
}
```

```
user@R0# show protocols
rsvp {
    interface all;
    interface fxp0.0 {
        disable;
    }
}
bgp {
    group IBGP {
        type internal;
        local-address 10.1.1.1;
        family inet-vpn {
            unicast;
        }
        neighbor 20.1.1.1;
        neighbor 30.1.1.1;
```

```
    }
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface lo0.0 {
            passive;
        }
        interface all;
    }
}
```

```
user@R0# show routing-instances
VPN1 {
    instance-type vrf;
    interface ge-0/0/2.0;
    route-distinguisher 100:100;
    vrf-target target:100:1;
    vrf-table-label;
    routing-options {
        forwarding-table {
            ip-tunnel-rpf-check {
                mode strict;
            }
        }
    }
    protocols {
        bgp {
            group External {
                type external;
                family inet {
                    unicast;
                }
                peer-as 200;
                neighbor 172.16.0.1;
            }
        }
    }
}
```

Verification

IN THIS SECTION

- Verifying Basic Configuration | 146
- Verifying Dynamic Tunnel Configuration | 147
- Verifying Anti-Spoofing Protection Configuration | 148

Confirm that the configuration is working properly.

Verifying Basic Configuration

Purpose

Verify the OSPF and BGP peering status between the Router R0 and Routers R1 and R2.

Action

From operational mode, run the **show ospf neighbor** and **show bgp summary**commands.

```
user@R0> show ospf neighbor
Address          Interface              State     ID               Pri  Dead
192.0.2.2        ge-0/0/0.0             Full      20.1.1.1         128    32
198.51.100.2     ge-0/0/1.0             Full      30.1.1.1         128    32

user@R0> show bgp summary
Groups: 2 Peers: 3 Down peers: 1
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l3vpn.0
                       0          0          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
20.1.1.1                100        182        178       0       0     1:20:27 Establ
  bgp.l3vpn.0: 0/0/0/0
30.1.1.1                100        230        225       0       0     1:41:51 Establ
  bgp.l3vpn.0: 0/0/0/0
172.16.0.1              200          0          0       0       0     1:42:08 Establ
```

Meaning

The OSPF and BGP sessions are up and running between the Routers R0, R1, and R2.

Verifying Dynamic Tunnel Configuration

Purpose

Verify the status of the next-hop-based dynamic GRE tunnels between the Router R0 and Routers R1 and R2.

Action

From operational mode, run the **show route table inet.3**, and the **show dynamic-tunnels database terse** commands.

```
user@R0> show route table inet.3

inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.0.2.0/24       *[Tunnel/300] 01:47:57
                       Tunnel
192.0.2.2/24       *[Tunnel/300] 01:47:57
                       Tunnel Composite

198.51.100.0/24    *[Tunnel/300] 01:47:57
                       Tunnel
198.51.100.2/24    *[Tunnel/300] 01:47:57
                       Tunnel Composite
```

```
user@R0> show dynamic-tunnels database terse
Table: inet.3

Destination-network: 192.0.2.0/24
Destination             Source             Next-hop                      Type         Status
192.0.2.2/24      192.0.2.1     0xb395e70 nhid 612         gre       Up

Destination-network: 198.51.100.0/24
Destination             Source                     Next-hop                     Type
```

```
Status
198.51.100.2      198.51.100.1     0xb395e70 nhid 612         gre       Up
```

Meaning

The two next-hop-based dynamic GRE tunnels, Tunnel 1 and Tunnel 2, are up.

Verifying Anti-Spoofing Protection Configuration

Purpose

Verify that the reverse path forwarding check has been enabled on the VRF routing instance on Router R0.

Action

From the operational mode, run the **show krt table VPN1.inet.0 detail**.

```
user@R0> show krt table VPN1.inet.0 detail
KRT tables:
VPN1.inet.0                  : GF: 1 krt-index: 8     ID: 0 kernel-id: 8
        flags: (null)
        tunnel rpf config data : enable, strict, filter [0], 0x2
        tunnel rpf tlv data : enable, strict, filter [0], 0x4
        unicast reverse path: disabled
        fast-reroute-priority: 0
        Permanent NextHops
          Multicast        : 0 Broadcast : 0
          Receive          : 0   Discard   : 0
          Multicast Discard: 0   Reject    : 0
          Local            : 0   Deny      : 0
          Table            : 0
```

Meaning

The configured reverse path forwarding check is enabled on the VRF routing instance in the strict mode.

Next-Hop-Based Dynamic Tunnel Localization Overview

IN THIS SECTION

- Benefits of Next-Hop-Based Dynamic Tunnel Localization | **149**
- Use Cases for Next-Hop-Based Dynamic Tunnel Localization | **149**
- Traffic Handling with Localization of Next-Hop-Based Dynamic Tunnels | **150**
- Configuring Next-Hop-Based Dynamic Tunnels Localization | **150**
- Troubleshooting Localized Next-Hop-Based Dynamic Tunnels | **153**
- Unsupported Features for Next-Hop-Based Dynamic Tunnels Localization | **155**

Next-hop-based dynamic tunnels include generic routing encapsulation (GRE) tunnels and MPLS-over-UDP tunnels. These tunnels provide a scaling advantage over the interface-based tunnels. However, unlike the interface-based tunnels, the next-hop-based dynamic tunnels are anchorless in nature, where the forwarding information of the tunnels is distributed to the Packet Forwarding Engines (PFEs) on every line card on the device. This limits the maximum number of tunnels supported on the device to the tunnel capacity of a single line card. With the support for localization, you can configure next-hop-based dynamic tunnel localization to create the forwarding information only on the PFE of a line card that is designated as the anchor PFE. The PFEs on the other line cards on the device have state forwarding information to steer the packets to the anchor PFE. This provides a scaling advantage by increasing the maximum number of tunnels supported on a device.

Benefits of Next-Hop-Based Dynamic Tunnel Localization

Provides a scaling advantage by increasing the maximum number of tunnels supported on a device.

Use Cases for Next-Hop-Based Dynamic Tunnel Localization

- The IPsec gateway devices that host a number of MS-MPC are used to terminate IPSec tunnels and are required to support moderate load. This support is affected with the use of next-hop-based dynamic tunnels when the scaling limit of the device is reached. With the localization of next-hop-based dynamic tunnels, the maximum number of the tunnels supported is increased, allowing the device to accommodate more tunnels at the cost of an extra fabric hop.
- For Internet or VPN gateway devices, such as a virtual public cloud data center, there is a need for the gateway devices to communicate with a large number of servers. The data center servers are reachable through next-hop-based dynamic tunnels. The anchorless property of the dynamic tunnels limits the overall scaling numbers of the device. The gateway devices host multiple MPCs, with

increased traffic demands. With the localization of the next-hop-based dynamic tunnels, the tunnels can be spread across the MPCs, thereby facilitating an increase in the tunnel scaling numbers.

Traffic Handling with Localization of Next-Hop-Based Dynamic Tunnels

With support for localization, the next-hop-based dynamic tunnel state is localized to an anchor Packet Forwarding Engine, and the other Packet Forwarding Engine has the tunnel state for steering traffic to the tunnel anchor.

Figure 8 on page 150 illustrates the forwarding path of next-hop-based dynamic tunnels without localization.

Figure 8: Forwarding Path of Next-Hop-Based Dynamic Tunnels Without Localization

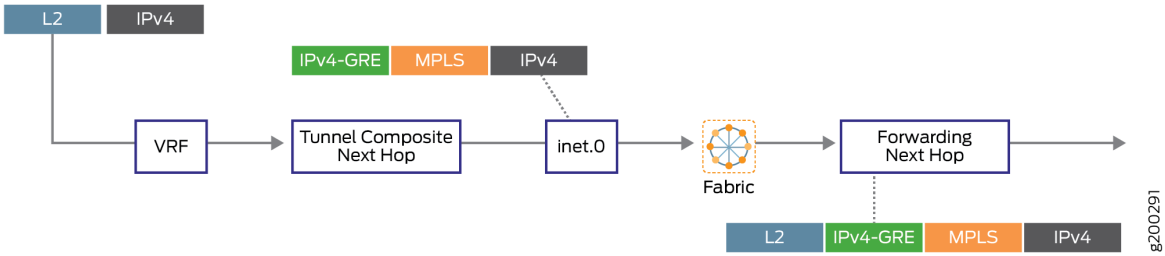
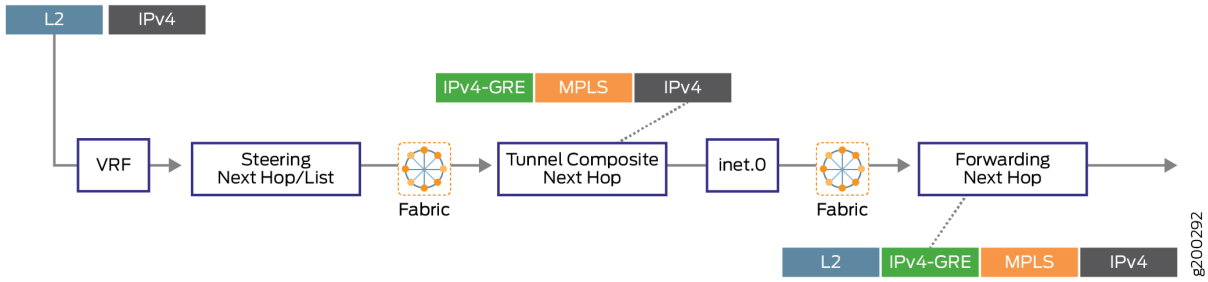


Figure 9 on page 150 illustrates the forwarding path of next-hop-based dynamic tunnels with localization.

Figure 9: Forwarding Path of Next-Hop-Based Dynamic Tunnels With Localization



Configuring Next-Hop-Based Dynamic Tunnels Localization

Localization support can be configured for newly created next-hop-based dynamic tunnels, or for existing non-local dynamic tunnels.

Configuring Localization for New Next-Hop-Based Dynamic Tunnels

The localization of next-hop-based dynamic tunnels uses a policy-based approach to specify prefix groups. In other words, route policies are used to apply the localization properties to the next-hop-based dynamic tunnels. Dynamic tunnel attribute profiles are created and configured under routing options for association with the prefix group using the policy.

1. Creating dynamic tunnel profiles.

   The dynamic tunnel profile specifies the tunnel type and the anchor Packet Forwarding Engine information. Multiple dynamic tunnel profiles can be created for localization of the dynamic tunnels. The values for the dynamic tunnel type can be GRE, UDP, or BGP-SIGNAL.

   Although BGP-SIGNAL is not a valid tunnel type, on assigning BGP-SIGNAL as the tunnel type, the tunnels created from the BGP-signalled attributes are localized. When using BGP-SIGNAL, the tunnel type is decided based on the type advertised by BGP in its TLV. BGP-SIGNAL tunnels are always next-hop-based tunnels. The GRE tunnels created dynamically by BGP-SIGNAL are always next-hop-based, even if the user has manually configured tunnels created by GRE to use IFLs.

   The anchor Packet Forwarding Engine value is the line card of the anchor Packet Forwarding Engine, for example, pfe-x/y/0. This information can be viewed from the `show interfaces terse` *pfe** command output.

   Sample Configuration:

```
[edit routing-options]
dynamic-tunnels {
    dynamic-tunnel-attributes attribute-1 {
        dynamic-tunnel-type <GRE | UDP | BGP-SIGNAL>;
        dynamic-tunnel-anchor-pfe pfe-1/0/0;
    }
}
```

2. Associating dynamic tunnel profile to prefix list.

   Configuring a policy with `dynamic-tunnel-attributes` as the action associates the dynamic tunnel to the prefix list. The policy `from` action allows the creation of tunnel with specified attributes for any matching condition, such as a prefix range, community, or source address of BGP routes, and so on.

   Sample configuration:

```
[edit policy-options]
policy-statement policy-name {
    term term {
```

```
        from {
            <route-filter | next-hop | community>>;
        }
        then {
            dynamic-tunnel-attributes <attribute-name>;
        }
    }
}
```

3. Including the tunnel policy under the forwarding table export policy.

   After the policy is configured, it is included in the forwarding table export policy for the parsing of the policy.

   Using the export-policy, the tunnel attributes get associated with the route. Whenever a route from BGP is queued for resolution, the forwarding table export policy is evaluated, and the tunnel attributes are obtained from the policy module based on the applied filters. The obtained tunnel attributes are then attached to the next hop in form of a tunnel composite next hop. The corresponding anchor forwarding structures, based on the Packet Forwarding Engine name and tunnel type, are created and sent to the forwarding table before a tunnel composite next hop is sent. However, if none of the attributes map to the tunnel composite next hop, then the forwarding structure is created on every Packet Forwarding Engine, similar to the non-localized dynamic tunnels.

   Sample configuration:

```
[edit routing-options]
forwarding-table {
    export  dynamic-tunnel;
}
```

Configuring Localization for Existing Next-Hop-Based Dynamic Tunnels

CAUTION: Making on the fly changes to dynamic tunnel attributes can result in an FPC crash due to high memory utilization. Hence, we recommend deactivating the dynamic-tunnels configuration before configuring localization.

To update tunnel attributes for existing next-hop-based dynamic tunnels, the following should be performed:

1. Deactivate `dynamic-tunnels` configuration under the `[edit routing-options]` hierarchy level.

Sample configuration:

```
[edit routing-options]
user@host# deactivate dynamic-tunnels
user@host# commit
```

2. Change tunnel attributes as required.

3. Activate `dynamic-tunnels` configuration under the `[edit routing-options]` hierarchy level.

   Sample configuration:

```
[edit routing-options]
user@host# activate dynamic-tunnels
user@host# commit
```

To configure localization for existing non-local next-hop-based dynamic tunnels:

> ⚠ **CAUTION**: Making on the fly changes to configure localization for existing non-local next-hop-based dynamic tunnels can result in an FPC crash due to high memory utilization. Hence, we recommend deactivating the dynamic-tunnels configuration before configuring localization.

1. Deactivate the `dynamic-tunnels` configuration at the `[edit routing-options]` hierarchy level.

2. Create tunnel-attributes profile and add policy for localizing the dynamic tunnels, similar to new next-hop-based dynamic tunnels.

3. Activate the `dynamic-tunnels` configuration.

Troubleshooting Localized Next-Hop-Based Dynamic Tunnels

With localization of next-hop-based dynamic tunnels, the tunnel composite next hops are associated with anchor Packet Forwarding Engine IDs. The following traceroute configuration statements at the `[edit routing-options]` hierarchy level help in troubleshooting the localized dynamic tunnels:

- `dynamic-tunnels traceoptions flag all`—Tracking creation and deletion of tunnel in DTM.
- `resolution traceoptions flag tunnel`—Tracking resolver operations on BGP route.
- `forwarding-table traceoptions flag all`—Tracking tunnels sent to the kernel.
- `traceoptions flag all`—Tracking of route learning process.

The following commands can be used to check if a route is using a localized next-hop-based dynamic tunnel:

1. **show route *prefix* extensive**—To obtain the indirect next hop.

   For example:

```
user@host> show route 1.2.3.4 extensive
MPLS-over-UDP-PE1.inet.0: 24 destinations, 26 routes (24 active, 0 holddown, 0 hidden)
1.2.3.4/32 (1 entry, 1 announced)
TSI:
KRT in-kernel 1.2.3.4/32 -> {indirect(1048577)}
Page 0 idx 1, (group pe1-ce1 type External) Type 1 val 0xb209a78 (adv_entry)
   Advertised metrics:
     Nexthop: Self
     AS path: [100] I
     Communities: target:600:1 encapsulation:mpls-in-udp(0xd)
```

2. **show krt indirect-next-hop index *indirect-next-hop* detail**—To check for anchor Packet Forwarding Engine field in the detailed output of the indirect next hop.

   For example:

```
user@host> show krt indirect-next-hop index 1048577 detail
Indirect Nexthop detail:
Index: 1048577 Protocol next-hop address: 1.1.1.6
  RIB Table: bgp.l3vpn.0               Label: Push 299808
  Policy Version: 2                    References: 11
  Locks: 3                             0xb227980
  Flags: 0x0
  INH Session ID: 0x0
  Ref RIB Table: unknown
  Export policy detail:
     (Dynamic tunnel hash : 309985522)
        Tunnel type: UDP, Reference count: 4, nhid: 1016
        Destination address: 1.1.1.6, Source address: 1.1.1.2
        Anchored-PFE: pfe-1/0/0
        VPN Label: Push 299808, TTL action: prop-ttl
      IGP FRR Interesting proto count : 11
      Chain IGP FRR Node Num          : 1
         IGP Resolver node(hex)       : 0xc838b94
```

```
          IGP Route handle(hex)        : 0xb1d7674      IGP rt_entry protocol        : Tunnel
          IGP Actual Route handle(hex) : 0x0            IGP Actual rt_entry protocol : Any
```

Unsupported Features for Next-Hop-Based Dynamic Tunnels Localization

Junos OS does not support the following functionality with localization for next-hop-based dynamic tunnels:

- Chained composite next hops at the `[edit routing-options forwarding-table chained-composite-next-hop ingress l3vpn]` hierarchy level.
- Anchor Packet Forwarding Engine resiliency.

  There is no resiliency support for next-hop-based dynamic tunnels with localization. After localization of the next-hop-based dynamic tunnels, the anchor Packer Forwarding Engine becomes the single entity for processing any given tunnel on the device. Although anchor Packer Forwarding Engine resiliency is not supported, for gateway devices, redundancy at the gateway device ensures that when the Packer Forwarding Engine to which the tunnel composite next hop is delegated goes down, the traffic must be rerouted to the redundant gateway device. The routing protocol process monitors the state of the Packer Forwarding Engine, and withdraws BGP advertisement of all the routes pointing to the tunnel composite next hops anchored on that Packer Forwarding Engine.

  Only the anchored Packet Forwarding Engine has the full-fledged tunnel composite next hop and all the other Packet Forwarding Engines have only steering entries to forward traffic to the anchor Packet Forwarding Engine. These steering entries are not withdrawn, when an anchor FPC goes down.
- Localization of next-hop-based dynamic tunnels is not supported on logical systems.
- IPv6 is not supported with localization of next-hop-based dynamic tunnels.
- With localization, the `show dynamic-tunnels database summary` command does not display accurate tunnels summary when the state of the anchor Packet Forwarding Engine line card is not up. As a workaround, use the `show dynamic-tunnels database` and `show dynamic-tunnels database terse` command output.

Overview of Next-Hop-Based Dynamic Tunneling Using IP-Over-IP Encapsulation

SUMMARY

IN THIS SECTION

Benefits | 156

What is IP-over-IP Dynamic Next Hop-based Tunneling? | 156

IP-over-IP Tunnel Stitching | 157

Benefits

IP-over-IP tunneling provides the following benefits:

- **Alternative to MPLS over UDP**—Can be used as an alternative to MPLS-over-UDP tunneling to provide IP service wherein there is a dedicated device per service.
- **Ability to steer specific traffic**—Enables smooth migration when MPLS and IP networks co-exist because routes can be filtered to steer specific traffic over IP tunnels as opposed to MPLS tunnels.
- **Ability to support tunnels at increasing scale**—Dynamic tunnel creation using BGP control plane can facilitate tunnel creation at scale.

What is IP-over-IP Dynamic Next Hop-based Tunneling?

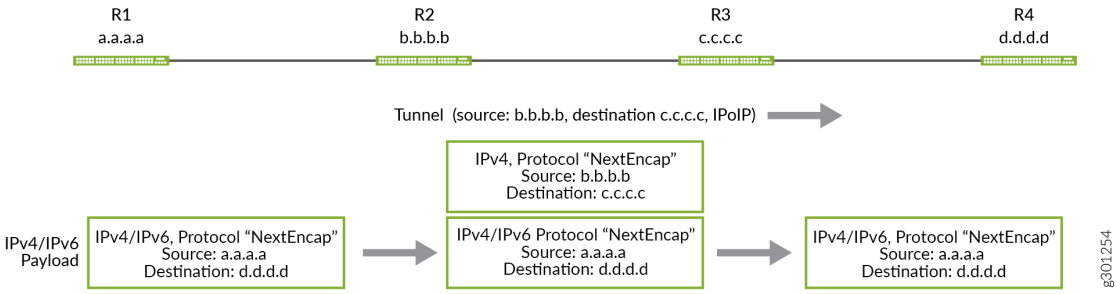
An IP network contains edge devices and core devices. To achieve higher scale and reliability among these devices, you need to logically isolate the core network from the external network that the edge devices interact with, by using an overlay encapsulation.

Starting in Junos OS Release 20.3R1, we support an IP-over-IP encapsulation to facilitate IP overlay construction over IP transport network. IP over IP relies on a next hop-based infrastructure to support a higher scale. The feature supports IPv4 encapsulation of IPv6 and IPv4 payload. Among the other overlay encapsulations supported, IP-over-IP encapsulation is the only kind that allows:

- transit devices to parse the inner payload and use inner packet fields for hash computation
- customer edge devices to route traffic into and out of the tunnel without any throughput reduction

On MX Series routers, routing protocol daemon (RPD) sends the encapsulation header with tunnel composite nexthop and the Packet Forwarding Engine (PFE) finds the tunnel destination address and forwards the packet. On PTX Series routers and QFX10000 switches, RPD sends fully resolved next hop-based tunnel to the Packet Forwarding Engine. BGP protocol is used to distribute routes and signal dynamic tunnels.

The following illustration depicts how IPv4 or IPv6 traffic are sent from R-1 to R-5 through an IP over IP tunnel established between R-2 and R-4:



IP-over-IP Tunnel Stitching

In Junos OS Release 21.3R1, we introduce IP-over-IP tunnel stitching on MX240, MX480, MX960, PTX1000, PTX10008, PTX10016, and QFX10002. You can use this feature to terminate an IP-over-IP tunnel on a device and initiate another tunnel on the same device. When a device receives the IP-over-IP packet, it de-encapsulates the outer packet header and inner packet lookup occurs. The inner IP packet header then points to another tunnel on the same device, where the same device encapsulates the packet again with another IP-over-IP header.

Example: Configuring Next-Hop-Based IP-Over-IP Dynamic Tunnels

SUMMARY

Learn how to configure next hop-based tunnels by using IP-over-IP encapsulation.

IN THIS SECTION

- Requirements | **157**
- Overview | **158**
- Configuring IP-over-IP Dynamic Tunnels with a Protocol Next Hop | **159**
- Example: Configure an IPoIP Tunnel in an MPLS Environment with LDP tunnel, Resolved Through inetcolor.0 Using Static Configuration | **172**
- Example: Configure an IPoIP Tunnel with LDP tunnel in an MPLS Cloud, Resolved through inetcolor.0 Using BGP Signaling | **188**
- Verification | **199**

Requirements

This example uses the following hardware and software components:

- 5 MX Series routers.
- Junos OS Release 20.3R1 or later version.
  - See the Feature Explorer for supported platforms.

Overview

IN THIS SECTION

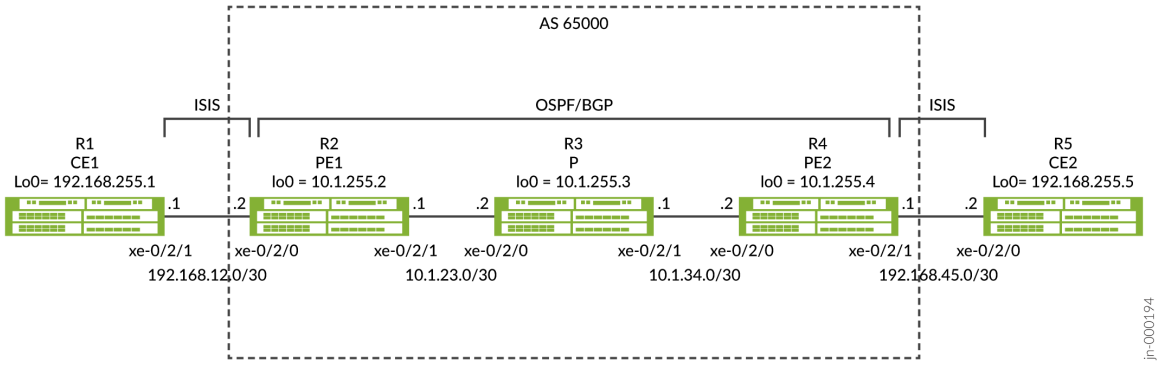
Topology | 158

Starting in Junos OS Release 20.3R1, we support an IP-over-IP encapsulation to facilitate IP overlay construction over IP transport network. This example shows the establishment of unicast IP-over-IP tunnels between devices with a protocol next hop (PNH) through an IBGP peering between R2 and R4, which are connected over an OSPF core, to exchange routes and signal dynamic tunnels.

Topology

Figure 1 illustrates an IP-over-IP scenario with 5 devices.

In this example, we are exchanging routes from R1 to R5 and vice versa through IP-over-IP dynamic tunnels established between the R2 and R4. Routes from R1 are exported to R2 and routes from R5 are exported to R4, by using the protocol IS-IS. We configure a unicast IPIP tunnel *Tunnel-01* from R2 to R4 and another tunnel *Tunnel-01* from R4 to R2. Route prefixes that are generated within the network masks from the configured destination-networks of the peer device are used for creating the tunnel and traffic flows in the opposite direction of the routes in the tunnel..



Configuring IP-over-IP Dynamic Tunnels with a Protocol Next Hop

IN THIS SECTION

- Verification | 167

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter commit from configuration mode.

R1

```
set interfaces xe-0/2/0 unit 0 description R1-to-R2
set interfaces xe-0/2/0 unit 0 family inet address 192.168.12.1/30
set interfaces xe-0/2/0 unit 0 family iso
set interfaces lo0 unit 0 family inet address 192.168.255.1/32
set interfaces lo0 unit 0 family iso address 49.0001.1920.1682.5501.00
set routing-options router-id 192.168.255.1
set protocols isis interface xe-0/2/0.0
set protocols isis interface lo0.0
set protocols isis level 1 disable
```

R2

```
set interfaces xe-0/2/0 description R2-to-R1
set interfaces xe-0/2/0 unit 0 family inet address 192.168.12.2/30
set interfaces xe-0/2/0 unit 0 family iso
set interfaces xe-0/2/1 description R2-to-R3
set interfaces xe-0/2/1 unit 0 family inet address 10.1.23.1/30
set interfaces lo0 unit 0 family inet address 10.1.255.2/32
set interfaces lo0 unit 0 family iso address 49.0001.0010.1255.0002.00
set policy-options policy-statement export-bgp term t1 from protocol bgp
set policy-options policy-statement export-bgp term t1 then accept
set policy-options policy-statement export-isis term t1 from protocol isis
set policy-options policy-statement export-isis term t1 then next-hop self
set policy-options policy-statement export-isis term t1 then accept
```

```
set routing-options resolution rib inet.0 resolution-ribs inet.3
set routing-options router-id 10.1.255.2
set routing-options autonomous-system 65000
set routing-options dynamic-tunnels Tunnel-01 source-address 10.1.255.2
set routing-options dynamic-tunnels Tunnel-01 ipip
set routing-options dynamic-tunnels Tunnel-01 destination-networks 10.1.255.0/24
set protocols bgp group iBGP type internal
set protocols bgp group iBGP local-address 10.1.255.2
set protocols bgp group iBGP family inet unicast
set protocols bgp group iBGP export export-isis
set protocols bgp group iBGP neighbor 10.1.255.4
set protocols isis interface xe-0/2/0.0
set protocols isis interface lo0.0
set protocols isis level 1 disable
set protocols isis export export-bgp
set protocols ospf area 0.0.0.0 interface xe-0/2/1.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

R3

```
set interfaces xe-0/2/0 unit 0 description R3-to-R2
set interfaces xe-0/2/0 unit 0 family inet address 10.1.23.2/30
set interfaces xe-0/2/1 unit 0 description R3-to-R4
set interfaces xe-0/2/1 unit 0 family inet address 10.1.34.1/30
set interfaces lo0 unit 0 family inet address 10.1.255.3/32
set routing-options router-id 10.1.255.3
set protocols ospf area 0.0.0.0 interface xe-0/2/0.0
set protocols ospf area 0.0.0.0 interface xe-0/2/1.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

R4

```
set interfaces xe-0/2/0 unit 0 description R4-to-R3
set interfaces xe-0/2/0 unit 0 family inet address 10.1.34.2/30
set interfaces xe-0/2/1 unit 0 description R4-to-R5
set interfaces xe-0/2/1 unit 0 family inet address 192.168.45.1/30
set interfaces xe-0/2/1 unit 0 family iso
set interfaces lo0 unit 0 family inet address 10.1.255.4/32
set interfaces lo0 unit 0 family iso address 49.0001.0010.1255.0004.00
set policy-options policy-statement export-bgp term t1 from protocol bgp
set policy-options policy-statement export-bgp term t1 then accept
```

```
set policy-options policy-statement export-isis term t1 from protocol isis
set policy-options policy-statement export-isis term t1 then next-hop self
set policy-options policy-statement export-isis term t1 then accept
set routing-options resolution rib inet.0 resolution-ribs inet.3
set routing-options router-id 10.1.255.4
set routing-options autonomous-system 65000
set routing-options dynamic-tunnels Tunnel-01 source-address 10.1.255.4
set routing-options dynamic-tunnels Tunnel-01 ipip
set routing-options dynamic-tunnels Tunnel-01 destination-networks 10.1.255.0/24
set protocols bgp group iBGP type internal
set protocols bgp group iBGP local-address 10.1.255.4
set protocols bgp group iBGP family inet unicast
set protocols bgp group iBGP export export-isis
set protocols bgp group iBGP neighbor 10.1.255.2
set protocols isis interface xe-0/2/1.0
set protocols isis interface lo0.0
set protocols isis level 1 disable
set protocols isis export export-bgp
set protocols ospf area 0.0.0.0 interface xe-0/2/0.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

R5

```
set interfaces xe-0/2/0 unit 0 description R5-to-R4
set interfaces xe-0/2/0 unit 0 family inet address 192.168.45.2/30
set interfaces xe-0/2/0 unit 0 family iso
set interfaces lo0 unit 0 family inet address 192.168.255.5/32
set interfaces lo0 unit 0 family iso address 49.0001.1920.1682.5505.00
set routing-options router-id 192.168.255.5
set protocols isis interface xe-0/2/0.0
set protocols isis interface lo0.0
set protocols isis level 1 disable
```

Configuring IP-IP dynamic tunnels with a Protocol Next Hop

Step-by-Step Procedure for R1

R1 and R5 have a similar configuration so we will only show the step-by-step procedure for R1.

1. Enter configuration mode on R1.

2. Configure the interface connected to R2 and interface lo0. Make sure to configure both family `inet` and `iso`. Family `iso` is needed for protocol IS-IS.

```
[edit]
user@R1# set interfaces xe-0/2/0 unit 0 description R1-to-R2
user@R1# set interfaces xe-0/2/0 unit 0 family inet address 192.168.12.1/30
user@R1# set interfaces xe-0/2/0 unit 0 family iso
user@R1# set interfaces lo0 unit 0 family inet address 192.168.255.1/32
user@R1# set interfaces lo0 unit 0 family iso address 49.0001.1920.1682.5501.00
```

3. Configure the Router ID.

```
[edit]
user@R1# set routing-options router-id 192.168.255.1
```

4. Configure protocols IS-IS. Routes are advertised between R1 and R2 using the IS-IS protocol.

```
[edit]
user@R1# set protocols isis interface xe-0/2/0.0
user@R1# set protocols isis interface lo0.0
user@R1# set protocols isis level 1 disable
```

5. Enter `commit` on R1 from the configuration mode.

Step-by-Step Procedure for R2

R2 and R4 have a similar configuration so we will only show the step-by-step procedure for R2.

1. Enter configuration mode on R2.

2. Configure the interfaces connected to R1 and R3 and interface lo0. Ensure to configure both family`inet` and `iso` on the interface connected to R1 and lo0.

```
[edit]
user@R2# set interfaces xe-0/2/0 description R2-to-R1
user@R2# set interfaces xe-0/2/0 unit 0 family inet address 192.168.12.2/30
user@R2# set interfaces xe-0/2/0 unit 0 family iso
user@R2# set interfaces xe-0/2/1 description R2-to-R3
user@R2# set interfaces xe-0/2/1 unit 0 family inet address 10.1.23.1/30
```

```
user@R2# set interfaces lo0 unit 0 family inet address 10.1.255.2/32
user@R2# set interfaces lo0 unit 0 family iso address 49.0001.0010.1255.0002.00
```

3. Configure protocols IS-IS for the interface connected to R1. The export policy to advertise BGP routes into IS-IS is shown in the policy configuration step.

```
[edit]
user@R2# set protocols isis interface xe-0/2/0.0
user@R2# set protocols isis interface lo0.0
user@R2# set protocols isis level 1 disable
user@R2# set protocols isis export export-bgp
```

4. Configure the OSPF protocol for the interface connected to R3 for lo0 reachability.

```
[edit]
user@R2# set protocols ospf area 0.0.0.0 interface xe-0/2/1.0
user@R2# set protocols ospf area 0.0.0.0 interface lo0.0
```

5. Configure the `router-id` and `autonomous-system`, and IBGP between R2 and R4. The export policy to advertise IS-IS routes into BGP is shown in the policy configuration step.

```
[edit]
user@R2# set routing-options router-id 10.1.255.2
user@R2# set routing-options autonomous-system 65000
user@R2# set protocols bgp group iBGP type internal
user@R2# set protocols bgp group iBGP local-address 10.1.255.2
user@R2# set protocols bgp group iBGP family inet unicast
user@R2# set protocols bgp group iBGP export export-isis
user@R2# set protocols bgp group iBGP neighbor 10.1.255.4
```

6. Configure the BGP and IS-IS export policies that were applied during the previous steps. The `export-bgp` policy is applied to protocols IS-IS as an export to advertise BGP routes into IS-IS and the `export-isis` policy is applied to BGP as an export to advertise IS-IS routes into BGP. The next-hop self option allows R2 to advertise the IS-IS routes into BGP with R2 as the next-hop instead of the interface next-hop of R1.

```
[edit]
user@R2# set policy-options policy-statement export-bgp term t1 from protocol bgp
user@R2# set policy-options policy-statement export-bgp term t1 then accept
```

```
user@R2# set policy-options policy-statement export-isis term t1 from protocol isis
user@R2# set policy-options policy-statement export-isis term t1 then next-hop self
user@R2# set policy-options policy-statement export-isis term t1 then accept
```

7. Configure the IP-IP dynamic tunnel *Tunnel-01* from R2 to R4. The configuration option `resolution-ribs inet.3` allows for route resolution to take place in inet.3 and is needed to establish the tunnel.

```
[edit]
user@R2# set routing-options resolution rib inet.0 resolution-ribs inet.3
user@R2# set routing-options dynamic-tunnels Tunnel-01 source-address 10.1.255.2
user@R2# set routing-options dynamic-tunnels Tunnel-01 ipip
user@R2# set routing-options dynamic-tunnels Tunnel-01 destination-networks 10.1.255.0/24
```

8. (Optional) - Alternative configuration for the IP-IP dynamic tunnel *Tunnel-01* from R2 to R4. Instead of configuring the `resolution-ribs inet.3` you can configure the tunnel preference lower than the protocol next-hop preference for the route to the tunnel endpoint. The route for R4 is learned using OSPF and has a preference of 10 and the default preference of the tunnel is 305. Configure the tunnel preference lower than the OSPF preference allows the tunnel to be preferred and to establish.

```
[edit]
user@R2# set routing-options dynamic-tunnels Tunnel-01 source-address 10.1.255.2
user@R2# set routing-options dynamic-tunnels Tunnel-01 ipip
user@R2# set routing-options dynamic-tunnels Tunnel-01 destination-networks 10.1.255.0/24
preference 9
```

9. Enter `commit` from the configuration mode on R2.

Step-by-Step Procedure for R3

1. Enter configuration mode on R3.
2. Configure the interfaces connected to R2 and R4 and interface lo0.

```
[edit]
user@R3# set interfaces xe-0/2/0 unit 0 description R3-to-R2
user@R3# set interfaces xe-0/2/0 unit 0 family inet address 10.1.23.2/30
user@R3# set interfaces xe-0/2/1 unit 0 description R3-to-R4
user@R3# set interfaces xe-0/2/1 unit 0 family inet address 10.1.34.1/30
user@R3# set interfaces lo0 unit 0 family inet address 10.1.255.3/32
```

3. Configure the Router ID.

```
[edit]
user@R3# set routing-options router-id 10.1.255.3
```

4. Configure the OSPF protocol for the interfaces connected to R2 and R4 for lo0 reachability.

```
[edit]
user@R3# set protocols ospf area 0.0.0.0 interface xe-0/2/0.0
user@R3# set protocols ospf area 0.0.0.0 interface xe-0/2/1.0
user@R3# set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

5. Enter `commit` from configuration mode on R3 device.

Results

Verify your configuration by checking the below configurations from devices as follows:

Here's how you can verify configurations on R2:

```
user@R2# show interfaces
```

```
xe-0/2/0 {
    description R2-to-R1;
    unit 0 {
        family inet {
            address 192.168.12.2/30;
        }
        family iso;
    }
}
xe-0/2/1 {
    description R2-to-R3;
    unit 0 {
        family inet {
            address 10.1.23.1/30;
        }
    }
}
lo0 {
    unit 0 {
```

```
        family inet {
            address 10.1.255.2/32;
        }
        family iso {
            address 49.0001.0010.1255.0002.00;
        }
    }
}
```

user@R2# show routing-options

```
resolution {
    rib inet.0 {
        resolution-ribs inet.3;
    }
}
router-id 10.1.255.2;
autonomous-system 65000;
dynamic-tunnels {
    Tunnel-01 {
        source-address 10.1.255.2;
        ipip;
        destination-networks {
            10.1.255.0/24;
        }
    }
}
```

user@R2# show protocols

```
bgp {
    group iBGP {
        type internal;
        local-address 10.1.255.2;
        family inet {
            unicast;
        }
        export export-isis;
        neighbor 10.1.255.4;
    }
}
```

```
isis {
    interface xe-0/2/0.0;
    interface lo0.0;
    level 1 disable;
    export export-bgp;
}
ospf {
    area 0.0.0.0 {
        interface xe-0/2/1.0;
        interface lo0.0;
    }
}
```

user@R2# show policy-options

```
policy-statement export-bgp {
    term t1 {
        from protocol bgp;
        then accept;
    }
}
policy-statement export-isis {
    term t1 {
        from protocol isis;
        then {
            next-hop self;
            accept;
        }
    }
}
```

Verification

IN THIS SECTION

- Verify Dynamic Tunnel Database | **168**
- Verify Route Table in inet.3 | **169**
- Verify BGP Routes Using the Tunnel | **170**
- Verify End-to-End Reachability | **171**

Verify Dynamic Tunnel Database

Purpose

To verify the dynamic tunnel database information, use the `show dynamic-tunnels database` operational mode command.

Action

```
user@R2> show dynamic-tunnels database
*- Signal Tunnels #- PFE-down
Table: inet.3

Destination-network: 10.1.255.0/24

Tunnel to: 10.1.255.4/32
  Reference count: 3
  Next-hop type: IPoIP (forwarding-nexthop)
    Source address: 10.1.255.2
    Next hop: tunnel-composite, 0x76b6c50, nhid 515
      Reference count: 2
      Ingress Route: [OSPF] 10.1.255.4/32, via metric 2
      Traffic Statistics: Packets 0, Bytes 0
      State: Up
  Aggregate Traffic Statistics:
    Tunnel Encapsulation: Dest 10.1.255.4, Src 10.1.255.2, IPoIP, Tunnel-Id 1
    Traffic Statistics: Packets 0, Bytes 0
```

```
user@R4> show dynamic-tunnels database
*- Signal Tunnels #- PFE-down
Table: inet.3

Destination-network: 10.1.255.0/24

Tunnel to: 10.1.255.2/32
  Reference count: 3
  Next-hop type: IPoIP (forwarding-nexthop)
    Source address: 10.1.255.4
    Next hop: tunnel-composite, 0x76b6c50, nhid 513
      Reference count: 2
```

```
      Ingress Route: [OSPF] 10.1.255.2/32, via metric 2
      Traffic Statistics: Packets 0, Bytes 0
      State: Up
  Aggregate Traffic Statistics:
    Tunnel Encapsulation: Dest 10.1.255.2, Src 10.1.255.4, IPoIP, Tunnel-Id 1
    Traffic Statistics: Packets 0, Bytes 0
```

Meaning

The output indicates that an IPoIP tunnel is established between R2 (`192.168.0.21` source) and R4 (`192.168.0.41` destination) and another IPoIP tunnel is established between R4 (`192.168.0.41` source) and R2 (`192.168.0.21` destination).

Verify Route Table in inet.3

Purpose

To verify routes generated on inet.3 table, use the `show route table inet.3` operational mode command.

Action

```
user@R2> show route table inet.3

inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.255.0/24      *[Tunnel/305] 02:02:44
                       Tunnel
10.1.255.4/32      *[Tunnel/305] 02:02:44, metric 2
                       Tunnel Composite, IPoIP (src 10.1.255.2 dest 10.1.255.4)
```

```
user@R4> show route table inet.3

inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.255.0/24      *[Tunnel/305] 6d 01:35:50
                       Tunnel
```

```
10.1.255.2/32      *[Tunnel/305] 6d 01:35:48, metric 2
                       Tunnel Composite, IPoIP (src 10.1.255.4 dest 10.1.255.2)
```

Meaning

The output indicates the route used for resolving the BGP traffic that will use the tunnel.

Verify BGP Routes Using the Tunnel

Purpose

To verify BGP routes received on R2 and R4 for R1 and R5 are using the tunnel.

Action

```
user@R2> show route protocol bgp

inet.0: 17 destinations, 17 routes (17 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.255.5/32   *[BGP/170] 02:42:48, MED 10, localpref 100, from 10.1.255.4
                      AS path: I, validation-state: unverified
                    >  via Tunnel Composite, IPoIP (src 10.1.255.2 dest 10.1.255.4)

inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
```

```
user@R4> show route protocol bgp

inet.0: 17 destinations, 17 routes (17 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.255.1/32   *[BGP/170] 00:13:30, MED 10, localpref 100, from 10.1.255.2
                      AS path: I, validation-state: unverified
                    >  via Tunnel Composite, IPoIP (src 10.1.255.4 dest 10.1.255.2)
```

```
inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
```

Meaning

The output indicates that R2 is using the tunnel for the BGP route to R5, and R4 is using the tunnel for the BGP route to R1.

Verify End-to-End Reachability

Purpose

Verify R1 can ping R5 by using the `ping 192.168.255.5 source 192.168.255.1 count 2` operational mode command.

Action

```
user@R1>ping 192.168.255.5 source 192.168.255.1 count 2
PING 192.168.255.5 (192.168.255.5): 56 data bytes
64 bytes from 192.168.255.5: icmp_seq=0 ttl=62 time=5.565 ms
64 bytes from 192.168.255.5: icmp_seq=1 ttl=62 time=5.957 ms

--- 192.168.255.5 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 5.565/5.761/5.957/0.196 ms
```

Meaning

The output from R1 shows that R1 can ping R5.

Example: Configure an IPoIP Tunnel in an MPLS Environment with LDP tunnel, Resolved Through inetcolor.0 Using Static Configuration

IN THIS SECTION

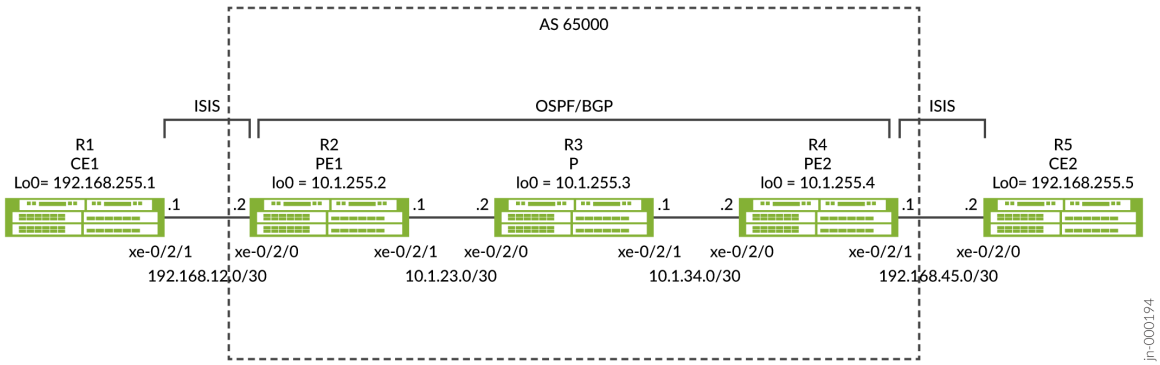
- Verification | **183**

By default, MPLS has a higher preference than IP. For example, with MPLS and LDP configured among R2, R3, and R4, wherein R2 is reachable with R4 through LDP, routes from R2 will get resolved over LDP, instead of IP-over-IP, because of the higher preference.

If you prefer to have a particular route to resolve over the IP-over-IP instead of LDP, you can do so by creating an inetcolor table, where IP-over-IP has a higher preference and setting BGP to resolve that route over inetcolor table instead of inet3 table. The following example shows you how to do so by using static configuration.

Topology

In this example, we are exchanging routes from R1 to R5 and vice versa through IP-over-IP dynamic tunnels established between the R2 and R4. Routes from R1 are exported to R2 and routes from R5 are exported to R4, by using the protocol IS-IS. We configure a unicast IPIP tunnel *Tunnel-01* from R2 to R4 and another tunnel *Tunnel-01* from R4 to R2. Route prefixes that are generated within the network masks from the configured destination-networks of the peer device are used for creating the tunnel and traffic flows in the opposite direction of the routes in the tunnel.



CLI Quick Configuration

R1

```
set interfaces xe-0/2/0 unit 0 description R1-to-R2
set interfaces xe-0/2/0 unit 0 family inet address 192.168.12.1/30
set interfaces xe-0/2/0 unit 0 family iso
set interfaces lo0 unit 0 family inet address 192.168.255.1/32
set interfaces lo0 unit 0 family iso address 49.0001.1920.1682.5501.00
set routing-options router-id 192.168.255.1
set protocols isis interface xe-0/2/0.0
set protocols isis interface lo0.0
set protocols isis level 1 disable
```

R2

```
set interfaces xe-0/2/0 description R2-to-R1
set interfaces xe-0/2/0 unit 0 family inet address 192.168.12.2/30
set interfaces xe-0/2/0 unit 0 family iso
set interfaces xe-0/2/1 description R2-to-R3
set interfaces xe-0/2/1 unit 0 family inet address 10.1.23.1/30
set interfaces xe-0/2/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.1.255.2/32
set interfaces lo0 unit 0 family iso address 49.0001.0010.1255.0002.00
set policy-options policy-statement export-bgp term t1 from protocol bgp
set policy-options policy-statement export-bgp term t1 then accept
set policy-options policy-statement export-isis term t1 from protocol isis
set policy-options policy-statement export-isis term t1 then next-hop self
set policy-options policy-statement export-isis term t1 then accept
set policy-options policy-statement ipip-tunnel-color term term-01 from route-filter
192.168.255.5/32 exact
set policy-options policy-statement ipip-tunnel-color term term-01 then community add red
set policy-options policy-statement ipip-tunnel-color term term-01 then accept
set policy-options policy-statement set-dynamic-tunnel-ep term t1 from route-filter
10.1.255.4/32 exact
set policy-options policy-statement set-dynamic-tunnel-ep term t1 then tunnel-end-point-address
10.1.255.4
set policy-options policy-statement set-dynamic-tunnel-ep term t1 then accept
set policy-options community red members color:0:100
set routing-options router-id 10.1.255.2
set routing-options autonomous-system 65000
```

```
set routing-options dynamic-tunnels Tunnel-01 source-address 10.1.255.2
set routing-options dynamic-tunnels Tunnel-01 ipip
set routing-options dynamic-tunnels Tunnel-01 destination-networks 10.1.255.0/24 dyn-tunnel-
attribute-policy set-dynamic-tunnel-ep
set routing-options dynamic-tunnels Tunnel-01 destination-networks 10.1.255.0/24 colors 100
set protocols bgp group iBGP type internal
set protocols bgp group iBGP local-address 10.1.255.2
set protocols bgp group iBGP import ipip-tunnel-color
set protocols bgp group iBGP family inet unicast extended-nexthop-color
set protocols bgp group iBGP export export-isis
set protocols bgp group iBGP neighbor 10.1.255.4
set protocols isis interface xe-0/2/0.0
set protocols isis interface lo0.0
set protocols isis level 1 disable
set protocols isis export export-bgp
set protocols ldp interface xe-0/2/1.0
set protocols mpls interface xe-0/2/1.0
set protocols ospf area 0.0.0.0 interface xe-0/2/1.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

R3

```
set interfaces xe-0/2/0 unit 0 description R3-to-R2
set interfaces xe-0/2/0 unit 0 family inet address 10.1.23.2/30
set interfaces xe-0/2/0 unit 0 family mpls
set interfaces xe-0/2/1 unit 0 description R3-to-R4
set interfaces xe-0/2/1 unit 0 family inet address 10.1.34.1/30
set interfaces xe-0/2/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.1.255.3/32
set routing-options router-id 10.1.255.3
set protocols ldp interface xe-0/2/0.0
set protocols ldp interface xe-0/2/1.0
set protocols mpls interface xe-0/2/0.0
set protocols mpls interface xe-0/2/1.0
set protocols ospf area 0.0.0.0 interface xe-0/2/0.0
set protocols ospf area 0.0.0.0 interface xe-0/2/1.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

R4

```
set interfaces xe-0/2/0 unit 0 description R4-to-R3
set interfaces xe-0/2/0 unit 0 family inet address 10.1.34.2/30
set interfaces xe-0/2/0 unit 0 family mpls
set interfaces xe-0/2/1 unit 0 description R4-to-R5
set interfaces xe-0/2/1 unit 0 family inet address 192.168.45.1/30
set interfaces xe-0/2/1 unit 0 family iso
set interfaces lo0 unit 0 family inet address 10.1.255.4/32
set interfaces lo0 unit 0 family iso address 49.0001.0010.1255.0004.00
set policy-options policy-statement export-bgp term t1 from protocol bgp
set policy-options policy-statement export-bgp term t1 then accept
set policy-options policy-statement export-isis term t1 from protocol isis
set policy-options policy-statement export-isis term t1 then next-hop self
set policy-options policy-statement export-isis term t1 then accept
set policy-options policy-statement ipip-tunnel-color term term-01 from route-filter
192.168.255.1/32 exact
set policy-options policy-statement ipip-tunnel-color term term-01 then community add red
set policy-options policy-statement ipip-tunnel-color term term-01 then accept
set policy-options policy-statement set-dynamic-tunnel-ep term t1 from route-filter
10.1.255.2/32 exact
set policy-options policy-statement set-dynamic-tunnel-ep term t1 then tunnel-end-point-address
10.1.255.2
set policy-options policy-statement set-dynamic-tunnel-ep term t1 then accept
set policy-options community red members color:0:100
set routing-options router-id 10.1.255.4
set routing-options autonomous-system 65000
set routing-options dynamic-tunnels Tunnel-01 source-address 10.1.255.4
set routing-options dynamic-tunnels Tunnel-01 ipip
set routing-options dynamic-tunnels Tunnel-01 destination-networks 10.1.255.0/24 dyn-tunnel-
attribute-policy set-dynamic-tunnel-ep
set routing-options dynamic-tunnels Tunnel-01 destination-networks 10.1.255.0/24 colors 100
set protocols bgp group iBGP type internal
set protocols bgp group iBGP local-address 10.1.255.4
set protocols bgp group iBGP import ipip-tunnel-color
set protocols bgp group iBGP family inet unicast extended-nexthop-color
set protocols bgp group iBGP export export-isis
set protocols bgp group iBGP neighbor 10.1.255.2
set protocols isis interface xe-0/2/1.0
set protocols isis interface lo0.0
set protocols isis level 1 disable
set protocols isis export export-bgp
```

```
set protocols ldp interface xe-0/2/0.0
set protocols mpls interface xe-0/2/0.0
set protocols ospf area 0.0.0.0 interface xe-0/2/0.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

R5

```
set interfaces xe-0/2/0 unit 0 description R5-to-R4
set interfaces xe-0/2/0 unit 0 family inet address 192.168.45.2/30
set interfaces xe-0/2/0 unit 0 family iso
set interfaces lo0 unit 0 family inet address 192.168.255.5/32
set interfaces lo0 unit 0 family iso address 49.0001.1920.1682.5505.00
set routing-options router-id 192.168.255.5
set protocols isis interface xe-0/2/0.0
set protocols isis interface lo0.0
set protocols isis level 1 disable
```

Procedure

Step-by-Step Procedure for R1

R1 and R5 have a similar configuration so we will only show the step-by-step procedure for R1.

1. Enter configuration mode on R1.
2. Configure the interface connected to R2 and interface lo0. Make sure to configure both family `inet` and `iso`. Family `iso` is needed for protocol IS-IS.

```
[edit]
user@R1# set interfaces xe-0/2/0 unit 0 description R1-to-R2
user@R1# set interfaces xe-0/2/0 unit 0 family inet address 192.168.12.1/30
user@R1# set interfaces xe-0/2/0 unit 0 family iso
user@R1# set interfaces lo0 unit 0 family inet address 192.168.255.1/32
user@R1# set interfaces lo0 unit 0 family iso address 49.0001.1920.1682.5501.00
```

3. Configure the Router ID.

```
[edit]
user@R1# set routing-options router-id 192.168.255.1
```

4. Configure protocols IS-IS. Routes are advertised between R1 and R2 using the IS-IS protocol.

```
[edit]
user@R1# set protocols isis interface xe-0/2/0.0
user@R1# set protocols isis interface lo0.0
user@R1# set protocols isis level 1 disable
```

5. Enter `commit` on R1 from the configuration mode.

Step-by-Step Procedure for R2

R2 and R4 have a similar configuration so we will only show the step-by-step procedure for R2.

1. Enter configuration mode on R2.

2. Configure the interfaces connected to R1 and R3 and interface lo0. Ensure to configure both family`inet` and `iso` on the interface connected to R1 and lo0, and to configure both family`inet` and `mpls` on the interface connected to R3.

```
[edit]
user@R2# set interfaces xe-0/2/0 description R2-to-R1
user@R2# set interfaces xe-0/2/0 unit 0 family inet address 192.168.12.2/30
user@R2# set interfaces xe-0/2/0 unit 0 family iso
user@R2# set interfaces xe-0/2/1 description R2-to-R3
user@R2# set interfaces xe-0/2/1 unit 0 family inet address 10.1.23.1/30
user@R2# set interfaces xe-0/2/0 unit 0 family mpls
user@R2# set interfaces lo0 unit 0 family inet address 10.1.255.2/32
user@R2# set interfaces lo0 unit 0 family iso address 49.0001.0010.1255.0002.00
```

3. Configure protocols IS-IS for the interface connected to R1. The export policy to advertise BGP routes into IS-IS is shown in the policy configuration step.

```
[edit]
user@R2# set protocols isis interface xe-0/2/0.0
user@R2# set protocols isis interface lo0.0
user@R2# set protocols isis level 1 disable
user@R2# set protocols isis export export-bgp
```

4. Configure the OSPF protocol for the interface connected to R3 for lo0 reachability.

```
[edit]
user@R2# set protocols ospf area 0.0.0.0 interface xe-0/2/1.0
user@R2# set protocols ospf area 0.0.0.0 interface lo0.0
```

5. Configure the LDP and MPLS protocols for the interface connected to R3.

```
[edit]
user@R2# set protocols ldp interface xe-0/2/1.0
user@R2# set protocols mpls interface xe-0/2/1.0
```

6. Configure the `router-id` and `autonomous-system` under the `routing-options` hierarchy, and configure IBGP between R2 and R4. The import policy to add a community to the routes learned using BGP and the export policy to advertise IS-IS routes into BGP is shown in the policy configuration step. Ensure to include `extended-nexthop-color` option to the `family inet unicast` configuration to allow resolution using the inetcolor.0 table.

```
[edit]
user@R2# set routing-options router-id 10.1.255.2
user@R2# set routing-options autonomous-system 65000
user@R2# set protocols bgp group iBGP type internal
user@R2# set protocols bgp group iBGP local-address 10.1.255.2
user@R2# set protocols bgp group iBGP import ipip-tunnel-color
user@R2# set protocols bgp group iBGP family inet unicast extended-nexthop-color
user@R2# set protocols bgp group iBGP export export-isis
user@R2# set protocols bgp group iBGP neighbor 10.1.255.4
```

7. Configure the IP-IP dynamic tunnel *Tunnel-01* from R2 to R4. The `colors` configuration option allows the tunnel to be created in the inetcolor.0 route table. The `dyn-tunnel-attribute-policy` *set-dynamic-tunnel-ep* configures a static tunnel endpoint. The policy is shown with the policy configuration step.

```
[edit]
user@R2# set routing-options dynamic-tunnels Tunnel-01 source-address 10.1.255.2
user@R2# set routing-options dynamic-tunnels Tunnel-01 ipip
user@R2# set routing-options dynamic-tunnels Tunnel-01 destination-networks 10.1.255.0/24 dyn-
tunnel-attribute-policy set-dynamic-tunnel-ep
```

```
user@R2# set routing-options dynamic-tunnels Tunnel-01 destination-networks 10.1.255.0/24
colors 100
```

8. Configure the policies that were applied during the previous configuration steps. The *export-bgp* policy advertises BGP routes into IS-IS. The *export-isis* policy advertises IS-IS routes into BGP with changing the next-hop to R2. The *ipip-tunnel-color* policy applies the community to the route that is matched on in the `colors` configuration for the dynamic tunnel. The *set-dynamic-tunnel-ep* policy configures R4 as the tunnel endpoint.

```
[edit]
user@R2# set policy-options policy-statement export-bgp term t1 from protocol bgp
user@R2# set policy-options policy-statement export-bgp term t1 then accept
user@R2# set policy-options policy-statement export-isis term t1 from protocol isis
user@R2# set policy-options policy-statement export-isis term t1 then next-hop self
user@R2# set policy-options policy-statement export-isis term t1 then accept
user@R2# set policy-options policy-statement ipip-tunnel-color term term-01 from route-filter
192.168.255.5/32 exact
user@R2# set policy-options policy-statement ipip-tunnel-color term term-01 then community
add red
user@R2# set policy-options policy-statement ipip-tunnel-color term term-01 then accept
user@R2# set policy-options policy-statement set-dynamic-tunnel-ep term t1 from route-filter
10.1.255.4/32 exact
user@R2# set policy-options policy-statement set-dynamic-tunnel-ep term t1 then tunnel-end-
point-address 10.1.255.4
user@R2# set policy-options policy-statement set-dynamic-tunnel-ep term t1 then accept
user@R2# set policy-options community red members color:0:100
```

9. Enter `commit` from the configuration mode.

Step-by-Step Procedure for R3

1. Enter configuration mode on R3.
2. Configure the interfaces connected to R2 and R4 and interface lo0. Ensure to configure both family `inet` and `mpls` on the interfaces connected to R2 and R4.

```
[edit]
user@R3# set interfaces xe-0/2/0 unit 0 description R3-to-R2
user@R3# set interfaces xe-0/2/0 unit 0 family inet address 10.1.23.2/30
user@R3# set interfaces xe-0/2/0 unit 0 family mpls
user@R3# set interfaces xe-0/2/1 unit 0 description R3-to-R4
```

```
user@R3# set interfaces xe-0/2/1 unit 0 family inet address 10.1.34.1/30
user@R3# set interfaces xe-0/2/1 unit 0 family mpls
user@R3# set interfaces lo0 unit 0 family inet address 10.1.255.3/32
```

3. Configure the Router ID.

```
[edit]
user@R3# set routing-options router-id 10.1.255.3
```

4. Configure the OSPF protocol for the interfaces connected to R2 and R4 for lo0 reachability.

```
[edit]
user@R3# set protocols ospf area 0.0.0.0 interface xe-0/2/0.0
user@R3# set protocols ospf area 0.0.0.0 interface xe-0/2/1.0
user@R3# set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

5. Configure the LDP and MPLS protocols for the interfaces connected to R2 and R4.

```
[edit]
user@R2# set protocols ldp interface xe-0/2/0.0
user@R2# set protocols ldp interface xe-0/2/1.0
user@R2# set protocols mpls interface xe-0/2/0.0
user@R2# set protocols mpls interface xe-0/2/1.0
```

6. Enter `commit` from configuration mode on R3 device.

Results

Verify your configuration by checking the below configurations from the devices.

Here's how you can check the configurations on R2:

`user@R2# show interfaces`

```
xe-0/2/0 {
    description R2-to-R1;
    unit 0 {
        family inet {
            address 192.168.12.2/30;
```

```
        }
        family iso;
    }
}
xe-0/2/1 {
    description R2-to-R3;
    unit 0 {
        family inet {
            address 10.1.23.1/30;
        }
        family mpls;
    }
}
lo0 {
    apply-groups-except global;
    unit 0 {
        family inet {
            address 10.1.255.2/32;
        }
        family iso {
            address 49.0001.0010.1255.0002.00;
        }
    }
}
```

user@R2# show protocols

```
bgp {
    group iBGP {
        type internal;
        local-address 10.1.255.2;
        import ipip-tunnel-color;
        family inet {
            unicast {
                extended-nexthop-color;
            }
        }
        export export-isis;
        neighbor 10.1.255.4;
    }
}
isis {
```

```
    interface xe-0/2/0.0;
    interface lo0.0;
    level 1 disable;
    export export-bgp;
}
ldp {
    interface xe-0/2/1.0;
}
mpls {
    interface xe-0/2/1.0;
}
ospf {
    area 0.0.0.0 {
        interface xe-0/2/1.0;
        interface lo0.0;
    }
}
```

user@R2#show routing-options

```
router-id 10.1.255.2;
autonomous-system 65000;
dynamic-tunnels {
    Tunnel-01 {
        source-address 10.1.255.2;
        ipip;
        destination-networks {
            10.1.255.0/24 {
                dyn-tunnel-attribute-policy set-dynamic-tunnel-ep;
                colors 100;
            }
        }
    }
}
```

user@R2#show policy-options

```
policy-statement export-bgp {
    term t1 {
        from protocol bgp;
        then accept;
```

```
    }
}
policy-statement export-isis {
    term t1 {
        from protocol isis;
        then {
            next-hop self;
            accept;
        }
    }
}
policy-statement ipip-tunnel-color {
    term term-01 {
        from {
            route-filter 192.168.255.5/32 exact;
        }
        then {
            community add red;
            accept;
        }
    }
}
policy-statement set-dynamic-tunnel-ep {
    term t1 {
        from {
            route-filter 10.1.255.4/32 exact;
        }
        then {
            tunnel-end-point-address 10.1.255.4;
            accept;
        }
    }
}
community red members color:0:100;
```

Verification

IN THIS SECTION

- Verify Route Resolution | **184**
- Verify Dynamic Tunnel Database | **185**

Verify Route Next-Hops | 186
Verify End-to-End Reachability | 187

Verify Route Resolution

Purpose

To verify the route resolution of routes in both inet.3 and inetcolor.0 tables, use the `show route table inet.3` and `show route table inetcolor.0` operational mode commands.

Action

```
user@R2> show route table inet.3

inet.3: 3 destinations, 4 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.255.0/24      *[Tunnel/305] 00:12:21
                       Tunnel
10.1.255.3/32      *[LDP/9] 1d 19:37:01, metric 1
                    >  to 10.1.23.2 via xe-0/2/1.0
10.1.255.4/32      *[LDP/9] 1d 19:32:25, metric 1
                    >  to 10.1.23.2 via xe-0/2/1.0, Push 299792
                    [Tunnel/305] 00:13:38, metric 2
                       Tunnel Composite, IPoIP (src 10.1.255.2 dest 10.1.255.4)
```

```
user@R2> show route table inetcolor.0

inetcolor.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.255.0-0<c>/24
                   *[Tunnel/305] 00:13:26
                       Tunnel
10.1.255.4-100<c>/64
```

```
                    *[Tunnel/305] 00:13:26, metric 2
                        Tunnel Composite, IPoIP (src 10.1.255.2 dest 10.1.255.4-100<c>)
```

Meaning

The R2 output indicates that on inet.3 table, the route `10.1.255.4` is getting resolved by LDP because of higher preference than IP-over-IP. On the other hand, in the newly created inetcolor.0 table, the route `10.1.255.4` is getting resolved through IP-over-IP tunnel with `<c>` attached.

Verify Dynamic Tunnel Database

Purpose

To verify the IP-over-IP dynamic tunnel created by routes in the inetcolor.0 table, use the `show dynamic-tunnels database terse` operational mode command.

Action

```
user@R2>show dynamic-tunnels database terse
*- Signal Tunnels #- PFE-down
Table: inet.3

Destination-network: 10.1.255.0/24

*- Signal Tunnels #- PFE-down
Table: inetcolor.0

Destination-network: 10.1.255.0-0<c>/24
Destination                     Source          Next-hop                Type       Status
10.1.255.4-100<c>/64            10.1.255.2      0x76b71cc nhid 592      IPoIP      Up (via
metric 2, tunnel-endpoint 10.1.255.4)
```

Meaning

The R2 output indicates that the route `192.168.0.41` has created a next-hop-based dynamic tunnel.

Verify Route Next-Hops

Purpose

To verify all next-hops of the route that is set to resolve through IP-over-IP, use the `show route 192.168.255.5 extensive expanded-nh` operational mode command.

Action

```
user@R2>show route 192.168.255.5 extensive expanded-nh

inet.0: 18 destinations, 18 routes (18 active, 0 holddown, 0 hidden)
192.168.255.5/32 (1 entry, 1 announced)
Installed-nexthop:
Indr Composite (0x76b7238) 10.1.255.4-100<c>
  Krt_cnh (0x6fb1928) Index:593
    Krt_inh (0x7164d3c) Index:1048575 PNH: 10.1.255.4-100<c>
      Tun-comp (0x76b71cc) Index:592 IPoIP src 10.1.255.2 dest 10.1.255.4-100<c> tunnel-endpoint
10.1.255.4
TSI:
KRT in-kernel 192.168.255.5/32 -> {composite(593)}
IS-IS level 2, LSP fragment 0
        *BGP    Preference: 170/-101
                Next hop type: Indirect, Next hop index: 0
                Address: 0x76b7238
                Next-hop reference count: 2
                Source: 10.1.255.4
                Next hop type: Tunnel Composite, Next hop index: 592
                Next hop: via Tunnel Composite, IPoIP (src 10.1.255.2 dest 10.1.255.4-100<c>
tunnel-endpoint 10.1.255.4), selected
                Protocol next hop: 10.1.255.4-100<c>
                Composite next hop: 0x6fb1928 593 INH Session ID: 0
                Indirect next hop: 0x7164d3c 1048575 INH Session ID: 0
                State: <Active Int Ext>
                Local AS: 65000 Peer AS: 65000
                Age: 20:14      Metric: 10      Metric2: 2
                Validation State: unverified
                Task: BGP_65000.10.1.255.4
                Announcement bits (2): 0-KRT 5-IS-IS
                AS path: I
                Communities: color:0:100
                Accepted
```

```
                Localpref: 100
                Router ID: 10.1.255.4
                Route-nexthop:
                Indr (0x76b7238) 10.1.255.4-100<c>
                  Krt_cnh (0x6fb1928) Index:593
                    Krt_inh (0x7164d3c) Index:1048575
                      Tun-comp (0x76b71cc) Index:592
                Thread: junos-main
                Composite next hops: 1
                        Protocol next hop: 10.1.255.4-100<c> Metric: 2
                        Composite next hop: 0x6fb1928 593 INH Session ID: 0
                        Indirect next hop: 0x7164d3c 1048575 INH Session ID: 0
                        Indirect path forwarding next hops: 1
                                Next hop type: Tunnel Composite
                                Tunnel type: IPoIP, (forwarding-nexthop), Reference count: 2,
nhid: 592
                                Destination address: 10.1.255.4-100<c>, Source address:
10.1.255.2
                                Tunnel endpoint: 10.1.255.4
                                10.1.255.4-100<c>/64 Originating RIB: inetcolor.0
                                  Metric: 2 Node path count: 1
                                  Forwarding nexthops: 1
                                        Next hop type: Tunnel Composite
                                        Tunnel type: IPoIP, (extended-attr), Reference count: 1,
nhid: 0
                                        Destination address: 10.1.255.4-100<c>, Source address:
10.1.255.2
```

Meaning

The output from R2 shows the expanded next hop for the `192.168.255.5` route. As R2 is an MX Series router, it sends a protocol next hop and an indirect next hop.

Verify End-to-End Reachability

Purpose

Verify R1 can ping R5 by using the `ping 192.168.255.5 source 192.168.255.1 count 2` operational mode command.

Action

```
user@R1>ping 192.168.255.5 source 192.168.255.1 count 2
PING 192.168.255.5 (192.168.255.5): 56 data bytes
64 bytes from 192.168.255.5: icmp_seq=0 ttl=62 time=6.009 ms
64 bytes from 192.168.255.5: icmp_seq=1 ttl=62 time=5.398 ms

--- 192.168.255.5 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 5.398/5.704/6.009/0.306 ms
```

Meaning

The output from R1 shows that R1 can ping R5.

Example: Configure an IPoIP Tunnel with LDP tunnel in an MPLS Cloud, Resolved through inetcolor.0 Using BGP Signaling

IN THIS SECTION

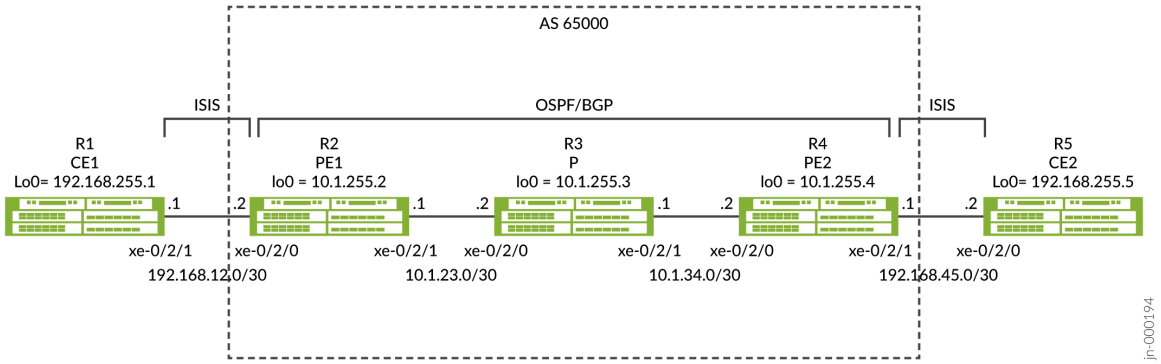
- CLI Quick Configuration | **189**
- Procedure | **192**
- Results | **196**

In an MPLS environment with LDP enabled, BGP routes get resolved through LDP on inet.3 table because MPLS has higher preference than IP.

If you still prefer to have your routes resolved through IP-over-IP in the MPLS environment, you can do so by creating an inetcolor.0 table that assigns higher preference for IP-over-IP and resolve chosen routes through IP-over-IP. To enable this feature by using BGP, the route resolution is performed on the remote end device of the tunnel and with the export policy configured on the remote device, routes are received and advertised through BGP signaling. This example shows how to configure this by using the BGP protocol configuration.

In this example, we are exchanging routes from R1 to R5 and vice versa through IP-over-IP dynamic tunnels established between the R2 and R4. Routes from R1 are exported to R2 and routes from R5 are exported to R4, by using the protocol IS-IS. We configure a unicast IPIP tunnel *Tunnel-01* from R2 to R4 and another tunnel *Tunnel-01* from R4 to R2. Route prefixes that are generated within the network

masks from the configured destination-networks of the peer device are used for creating the tunnel and traffic flows in the opposite direction of the routes in the tunnel.



CLI Quick Configuration

R1

```
set interfaces xe-0/2/0 unit 0 description R1-to-R2
set interfaces xe-0/2/0 unit 0 family inet address 192.168.12.1/30
set interfaces xe-0/2/0 unit 0 family iso
set interfaces lo0 unit 0 family inet address 192.168.255.1/32
set interfaces lo0 unit 0 family iso address 49.0001.1920.1682.5501.00
set routing-options router-id 192.168.255.1
set protocols isis interface xe-0/2/0.0
set protocols isis interface lo0.0
set protocols isis level 1 disable
```

R2

```
set interfaces xe-0/2/0 description R2-to-R1
set interfaces xe-0/2/0 unit 0 family inet address 192.168.12.2/30
set interfaces xe-0/2/0 unit 0 family iso
set interfaces xe-0/2/1 description R2-to-R3
set interfaces xe-0/2/1 unit 0 family inet address 10.1.23.1/30
set interfaces xe-0/2/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.1.255.2/32
set interfaces lo0 unit 0 family iso address 49.0001.0010.1255.0002.00
set policy-options policy-statement export-bgp term t1 from protocol bgp
set policy-options policy-statement export-bgp term t1 then accept
set policy-options policy-statement export-tunnel-route term t1 from route-filter
```

```
192.168.255.1/32 exact
set policy-options policy-statement export-tunnel-route term t1 then tunnel-attribute set tunnel-
attr-01
set policy-options policy-statement export-tunnel-route term t1 then next-hop self
set policy-options policy-statement export-tunnel-route term t1 then accept
set policy-options tunnel-attribute tunnel-attr-01 tunnel-type ipip
set policy-options tunnel-attribute tunnel-attr-01 tunnel-color 100
set policy-options tunnel-attribute tunnel-attr-01 remote-end-point 10.1.255.4
set routing-options router-id 10.1.255.2
set routing-options autonomous-system 65000
set routing-options dynamic-tunnels Tunnel-01 source-address 10.1.255.2
set routing-options dynamic-tunnels Tunnel-01 bgp-signal
set routing-options dynamic-tunnels Tunnel-01 destination-networks 10.1.255.0/24 colors 100
set protocols bgp group iBGP type internal
set protocols bgp group iBGP local-address 10.1.255.2
set protocols bgp group iBGP family inet unicast extended-nexthop-tunnel
set protocols bgp group iBGP export export-tunnel-route
set protocols bgp group iBGP neighbor 10.1.255.4
set protocols isis interface xe-0/2/0.0
set protocols isis interface lo0.0
set protocols isis level 1 disable
set protocols isis export export-bgp
set protocols ldp interface xe-0/2/1.0
set protocols mpls interface xe-0/2/1.0
set protocols ospf area 0.0.0.0 interface xe-0/2/1.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

R3

```
set interfaces xe-0/2/0 unit 0 description R3-to-R2
set interfaces xe-0/2/0 unit 0 family inet address 10.1.23.2/30
set interfaces xe-0/2/0 unit 0 family mpls
set interfaces xe-0/2/1 unit 0 description R3-to-R4
set interfaces xe-0/2/1 unit 0 family inet address 10.1.34.1/30
set interfaces xe-0/2/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.1.255.3/32
set routing-options router-id 10.1.255.3
set protocols ldp interface xe-0/2/0.0
set protocols ldp interface xe-0/2/1.0
set protocols mpls interface xe-0/2/0.0
set protocols mpls interface xe-0/2/1.0
set protocols ospf area 0.0.0.0 interface xe-0/2/0.0
```

```
set protocols ospf area 0.0.0.0 interface xe-0/2/1.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

R4

```
set interfaces xe-0/2/0 unit 0 description R4-to-R3
set interfaces xe-0/2/0 unit 0 family inet address 10.1.34.2/30
set interfaces xe-0/2/0 unit 0 family mpls
set interfaces xe-0/2/1 unit 0 description R4-to-R5
set interfaces xe-0/2/1 unit 0 family inet address 192.168.45.1/30
set interfaces xe-0/2/1 unit 0 family iso
set interfaces lo0 unit 0 family inet address 10.1.255.4/32
set interfaces lo0 unit 0 family iso address 49.0001.0010.1255.0004.00
set policy-options policy-statement export-bgp term t1 from protocol bgp
set policy-options policy-statement export-bgp term t1 then accept
set policy-options policy-statement export-tunnel-route term t1 from route-filter
192.168.255.5/32 exact
set policy-options policy-statement export-tunnel-route term t1 then tunnel-attribute set tunnel-
attr-01
set policy-options policy-statement export-tunnel-route term t1 then next-hop self
set policy-options policy-statement export-tunnel-route term t1 then accept
set policy-options tunnel-attribute tunnel-attr-01 tunnel-type ipip
set policy-options tunnel-attribute tunnel-attr-01 tunnel-color 100
set policy-options tunnel-attribute tunnel-attr-01 remote-end-point 10.1.255.2
set routing-options router-id 10.1.255.4
set routing-options autonomous-system 65000
set routing-options dynamic-tunnels Tunnel-01 source-address 10.1.255.4
set routing-options dynamic-tunnels Tunnel-01 bgp-signal
set routing-options dynamic-tunnels Tunnel-01 destination-networks 10.1.255.0/24 colors 100
set protocols bgp group iBGP type internal
set protocols bgp group iBGP local-address 10.1.255.4
set protocols bgp group iBGP family inet unicast extended-nexthop-tunnel
set protocols bgp group iBGP export export-tunnel-route
set protocols bgp group iBGP neighbor 10.1.255.2
set protocols isis interface xe-0/2/1.0
set protocols isis interface lo0.0
set protocols isis level 1 disable
set protocols isis export export-bgp
set protocols ldp interface xe-0/2/0.0
set protocols mpls interface xe-0/2/0.0
```

```
set protocols ospf area 0.0.0.0 interface xe-0/2/0.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

R5

```
set interfaces xe-0/2/0 unit 0 description R5-to-R4
set interfaces xe-0/2/0 unit 0 family inet address 192.168.45.2/30
set interfaces xe-0/2/0 unit 0 family iso
set interfaces lo0 unit 0 family inet address 192.168.255.5/32
set interfaces lo0 unit 0 family iso address 49.0001.1920.1682.5505.00
set routing-options router-id 192.168.255.5
set protocols isis interface xe-0/2/0.0
set protocols isis interface lo0.0
set protocols isis level 1 disable
```

Procedure

Step-by-Step Procedure for R1

R1 and R5 have a similar configuration so we will only show the step-by-step procedure for R1.

1. Enter configuration mode on R1.
2. Configure the interface connected to R2 and interface lo0. Make sure to configure both family `inet` and `iso`. Family `iso` is needed for protocol IS-IS.

```
[edit]
user@R1# set interfaces xe-0/2/0 unit 0 description R1-to-R2
user@R1# set interfaces xe-0/2/0 unit 0 family inet address 192.168.12.1/30
user@R1# set interfaces xe-0/2/0 unit 0 family iso
user@R1# set interfaces lo0 unit 0 family inet address 192.168.255.1/32
user@R1# set interfaces lo0 unit 0 family iso address 49.0001.1920.1682.5501.00
```

3. Configure the Router ID.

```
[edit]
user@R1# set routing-options router-id 192.168.255.1
```

4. Configure protocols IS-IS. Routes are advertised between R1 and R2 using the IS-IS protocol.

```
[edit]
user@R1# set protocols isis interface xe-0/2/0.0
user@R1# set protocols isis interface lo0.0
user@R1# set protocols isis level 1 disable
```

5. Enter `commit` on R1 from the configuration mode.

Step-by-Step Procedure for R2

R2 and R4 have a similar configuration so we will only show the step-by-step procedure for R2.

1. Enter configuration mode on R2.

2. Configure the interfaces connected to R1 and R3 and interface lo0. Ensure to configure both family`inet` and `iso` on the interface connected to R1 and lo0, and to configure both family`inet` and `mpls` on the interface connected to R3.

```
[edit]
user@R2# set interfaces xe-0/2/0 description R2-to-R1
user@R2# set interfaces xe-0/2/0 unit 0 family inet address 192.168.12.2/30
user@R2# set interfaces xe-0/2/0 unit 0 family iso
user@R2# set interfaces xe-0/2/1 description R2-to-R3
user@R2# set interfaces xe-0/2/1 unit 0 family inet address 10.1.23.1/30
user@R2# set interfaces xe-0/2/0 unit 0 family mpls
user@R2# set interfaces lo0 unit 0 family inet address 10.1.255.2/32
user@R2# set interfaces lo0 unit 0 family iso address 49.0001.0010.1255.0002.00
```

3. Configure protocols IS-IS for the interface connected to R1. The export policy to advertise BGP routes into IS-IS is shown in the policy configuration step.

```
[edit]
user@R2# set protocols isis interface xe-0/2/0.0
user@R2# set protocols isis interface lo0.0
user@R2# set protocols isis level 1 disable
user@R2# set protocols isis export export-bgp
```

4. Configure the OSPF protocol for the interface connected to R3 for lo0 reachability.

```
[edit]
user@R2# set protocols ospf area 0.0.0.0 interface xe-0/2/1.0
user@R2# set protocols ospf area 0.0.0.0 interface lo0.0
```

5. Configure the LDP and MPLS protocols for the interface connected to R3.

```
[edit]
user@R2# set protocols ldp interface xe-0/2/1.0
user@R2# set protocols mpls interface xe-0/2/1.0
```

6. Configure the `router-id` and `autonomous-system` under the `routing-options` hierarchy, and configure IBGP between R2 and R4. The import policy to add a community to the routes learned using BGP and the export policy to advertise IS-IS routes into BGP and set the tunnel attributes is shown in the policy configuration step. Ensure to include the `extended-nexthop-tunnel` option with the `family inet unicast` configuration to allow resolution using the inetcolor.0 table.

```
[edit]
user@R2# set routing-options router-id 10.1.255.2
user@R2# set routing-options autonomous-system 65000
user@R2# set protocols bgp group iBGP type internal
user@R2# set protocols bgp group iBGP local-address 10.1.255.2
user@R2# set protocols bgp group iBGP family inet unicast extended-nexthop-tunnel
user@R2# set protocols bgp group iBGP export export-tunnel-route
user@R2# set protocols bgp group iBGP neighbor 10.1.255.4
```

7. Configure routing options on R2 to create a tunnel from R2 to R4. The `bgp-signal` option enables the tunnel creation signaled by BGP. The `colors` configuration option allows the tunnel to be created in the inetcolor.0 route table.

```
[edit]
user@R2# set routing-options dynamic-tunnels Tunnel-01 source-address 10.1.255.2
user@R2# set routing-options dynamic-tunnels Tunnel-01 bgp-signal
user@R2# set routing-options dynamic-tunnels Tunnel-01 destination-networks 10.1.255.0/24
colors 100
```

8. Configure the policies that were applied during the previous configuration steps. The *export-bgp* policy advertises BGP routes into IS-IS. The *export-tunnel-route* policy advertises the IS-IS route

from R1 into BGP with the tunnel-attribute and changes the next-hop to R2. The *tunnel-attr-01* tunnel-attribute sets the tunnel-type, the tunnel endpoint, and the color matched on in the colors configuration for the dynamic tunnel.

```
[edit]
user@R2# set policy-options policy-statement export-bgp term t1 from protocol bgp
user@R2# set policy-options policy-statement export-bgp term t1 then accept
user@R2# set policy-options policy-statement export-tunnel-route term t1 from route-filter
192.168.255.1/32 exact
user@R2# set policy-options policy-statement export-tunnel-route term t1 then tunnel-
attribute set tunnel-attr-01
user@R2# set policy-options policy-statement export-tunnel-route term t1 then next-hop self
user@R2# set policy-options policy-statement export-tunnel-route term t1 then accept
user@R2# set policy-options tunnel-attribute tunnel-attr-01 tunnel-type ipip
user@R2# set policy-options tunnel-attribute tunnel-attr-01 tunnel-color 100
user@R2# set policy-options tunnel-attribute tunnel-attr-01 remote-end-point 10.1.255.4
```

9. Enter commit from the configuration mode.

Step-by-Step Procedure for R3

1. Enter configuration mode on R3.
2. Configure the interfaces connected to R2 and R4 and interface lo0. Ensure to configure both family inet and mpls on the interfaces connected to R2 and R4.

```
[edit]
user@R3# set interfaces xe-0/2/0 unit 0 description R3-to-R2
user@R3# set interfaces xe-0/2/0 unit 0 family inet address 10.1.23.2/30
user@R3# set interfaces xe-0/2/0 unit 0 family mpls
user@R3# set interfaces xe-0/2/1 unit 0 description R3-to-R4
user@R3# set interfaces xe-0/2/1 unit 0 family inet address 10.1.34.1/30
user@R3# set interfaces xe-0/2/1 unit 0 family mpls
user@R3# set interfaces lo0 unit 0 family inet address 10.1.255.3/32
```

3. Configure the Router ID.

```
[edit]
user@R3# set routing-options router-id 10.1.255.3
```

4. Configure the OSPF protocol for the interfaces connected to R2 and R4 for lo0 reachability.

```
[edit]
user@R3# set protocols ospf area 0.0.0.0 interface xe-0/2/0.0
user@R3# set protocols ospf area 0.0.0.0 interface xe-0/2/1.0
user@R3# set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

5. Configure the LDP and MPLS protocols for the interfaces connected to R2 and R4.

```
[edit]
user@R2# set protocols ldp interface xe-0/2/0.0
user@R2# set protocols ldp interface xe-0/2/1.0
user@R2# set protocols mpls interface xe-0/2/0.0
user@R2# set protocols mpls interface xe-0/2/1.0
```

6. Enter `commit` from configuration mode on R3 device.

Results

You can check your configurations by using the following show commands from configuration mode.

Here's how you can check the configurations on R2 device:

`user@R2# show interfaces`

```
xe-0/2/0 {
    description R2-to-R1;
    unit 0 {
        family inet {
            address 192.168.12.2/30;
        }
        family iso;
    }
}
xe-0/2/1 {
    description R2-to-R3;
    unit 0 {
        family inet {
            address 10.1.23.1/30;
        }
        family mpls;
```

```
    }
}
lo0 {
    apply-groups-except global;
    unit 0 {
        family inet {
            address 10.1.255.2/32;
        }
        family iso {
            address 49.0001.0010.1255.0002.00;
        }
    }
}
```

user@R2# show protocols

```
bgp {
    group iBGP {
        type internal;
        local-address 10.1.255.2;
        family inet {
            unicast {
                extended-nexthop-tunnel;
            }
        }
        export export-tunnel-route;
        neighbor 10.1.255.4;
    }
}
isis {
    interface xe-0/2/0.0;
    interface lo0.0;
    level 1 disable;
    export export-bgp;
}
ldp {
    interface xe-0/2/1.0;
}
mpls {
    interface xe-0/2/1.0;
}
ospf {
```

```
    area 0.0.0.0 {
        interface xe-0/2/1.0;
        interface lo0.0;
    }
}
```

user@R2# show routing-options

```
router-id 10.1.255.2;
autonomous-system 65000;
dynamic-tunnels {
    Tunnel-01 {
        source-address 10.1.255.2;
        bgp-signal;
        destination-networks {
            10.1.255.0/24 colors 100;
        }
    }
}
```

user@R2# show policy-options

```
policy-statement export-bgp {
    term t1 {
        from protocol bgp;
        then accept;
    }
}
policy-statement export-tunnel-route {
    term t1 {
        from {
            route-filter 192.168.255.1/32 exact;
        }
        then {
            tunnel-attribute set tunnel-attr-01;
            next-hop self;
            accept;
        }
    }
}
tunnel-attribute tunnel-attr-01 {
```

```
    tunnel-type ipip;
    tunnel-color 100;
    remote-end-point 10.1.255.4;
}
```

Verification

IN THIS SECTION

- Verify BGP Routes | **199**
- Verify Received Routes | **200**
- Verify the Dynamic Tunnel | **201**
- Verify Route Resolution | **202**
- Verify End-to-End Reachability | **202**

Verify BGP Routes

Purpose

Verify routes sent by using BGP protocol.

Action

R2

```
user@R2> show route protocol bgp

inet.0: 18 destinations, 18 routes (18 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.255.5/32   *[BGP/170] 01:21:51, MED 10, localpref 100, from 10.1.255.4
                      AS path: I, validation-state: unverified
                    >  via Tunnel Composite, IPoIP (src 10.1.255.2 dest 10.1.255.4-100<c> tunnel-
endpoint 10.1.255.2)

inet.3: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
```

```
iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

mpls.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)

inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

inetcolor.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
```

Meaning

The output shows the routes from BGP.

Verify Received Routes

Purpose

Verify the routes received through BGP by using the following operational mode commands.

Action

R2

```
user@R2> show route receive-protocol bgp 10.1.255.4 192.168.255.5 extensive

inet.0: 18 destinations, 18 routes (18 active, 0 holddown, 0 hidden)
* 192.168.255.5/32 (1 entry, 1 announced)
     Accepted
     Nexthop: 10.1.255.4
     MED: 10
     Localpref: 100
     AS path: I
     Tunnel type: ipip, Tunnel color: 100, Remote end point: 10.1.255.2
```

Meaning

The R2 output indicates the routes received on the devices.

Verify the Dynamic Tunnel

Purpose

Verify the dynamic tunnel is up and BGP signaled.

Action

R2

```
user@R2> show dynamic-tunnels database
*- Signal Tunnels #- PFE-down
Table: inet.3

Destination-network: 10.1.255.0/24

*- Signal Tunnels #- PFE-down
Table: inetcolor.0

Destination-network: 10.1.255.0-0<c>/24

Tunnel to: 10.1.255.4-100<c>/64
  Reference count: 3
  Next-hop type: IPoIP (bgp-signalled forwarding-nexthop)     Tunnel-endpoint: 10.1.255.2
    Source address: 10.1.255.2
    Next hop: tunnel-composite, 0x76b7238, nhid 592
      Reference count: 2
      Ingress Route: [OSPF] 10.1.255.4/32, via metric 2
      Tunnel Endpoint Ingress Route: [Direct] 10.1.255.2/32
      Traffic Statistics: Packets 0, Bytes 0
      State: Up
  Aggregate Traffic Statistics:
    Tunnel Encapsulation: Dest 10.1.255.2, Src 10.1.255.2, IPoIP, Tunnel-Id 1
    Traffic Statistics: Packets 0, Bytes 0
```

Meaning

The R2 output indicates the tunnel is up and BGP signaled.

Verify Route Resolution

Purpose

To verify the route resolution of the route in the inetcolor.0 table, use the `show route table inetcolor.0` operational mode commands.

Action

```
user@R2> show route table inetcolor.0

inetcolor.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.255.0-0<c>/24
                   *[Tunnel/305] 01:33:33
                       Tunnel
10.1.255.4-100<c>/64
                   *[Tunnel/305] 01:28:44, metric 2
                       Tunnel Composite, BGP-Signal (src 10.1.255.2 dest 10.1.255.4-100<c>)
```

Meaning

The R2 output indicates the tunnel to `10.1.255.4` is BGP signaled.

Verify End-to-End Reachability

Purpose

Verify R1 can ping R5 by using the `ping 192.168.255.5 source 192.168.255.1 count 2` operational mode command.

Action

```
user@R1>ping 192.168.255.5 source 192.168.255.1 count 2
PING 192.168.255.5 (192.168.255.5): 56 data bytes
64 bytes from 192.168.255.5: icmp_seq=0 ttl=63 time=2.784 ms
64 bytes from 192.168.255.5: icmp_seq=1 ttl=63 time=1.904 ms

--- 192.168.255.5 ping statistics ---
```

```
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.904/2.344/2.784/0.440 ms
```

Meaning

The output from R1 shows that R1 can ping R5.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
19.2R1	Starting in Junos Release 19.2R1, on MX Series routers with MPCs and MICs, carrier supporting carrier (CSC) architecture can be deployed with MPLS-over-UDP tunnels carrying MPLS traffic over dynamic IPv4 UDP tunnels that are established between supporting carrier's PE devices.
18.3R1	Starting in Junos OS Release 18.3R1, MPLS-over-UDP tunnels are supported on PTX Series routers and QFX Series switches.
18.2R1	Starting in Junos OS Release 18.2R1, on PTX series routers and QFX10000 with unidirectional MPLS-over-UDP tunnels, you must configure the remote PE device with an input filter for MPLS-over-UDP packets, and an action for decapsulating the IP and UDP headers for forwarding the packets in the reverse tunnel direction.
17.4R1	Starting in Junos OS Release 17.4R1, on MX Series routers, the next-hop-based dynamic MPLS-over-UDP tunnels are signaled using BGP encapsulation extended community.
17.1R1	Starting in Junos OS Release 17.1, on MX Series routers with MPCs and MICs, the scaling limit of MPLS-over-UDP tunnels is increased.

RELATED DOCUMENTATION

Configuring GRE Tunnels for Layer 3 VPNs

ip-tunnel-rpf-check

PART 3

MPLS Traffic

- Managing MPLS Traffic | **205**
- Protecting MPLS Traffic | **360**
- Measuring MPLS Traffic | **484**

CHAPTER 5

Managing MPLS Traffic

IN THIS CHAPTER

- Bidirectional Forwarding Detection (BFD) for MPLS | **205**
- Firewall Filters for MPLS | **218**
- System Log Messages and SNMP Traps for MPLS | **232**
- Load Balancing MPLS Traffic | **234**
- Shared Risk Link Groups for MPLS | **275**

Bidirectional Forwarding Detection (BFD) for MPLS

IN THIS SECTION

- Configuring Bidirectional Forwarding Detection for MPLS (CLI Procedure) | **205**
- BFD-Triggered Local Repair for Rapid Convergence | **210**
- Configuring BFD for MPLS IPv4 LSPs | **213**
- Platform-Specific BFD for MPLS Behavior | **217**

Configuring Bidirectional Forwarding Detection for MPLS (CLI Procedure)

IN THIS SECTION

- Configuring BFD on Provider Edge and Provider Switches for an LDP-Based LSP | **206**
- Configuring BFD on Provider Edge and Provider Switches for an RSVP-Based LSP | **209**

You can configure the Bidirectional Forwarding Detection (BFD) protocol on detect failures in the MPLS label-switch path (LSP). The BFD protocol is a simple hello mechanism that detects failures in a network. Hello packets are sent at a specified, regular interval. A neighbor failure is detected when the routing device stops receiving a reply from the neighbor after a specified interval. BFD works with a wide variety of network environments and topologies. The failure detection timers for BFD have shorter time limits than those of the failure detection mechanisms for static routes, and thus provide faster detection. These timers are also adaptive. For example, a timer can adapt to a higher value if an adjacency fails, or a neighbor can negotiate a higher value than the one configured.

Use Feature Explorer to confirm platform and release support for specific features.

Review the "Platform-Specific BFD for MPLS Behavior" on page 217 section for notes related to your platform.

This topic describes configuring the provider edge (PE) switches and the provider switches to support for LDP-based LSPs and RSVP-based LSPs.

This topic includes:

Configuring BFD on Provider Edge and Provider Switches for an LDP-Based LSP

You can enable BFD for the LDP-based LSPs or RSVP-based LSPs associated with a specific forwarding equivalence class (FEC). Alternatively, you can configure an Operation Administration and Maintenance (OAM) ingress policy to enable BFD on a range of FEC addresses.

Before you configure BFD for an LDP-based based LSP, you must configure the basic components for an MPLS network:

- Configure two PE switches. See "Configuring MPLS on Provider Edge Switches Using IP-Over-MPLS" on page 83.
- Configure one or more provider switches. See "Configuring MPLS on EX8200 and EX4500 Provider Switches" on page 93.

To configure BFD on PE and provider switches:

1. Define an OAM policy:

```
[edit]
user@switch# set protocols ldp oam ingress-policy policy-name
```

2. Specify the FEC on which you want to enable OAM:

```
[edit]
user@switch# set protocols ldp oam fec address
```

3. Specify the minimum transmit and receive interval for the BFD configuration:

NOTE: If you configure the `minimum-interval` statement, you do not need to configure the `minimum-receive-interval` statement or the `minimum-transmit-interval` statement.

```
[edit]
user@switch# set protocols ldp oam bfd-liveness-detection minimum-interval time
```

or

```
[edit]
user@switch# set protocols ldp oam bfd-liveness-detection minimum-receive-interval time
user@switch# set protocols ldp oam bfd-liveness-detection minimum-transmit-interval time
```

4. Specify the detection time multiplier. The negotiated transmit interval multiplied by this value gives the detection time for the receiving system in Asynchronous mode:

```
[edit]
user@switch# set protocols ldp oam bfd-liveness-detection multiplier multiplier
```

5. Specify the minimum transmit interval (or the minimum receive interval).

```
[edit]
user@switch# set protocols ldp oam bfd-liveness-detection transmit-interval minimum-
interval time
```

6. Specify a threshold for detecting the adaptation of the detection time:

```
[edit]
user@switch# set protocols ldp oam bfd-liveness-detection detection-time threshold time
```

7. Configure route and next-hop action in the event of a BFD session failure event on the LDP-based LSP:

```
[edit]
user@switch# set protocols ldp oam bfd-liveness-detection failure-action action
```

> **NOTE**: When a BFD session goes down, you can configure the Junos OS to resignal the LSP path or to simply disable the LSP path. You can configure a standby LSP path to handle traffic while the primary LSP path is unavailable. The switch can automatically recover from LSP failures that can be detected by BFD. By default, if a BFD session fails, the event is simply logged.

8. Specify how long the BFD session must be up before adding the route or next hop. Specifying a time of 0 seconds causes the route or next hop to be added as soon as the BFD session comes back up.

```
[edit]
user@switch# set protocols ldp oam bfd-liveness-detection holddown-interval time
```

9. Enable tracing of FECs for LDP-based LSPs and specify a source address for sending probes. Then, specify a wait interval, after which to send the probe packet.

```
[edit]
user@switch# set protocols ldp oam periodic-traceroute source address
user@switch# set protocols ldp oam periodic-traceroute wait time
```

10. Specify the duration of the LSP ping interval in seconds:

```
[edit]
user@switch# set protocols ldp oam lsp-ping-interval time
```

11. Specify the action to be taken for the OAM policy:

```
[edit]
user@switch# set policy-options policy-statement policy-name then accept
```

12. Apply the BFD configurations at the MPLS hierarchy level for the configuration to inherit the statements in the configuration group:

```
[edit]
user@switch# set apply-groups MPLS
```

Configuring BFD on Provider Edge and Provider Switches for an RSVP-Based LSP

When BFD is configured for an RSVP-based LSP on the ingress switch, it is enabled on the primary path and on all standby secondary paths for that LSP. You can enable BFD for all LSPs on a switch or for specific LSPs. If you configure BFD for a specific LSP, whatever values configured globally for BFD are overridden on that LSP. The BFD sessions originate only at the ingress switch and terminate at the egress switch.

Before you configure BFD for an RSVP-based LSP, you must configure the basic components for an MPLS network:

- Configure two PE switches. See "Configuring MPLS on Provider Edge Switches Using IP-Over-MPLS" on page 83.
- Configure one or more provider switches. See "Configuring MPLS on EX8200 and EX4500 Provider Switches" on page 93.

To configure BFD on PE and provider switches:

1. Specify the minimum transmit and receive interval for the BFD configuration:

> **NOTE**: If you configure the `minimum-interval` statement, you do not need to configure the `minimum-receive-interval` statement or the `minimum-transmit-interval` statement.

```
[edit]
user@switch# set protocols mpls label-switched-path lsp-name oam bfd-liveness-detection
minimum-interval time
```

or

```
[edit]
user@switch# set protocols mpls label-switched-path lsp-name oam bfd-liveness-detection
minimum-receive-interval time
```

```
user@switch# set protocols mpls label-switched-path lsp-name oam bfd-liveness-detection
minimum-transmit-interval time
```

2. Specify the detection time multiplier. The negotiated transmit interval multiplied by this value gives the detection time for the receiving system in Asynchronous mode:

```
[edit]
user@switch# set protocols mpls label-switched-path lsp-name oam bfd-liveness-detection
multiplier multiplier
```

3. Specify the minimum transmit interval (or the minimum receive interval):

```
[edit]
user@switch# set protocols mpls label-switched-path lsp-name oam bfd-liveness-detection
transmit-interval minimum-interval time
```

4. Configure route and next-hop actions in the event of a BFD session failure event on the RSVP-based LSP:

```
[edit]
user@switch# set protocols mpls label-switched-path lsp-name oam bfd-liveness-detection
failure-action action
```

NOTE: When a BFD session goes down, you can configure the Junos OS to resignal the LSP path or to simply disable the LSP path. You can configure a standby LSP path to handle traffic while the primary LSP path is unavailable. The switch can automatically recover from LSP failures that can be detected by BFD. By default, if a BFD session fails, the event is simply logged if you do not specifically configure a failure action.

BFD-Triggered Local Repair for Rapid Convergence

IN THIS SECTION

- Understanding BFD-Triggered Local Protection | **211**

Understanding BFD-Triggered Local Protection

IN THIS SECTION

- Purpose of BFD-Triggered Local Repair | 211
- Configuring BFD-Triggered Local Repair | 212
- Disabling BFD-Triggered Local Repair | 212

The time it takes for a network to converge following a link or node failure can vary dramatically based on a number of factors, including network size, the protocols used, and network design. However, while each particular convergence event is different, the process of convergence is essentially consistent. The failure is detected, the failure is reported (flooded) in the network, an alternate path is found for traffic, and the forwarding plane is updated to pass traffic on a new path.

This overview discusses how Bidirectional Forwarding Detection (BFD)-triggered local repair contributes to a quicker restoration time for rapid convergence in an MPLS network.

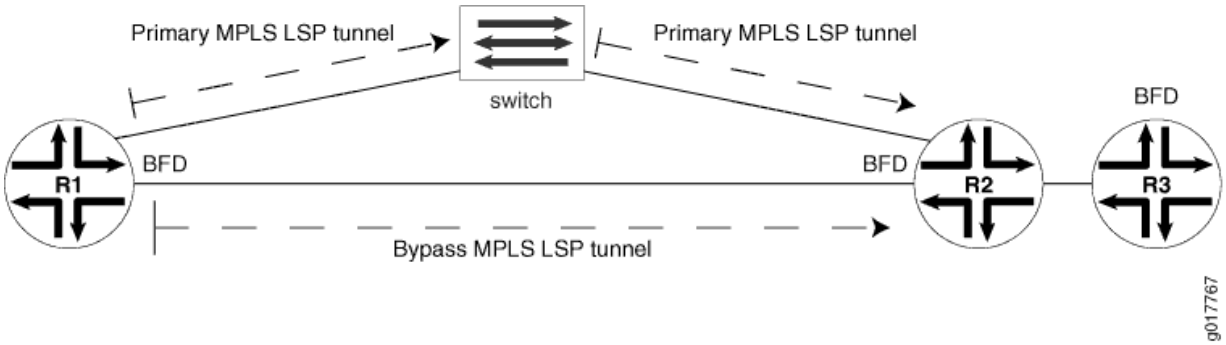
Purpose of BFD-Triggered Local Repair

In Junos OS, general MPLS traffic protection for RSVP-signaled label-switched path (LSP) failures is provided by several complementary mechanisms. These protection mechanisms include local protection (fast reroute, link protection, and node-link protection) and path protection (primary and secondary paths). Local protection in conjunction with path protection can provide minimum packet loss for an LSP, and control the way the LSP is rerouted after a failure. Traditionally, both types of protection rely on fast detection of connectivity failure at the physical level. However, for transmission media without fast physical level detection, Junos OS supports BFD and MPLS ping for fast failure detection.

With links between routers, when a route goes down, the routing protocol process recalculates the next best path. When MPLS fast reroute (FRR) is enabled, ifl messages are flooded to all Flexible PIC Concentrators (FPCs). The edge FPC enables the bypass MPLS LSP tunnel. Lastly, all routes are repaired and sent through the bypass MPLS LSP tunnel. The amount of time it takes to repair all routes is proportional to the number of routes.

This repair scenario becomes more difficult when a switch lies between two links. See Figure 1.

Figure 10: Topology with BFD-Triggered Local Repair



When a link goes down at the remote end, the failure is not detected at the local end until the interior gateway protocol (IGP) goes down. To wait for the routing protocol process to recalculate the next best path takes too much time.

With BFD-triggered local repair enabled, the Packet Forwarding Engine completes the repair first, using the bypass MPLS LSP tunnel (that is preconfigured and installed), then informs the routing protocol process to recalculate a new route. By doing this, when the primary MPLS LSP tunnel goes down, the FPC can intermittently and immediately divert traffic to the FPC with the bypass MPLS LSP tunnel.

Using local repair in this way achieves a faster restoration time of less than 50 ms.

Configuring BFD-Triggered Local Repair

BFD-triggered local repair is not configurable, but is part of the default configuration.

BFD-triggered local repair works within the legacy Junos OS features MPLS-FRR, BFD for IGP, and loop-free alternates (LFAs).

Disabling BFD-Triggered Local Repair

By default, BFD-triggered local repair is enabled for all routing interfaces. If desired, you can disable BFD-triggered local repair at the [**edit routing-options**] hierarchy level.

To explicitly disable BFD-triggered local repair:

1. Include the `no-bfd-triggered-local-repair` statement at the [edit routing-options] hierarchy level:

```
user@host# set no-bfd-triggered-local-repair
```

2. (Optional) Verify your configuration settings before committing them by using the `show routing-options` command.

```
user@host# run show routing-options
```

Confirm your configuration by issuing the **show routing-options** command.

```
user@host# show routing-options
...
    no-bfd-triggered-local-repair;
}
```

NOTE: When you disable this feature, you must also restart routing by including the **graceful-restart** statement for the IGP. For example, for OSPF, this is accomplished by including the **graceful-restart** statement at the `[edit protocols ospf]` hierarchy level.

Configuring BFD for MPLS IPv4 LSPs

IN THIS SECTION

- Configuring BFD for RSVP-Signaled LSPs | **214**
- Configuring a Failure Action for the BFD Session on an RSVP LSP | **216**

You can configure Bidirectional Forwarding Detection (BFD) protocol on MPLS IPv4 LSPs as outlined in the Internet draft draft-ietf-bfd-mpls-02.txt, *BFD for MPLS LSPs*. BFD is used as a periodic Operation, Administration, and Maintenance (OAM) feature for LSPs to detect LSP data plane faults. You can configure BFD for LSPs that use either LDP or RSVP as the signaling protocol.

NOTE: BFD for MPLS IPv4 LSP is based on the Routing Engine and is not distributed. As a result, the minimum supported BFD timer interval is (100 ms * 3) per one LSP session, and for scaled LSP sessions, the minimum supported BFD timer interval is (300 ms * 3). As you increase the number of LSP sessions with BFD, you must also increase (scale) the interval timers to support the network.

> For Routing Engine switchover instances with nonstop active routing (NSR) support, the minimum supported BFD timer interval is (2.5 seconds * 3).

You can also use the LSP `ping` commands to detect LSP data plane faults. However, BFD has a couple of benefits: it requires less computer processing than LSP `ping` commands and can quickly detect faults in large numbers of LSPs (LSP `ping` commands must be issued for each LSP individually). On the other hand, BFD cannot be used to verify the control plane against the data plane at the egress LSR, which is possible when an LSP `ping` echo request is associated with a forwarding equivalence class (FEC).

The BFD failure detection timers are adaptive and can be adjusted to be more or less aggressive. For example, the timers can adapt to a higher value if the adjacency fails, or a neighbor can negotiate a higher value for a timer than the configured value. The timers adapt to a higher value when a BFD session flap occurs more than three times in a span of 15 seconds. A back-off algorithm increases the receive (Rx) interval by two if the local BFD instance is the reason for the session flap. The transmission (Tx) interval is increased by two if the remote BFD instance is the reason for the session flap. You can use the `clear bfd adaptation` command to return BFD interval timers to their configured values. The `clear bfd adaptation` command is hitless, meaning that the command does not affect traffic flow on the routing device.

Starting from Junos OS Release 13.2R4, 13.3R2, and 14.1, you can set the time interval between LSP ping messages and the number of LSP ping responses, respectively, after which the Bidirectional Forwarding Detection (BFD) session is brought down. To so do, you configure the `lsp-ping-interval` statement and the `lsp-ping-multiplier` statement at the `[edit protocols mpls oam]` hierarchy level.

For configuration instructions for LDP-signaled LSPs, see "Configuring BFD for LDP LSPs" on page 1408. For configuration instructions for RSVP-signaled LSPs, see the following section.

Configuring BFD for RSVP-Signaled LSPs

BFD for RSVP supports unicast IPv4 LSPs. When BFD is configured for an RSVP LSP on the ingress router, it is enabled on the primary path and on all standby secondary paths for that LSP. The source IP address for outgoing BFD packets from the egress side of an MPLS BFD session is based on the outgoing interface IP address. You can enable BFD for all LSPs on a router or for specific LSPs. If you configure BFD for a specific LSP, whatever values configured globally for BFD are overridden. The BFD sessions originate only at the ingress router and terminate at the egress router.

An error is logged whenever a BFD session for a path fails. The following example shows how BFD for RSVP LSP log messages might appear:

```
RPD_MPLS_PATH_BFD_UP: MPLS BFD session for path path1 up on LSP R0_to_R3
RPD_MPLS_PATH_BFD_DOWN: MPLS BFD session for path path1 down on LSP R0_to_R3
```

You can configure BFD for all of the RSVP LSPs on the router, a specific LSP, or the primary path of a specific LSP. To configure BFD for RSVP LSPs, include the `oam` and `bfd-liveness-detection` statements.

```
oam {
    bfd-liveness-detection {
        failure-action {
            make-before-break teardown-timeout seconds;
            teardown;
        }
        failure-action teardown;
        minimum-interval milliseconds;
        minimum-receive-interval milliseconds;
        minimum-transmit-interval milliseconds;
        multiplier detection-time-multiplier;
    }
    lsp-ping-interval time-interval;
    lsp-ping-multiplier multiplier;
}
```

You can configure this statement at the following hierarchy levels:

- `[edit protocols mpls]`
- `[edit protocols mpls label-switched-path lsp-name]`
- `[edit protocols mpls label-switched-path lsp-name primary path-name]`

The `bfd-liveness-detection` statement includes the following options:

- `minimum-interval`—Specifies the minimum transmit and receive interval.
- `minimum-receive-interval`—Specifies the minimum receive interval. The range is from 1 through 255,000 milliseconds.
- `minimum-transmit-interval`—Specifies the minimum transmit interval. The range is from 1 through 255,000 milliseconds.
- `lsp-ping-multiplier`—Specifies the detection time multiplier. The range is from 1 through 255.

NOTE: To avoid triggering false negatives, configure a BFD fault detection time that is longer than the fast reroute time.

You can also configure the `lsp-ping-interval` option to adjust the time interval between LSP pings. The LSP ping command for RSVP-signaled LSPs is `ping mpls rsvp`. For more information on the `ping mpls rsvp` command, see the CLI Explorer.

Configuring a Failure Action for the BFD Session on an RSVP LSP

When the BFD session for an RSVP LSP goes down, the LSP is torn down and resignaled. Traffic can be switched to a standby LSP, or you can simply tear down the LSP path. Any actions performed are logged.

When a BFD session for an RSVP LSP path goes down, you can configure the Junos OS to resignal the LSP path or to simply disable the LSP path. A standby LSP path could be configured to handle traffic while the primary LSP path is unavailable. The router can automatically recover from LSP failures that can be detected by BFD. By default, if a BFD session fails, the event is simply logged.

To enable the Junos OS to tear down an RSVP LSP path in the event of a BFD event, include the `failure-action` statement:

```
failure-action {
    make-before-break teardown-timeout seconds;
    teardown;
}
```

For a list of the hierarchy levels at which you can include this statement, see the statement summary section for this statement.

You can configure either the `teardown` or `make-before-break` options:

- `teardown`—Causes the LSP path to be taken down and resignaled immediately.
- `make-before-break`—Causes the Junos OS to attempt to signal a new LSP path before tearing down the old LSP path. You can also configure the `teardown-timeout` option to automatically tear down the LSP after the time period specified if the attempt to resignal the LSP fails within the `teardown-timeout` interval. If you specify a value of 0 for the `teardown-timeout` interval, the LSP is taken down and resignaled immediately (the same behavior as when you configure the `teardown` option).

To configure a failure action for all of the RSVP LSPs, include the `failure-action` statement at the `[edit protocols mpls oam bfd-liveness-detection]` hierarchy level. To configure a failure action for a specific RSVP LSP, include the `failure-action` statement at the `[edit protocols mpls label-switched-path` *`lsp-name`* `oam bfd-liveness-detection]` hierarchy level.

To configure a failure action for a specific primary path, include the `failure-action` statement at the `[edit protocols mpls label-switched path` *`lsp-name`* `primary` *`path-name`* `oam bfd-liveness-detection]` hierarchy level. To configure a failure action for a specific secondary LSP path, include the `failure-action` statement at the `[edit protocols mpls label-switched-path` *`lsp-name`* `secondary` *`path-name`* `oam bfd-liveness-detection]` hierarchy level.

Platform-Specific BFD for MPLS Behavior

IN THIS SECTION

- Platform-Specific BFD for MPLS Behavior | 217

Use Feature Explorer to confirm platform and release support for specific features.

Use the following table to review platform-specific behaviors for your platform:

Platform-Specific BFD for MPLS Behavior

Platform	Difference
ACX Series	• ACX7000 Series devices support timer intervals of 1000ms or higher for MPLS LSP sessions.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
13.2R4	Starting from Junos OS Release 13.2R4, 13.3R2, and 14.1, you can set the time interval between LSP ping messages and the number of LSP ping responses, respectively, after which the Bidirectional Forwarding Detection (BFD) session is brought down.

RELATED DOCUMENTATION

Basic MPLS Configuration | 46

Firewall Filters for MPLS

IN THIS SECTION

- Configuring MPLS Firewall Filters and Policers on Routers | 218
- Overview of MPLS Firewall Filters on Loopback Interface | 227
- Configuring MPLS Firewall Filters and Policers on Switches | 229

Configuring MPLS Firewall Filters and Policers on Routers

IN THIS SECTION

- Configuring MPLS Firewall Filters | 218
- Examples: Configuring MPLS Firewall Filters | 219
- Configuring Policers for LSPs | 220
- Example: Configuring an LSP Policer | 222
- Configuring Automatic Policers | 223
- Writing Different DSCP and EXP Values in MPLS-Tagged IP Packets | 227
- Platform-Specific Behavior | 227

You can configure an MPLS firewall filter to count packets based on the EXP bits for the top-level MPLS label in a packet. You can also configure policers for MPLS LSPs.

The following sections discuss MPLS firewall filters and policers:

Configuring MPLS Firewall Filters

You can configure an MPLS firewall filter to count packets based on the EXP bits for the top-level MPLS label in a packet. You can then apply this filter to a specific interface. You can also configure a policer for the MPLS filter to police (that is, rate-limit) the traffic on the interface to which the filter is attached.

You can configure the following match criteria attributes for MPLS filters at the `[edit firewall family mpls filter` *`filter-name`* `term` *`term-name`* `from]` hierarchy level:

- `exp`

- exp-except

These attributes can accept EXP bits in the range 0 through 7. You can configure the following choices:

- A single EXP bit—for example, `exp 3;`
- Several EXP bits—for example, `exp 0, 4;`
- A range of EXP bits—for example, `exp [0-5];`

If you do not specify a match criterion (that is, you do not configure the `from` statement and use only the `then` statement with the `count` action keyword), all the MPLS packets passing through the interface on which the filter is applied will be counted.

You also can configure any of the following action keywords at the `[edit firewall family mpls filter` *`filter-name`* `term` *`term-name`* `then]` hierarchy level:

- count
- accept
- discard
- next
- policer

For more information about how to configure firewall filters, see the Routing Policies, Firewall Filters, and Traffic Policers User Guide. For more information about how to configure interfaces, see the Junos OS Network Interfaces Library for Routing Devices and the Junos OS Services Interfaces Library for Routing Devices.

Examples: Configuring MPLS Firewall Filters

The following examples illustrate how you might configure an MPLS firewall filter and then apply the filter to an interface. This filter is configured to count MPLS packets with EXP bits set to either 0 or 4.

The following shows a configuration for an MPLS firewall filter:

```
[edit firewall]
family mpls {
    filter expf {
        term expt0 {
            from {
                exp 0,4;
            }
```

```
            then {
                count counter0;
                accept;
            }
        }
    }
}
```

The following shows how to apply the MPLS firewall filter to an interface:

```
[edit interfaces]
so-0/0/0 {
    mtu 4474;
    encapsulation ppp;
    sonet-options {
        fcs 32;
    }
    unit 0 {
        point-to-point;
        family mpls {
            filter {
                input expf;
                output expf;
            }
        }
    }
}
```

The MPLS firewall filter is applied to the input and output of an interface (see the `input` and `output` statements in the preceding example).

Configuring Policers for LSPs

MPLS LSP policing allows you to control the amount of traffic forwarded through a particular LSP. Policing helps to ensure that the amount of traffic forwarded through an LSP never exceeds the requested bandwidth allocation. LSP policing is supported on regular LSPs, LSPs configured with DiffServ-aware traffic engineering, and multiclass LSPs. You can configure multiple policers for each multiclass LSP. For regular LSPs, each LSP policer is applied to all of the traffic traversing the LSP. The policer's bandwidth limitations become effective as soon as the total sum of traffic traversing the LSP exceeds the configured limit.

You configure the multiclass LSP and DiffServ-aware traffic engineering LSP policers in a filter. The filter can be configured to distinguish between the different class types and apply the relevant policer to each class type. The policers distinguish between class types based on the EXP bits.

You configure LSP policers under the `family any` filter. The `family any` filter is used because the policer is applied to traffic entering the LSP. This traffic might be from different families: IPv6, MPLS, and so on. You do not need to know what sort of traffic is entering the LSP, as long as the match conditions apply to all types of traffic.

You can configure only those match conditions that apply across all types of traffic. The following are the supported match conditions for LSP policers:

- `forwarding-class`
- `packet-length`
- `interface`
- `interface-set`

To enable a policer on an LSP, first you need to configure a policing filter and then include it in the LSP configuration. For information about how to configure policers, see the Routing Policies, Firewall Filters, and Traffic Policers User Guide.

To configure a policer for an LSP, specify a filter by including the `filter` option to the `policing` statement:

```
policing {
    filter filter-name;
}
```

You can include the `policing` statement at the following hierarchy levels:

- `[edit protocols mpls label-switched-path lsp-name]`
- `[edit protocols mpls static-label-switched-path lsp-name]`
- `[edit logical-systems logical-system-name protocols mpls label-switched-path lsp-name]`
- `[edit logical-systems logical-system-name protocols mpls static-label-switched-path lsp-name]`

LSP Policer Limitations

When configuring MPLS LSP policers, be aware of the following limitations:

- LSP policers are supported for packet LSPs only.
- LSP policers are supported for unicast next hops only. Multicast next hops are not supported.

- LSP policers are not supported on aggregated interfaces.
- The LSP policer runs before any output filters.
- Traffic sourced from the Routing Engine (for example, ping traffic) does not take the same forwarding path as transit traffic. This type of traffic cannot be policed.
- LSP policers work on all T Series routers and on M Series routers that have the Internet Processor II application-specific integrated circuit (ASIC).
- LSP policers are not supported for point-to-multipoint LSPs.

Example: Configuring an LSP Policer

The following example shows how you can configure a policing filter for an LSP:

```
[edit firewall]
policer police-ct1 {
    if-exceeding {
        bandwidth-limit 50m;
        burst-size-limit 1500;
    }
    then {
        discard;
    }
}
policer police-ct0 {
    if-exceeding {
        bandwidth-limit 200m;
        burst-size-limit 1500;
    }
    then {
        discard;
    }
}
family any {
    filter bar {
        term discard-ct0 {
            then {
                policer police-ct0;
                accept;
            }
        }
```

```
    }
    term discard-ct1 {
        then {
            policer police-ct1;
            accept;
        }
    }
}
```

Configuring Automatic Policers

Automatic policing of LSPs allows you to provide strict service guarantees for network traffic. Such guarantees are especially useful in the context of Differentiated Services for traffic engineered LSPs, providing better emulation for ATM wires over an MPLS network. For more information about Differentiated Services for LSPs, see DiffServ-Aware Traffic Engineering Introduction.

Differentiated Services for traffic engineered LSPs allow you to provide differential treatment to MPLS traffic based on the EXP bits. To ensure these traffic guarantees, it is insufficient to simply mark the traffic appropriately. If traffic follows a congested path, the requirements might not be met.

LSPs are guaranteed to be established along paths where enough resources are available to meet the requirements. However, even if the LSPs are established along such paths and are marked properly, these requirements cannot be guaranteed unless you ensure that no more traffic is sent to an LSP than there is bandwidth available.

It is possible to police LSP traffic by manually configuring an appropriate filter and applying it to the LSP in the configuration. However, for large deployments it is cumbersome to configure thousands of different filters. Configuration groups cannot solve this problem either, since different LSPs might have different bandwidth requirements, requiring different filters. To police traffic for numerous LSPs, it is best to configure automatic policers.

When you configure automatic policers for LSPs, a policer is applied to all of the LSPs configured on the router. However, you can disable automatic policing on specific LSPs.

NOTE: When you configure automatic policers for DiffServ-aware traffic engineering LSP, GRES is not supported.

NOTE: You cannot configure automatic policing for LSPs carrying CCC traffic.

The following sections describe how to configure automatic policers for LSPs:

Configuring Automatic Policers for LSPs

To configure automatic policers for standard LSPs (neither DiffServ-aware traffic engineered LSPs nor multiclass LSPs), include the `auto-policing` statement with either the `class all` *`policer-action`* option or the `class ct0` *`policer-action`* option:

```
auto-policing {
    class all policer-action;
    class ct0 policer-action;
}
```

You can include this statement at the following hierarchy levels:

- `[edit protocols mpls]`
- `[edit logical-systems` *`logical-system-name`* `protocols mpls]`

You can configure the following policer actions for automatic policers:

- `drop`—Drop all packets.
- `loss-priority-high`—Set the packet loss priority (PLP) to high.
- `loss-priority-low`—Set the PLP to low.

These policer actions are applicable to all types of LSPs. The default policer action is to do nothing.

Automatic policers for LSPs police traffic based on the amount of bandwidth configured for the LSPs. You configure the bandwidth for an LSP using the `bandwidth` statement at the `[edit protocols mpls label-switched-path` *`lsp-path-name`*`]` hierarchy level. If you have enabled automatic policers on a router, change the bandwidth configured for an LSP, and commit the revised configuration, the change does not take affect on the active LSPs. To force the LSPs to use the new bandwidth allocation, issue a `clear mpls lsp` command.

NOTE: You cannot configure automatic policers for LSPs that traverse aggregated interfaces or Multilink Point-to-Point Protocol (MLPPP) interfaces.

Configuring Automatic Policers for DiffServ-Aware Traffic Engineering LSPs

To configure automatic policers for DiffServ-aware traffic engineering LSPs and for multiclass LSPs, include the `auto-policing` statement:

```
auto-policing {
    class all policer-action;
    class ctnumber policer-action;
}
```

You can include this statement at the following hierarchy levels:

- `[edit protocols mpls]`
- `[edit logical-systems logical-system-name protocols mpls]`

You include either the `class all policer-action` statement or a `class ctnumber policer-action` statement for each of one or more classes (you can configure a different policer action for each class). For a list of the actions that you can substitute for the *policer-action* variable, see "Configuring Automatic Policers for LSPs" on page 224. The default policer action is to do nothing.

NOTE: You cannot configure automatic policers for LSPs that traverse aggregated interfaces or MLPPP interfaces.

Configuring Automatic Policers for Point-to-Multipoint LSPs

You can configure automatic policers for point-to-multipoint LSPs by including the `auto-policing` statement with either the `class all policer-action` option or the `class ct0 policer-action` option. You only need to configure the `auto-policing` statement on the primary point-to-multipoint LSP (for more information on primary point-to-multipoint LSPs, see Configuring the Primary Point-to-Multipoint LSP). No additional configuration is required on the subLSPs for the point-to-multipoint LSP. Point-to-multipoint automatic policing is applied to all branches of the point-to-multipoint LSP. In addition, automatic policing is applied to any local VRF interfaces that have the same forwarding entry as a point-to-multipoint branch.

The automatic policer configuration for point-to-multipoint LSPs is identical to the automatic policer configuration for standard LSPs. For more information, see "Configuring Automatic Policers for LSPs" on page 224.

Disabling Automatic Policing on an LSP

When you enable automatic policing, all of the LSPs on the router or logical system are affected. To disable automatic policing on a specific LSP on a router where you have enabled automatic policing, include the `policing` statement with the `no-auto-policing` option:

```
policing no-auto-policing;
```

You can include this statement at the following hierarchy levels:

- [edit protocols mpls label-switched-path *lsp-name*]
- [edit logical-systems *logical-system-name* protocols mpls label-switched-path *lsp-name*]

Example: Configuring Automatic Policing for an LSP

Configure automatic policing for a multiclass LSP, specifying different actions for class types `ct0`, `ct1`, `ct2`, and `ct3`.

```
[edit protocols mpls]
diffserv-te {
    bandwidth-model extended-mam;
}
auto-policing {
    class ct1 loss-priority-low;
    class ct0 loss-priority-high;
    class ct2 drop;
    class ct3 loss-priority-low;
}
traffic-engineering bgp-igp;
label-switched-path sample-lsp {
    to 3.3.3.3;
    bandwidth {
        ct0 11;
        ct1 1;
        ct2 1;
        ct3 1;
    }
}
interface fxp0.0 {
    disable;
}
```

```
interface t1-0/5/3.0;
interface t1-0/5/4.0;
```

Writing Different DSCP and EXP Values in MPLS-Tagged IP Packets

You can selectively set the DiffServ code point (DSCP) field of MPLS-tagged IPv4 and IPv6 packets to 0 without affecting output queue assignment, and continue to set the MPLS EXP field according to the configured rewrite table, which is based on forwarding classes. You can accomplish this by configuring a firewall filter for the MPLS-tagged packets.

Platform-Specific Behavior

Use Feature Explorer to confirm platform and release support for specific features.

Use the following tables to review platform-specific behavior for your platform:

Platform	Difference
PTX Series Routers	The PTX10003 router only supports regular LSPs.

Overview of MPLS Firewall Filters on Loopback Interface

IN THIS SECTION

- Benefits of Adding MPLS Firewall Filters on the Loopback Interface | **228**
- Guidelines and Limitations | **228**
- Platform-Specific MPLS firewall filters Behavior | **228**

Although all interfaces are important, the loopback interface might be the most important because it is the link to the Routing Engine, which runs and manages all the routing protocols. The loopback interface is a gateway for all the control traffic that enters the Routing Engine of the switch. You can control traffic by configuring a firewall filter on the loopback interface (lo0) on family mpls in QFX5100, QFX5110, QFX5200, and QFX5210 switches. Loopback firewall filters affect only traffic destined for the Routing Engine CPU. You can apply a loopback firewall filter only in the *ingress* direction (packets entering the interface). Starting with Junos OS Release 19.2R1, you can apply an MPLS firewall filter to a loopback interface on a label switch router (LSR) on QFX5100, QFX5110, QFX5200, and QFX5210 switches.

When you configure an MPLS firewall filter, you define filtering criteria (*terms, with match conditions*) for the packets and an *action* for the switch to take if the packets match the filtering criteria. Because you apply the filter to a loopback interface, you must explicitly specify the time to live (TTL) match condition under `family mpls` and set its TTL value to 1 (`ttl=1`). The TTL is an 8-bit (IPv4) header field that signifies the remaining time an IP packet has left before its life ends and is dropped. You can also match packets with other MPLS qualifiers such as `label`, `exp`, Layer 4 `source port`, and Layer 4 `destination port`.

Benefits of Adding MPLS Firewall Filters on the Loopback Interface

- Protects the Routing Engine by ensuring that it accepts traffic only from trusted networks.
- Helps protect the Routing Engine from denial-of-service attacks.
- Gives you the flexibility to match packets on the source port and destination port. For example, if you run a traceroute, you can selectively filter traffic by choosing either TCP or UDP.

Guidelines and Limitations

- You can apply a loopback firewall filter only in the *ingress* direction
- Only MPLS fields `label`, `exp`, `ttl=1` and Layer 4 fields `tcp` and `udp` port numbers are supported.
- Only `accept`, `discard`, and `count` actions are supported.
- You must explicitly specify `ttl=1` under `family mpls` to match on TLL packets.
- Filters applied on the loopback interface cannot be matched on the destination port (inner payload) of an IPv6 packet.
- You cannot apply a filter on packets that have more than two MPLS labels.
- You cannot specify a port range for TCP or UDP match conditions.
- Only 255 firewall terms are supported.

Platform-Specific MPLS firewall filters Behavior

Table 7: Platform-Specific Behavior

Platform	Difference
QFX5100 / QFX5110 / QFX5200 / QFX5210	Supports MPLS firewall filters on `lo0` from Junos OS 19.2R1; only ingress direction; `ttl=1` required under `family mpls`.
MX Series	Supports MPLS filters on `lo0`; `fxp0` supports only IP filters, not MPLS.
PTX Series	Supports MPLS filters on `lo0`.

Configuring MPLS Firewall Filters and Policers on Switches

IN THIS SECTION

- Configuring an MPLS Firewall Filter | 230
- Applying an MPLS Firewall Filter to an MPLS Interface | 230
- Applying an MPLS Firewall Filter to a Loopback Interface | 231
- Configuring Policers for LSPs | 231

You can configure firewall filters to filter MPLS traffic. To use an MPLS firewall filter, you must first configure the filter and then apply it to an interface you have configured for forwarding MPLS traffic. You can also configure a policer for the MPLS filter to police (that is, rate-limit) the traffic on the interface to which the filter is attached.

When you configure an MPLS firewall filter, you define the filtering criteria (terms, with match conditions) and an action for the switch to take if the packets match the filtering criteria.

NOTE: You can only configure MPLS filters in the ingress direction. Egress MPLS firewall filters are not supported.

Configuring an MPLS Firewall Filter

To configure an MPLS firewall filter:

1. Configure the filter name, term name, and at least one match condition—for example, match on MPLS packets with EXP bits set to either 0 or 4:

```
[edit firewall family mpls]
user@switch# set filter ingress-exp-filter term term-one from exp 0,4
```

2. In each firewall filter term, specify the actions to take if the packet matches all the conditions in that term—for example, count MPLS packets with EXP bits set to either 0 or 4:

```
[edit firewall family mpls filter ingress-exp-filter term term-one then]
user@switch# set count counter0
user@switch# set accept
```

3. When you are finished, follow the steps below to apply the filter to an interface.

Applying an MPLS Firewall Filter to an MPLS Interface

To apply the MPLS firewall filter to an interface you have configured for forwarding MPLS traffic (using the `family mpls` statement at the `[edit interfaces` *interface-name* `unit` *unit-number*`]` hierarchy level):

NOTE: You can apply firewall filters only to filter MPLS packets that enter an interface.

1. Apply the firewall filter to an MPLS interface—for example, apply the firewall filter to interface xe-0/0/5:

```
[edit interfaces]
user@switch# set xe-0/0/5 unit 0 family mpls filter input ingress-exp-filter
```

2. Review your configuration and issue the `commit` command:

```
[edit interfaces]
user@switch# commit
commit complete
```

Applying an MPLS Firewall Filter to a Loopback Interface

To apply an MPLS firewall filter to a loopback interface (lo0):

1. First, specify the packet format by using the *packet-format-match* command. You must restart the PFE every time you configure this command.
2. Configure the firewall filter match conditions and actions as described in "Configuring MPLS Firewall Filters and Policers on Switches" on page 229. You must explicitly set the TTL match condition to (`ttl=1`). You can also match packets with other MPLS qualifiers such as `label`, `exp`, and Layer 4 `source port`, and `destination port`.
3. Apply the filter to the loopback interface as an input filter.

```
[edit interfaces]
user@switch# set lo0 unit 0 family mpls filter input ingress-exp-filter
```

4. Review your configuration and issue the `commit` command:

```
[edit interfaces]
user@switch# commit
commit complete
```

The following is an example configuration.

```
set groups lo_mpls_filter interfaces lo0 unit 0 family mpls filter input mpls_lo
set groups lo_mpls_filter firewall family mpls filter mpls_lo term mpls_lo_term from ttl 1
set groups lo_mpls_filter firewall family mpls filter mpls_lo term mpls_lo_term from ip-version ipv4
protocol udp source-port 10
set groups lo_mpls_filter firewall family mpls filter mpls_lo term mpls_lo_term from ip-version ipv4
protocol udp destination-port 11
set groups lo_mpls_filter firewall family mpls filter mpls_lo term mpls_lo_term then count c1
set groups lo_mpls_filter firewall family mpls filter mpls_lo term mpls_lo_term then accept
```

Configuring Policers for LSPs

You can send traffic matched by an MPLS filter to a two-color policer or three-color policer. MPLS LSP policing allows you to control the amount of traffic forwarded through a particular LSP. Policing helps to ensure that the amount of traffic forwarded through an LSP never exceeds the requested bandwidth allocation. LSP policing is supported on regular LSPs, LSPs configured with DiffServ-aware traffic engineering, and multiclass LSPs. You can configure multiple policers for each multiclass LSP. For regular LSPs, each LSP policer is applied to all of the traffic traversing the LSP. The policer's bandwidth

limitations become effective as soon as the total sum of traffic traversing the LSP exceeds the configured limit.

You configure the multiclass LSP and DiffServ-aware traffic engineering LSP policers in a filter. The filter can be configured to distinguish between the different class types and apply the relevant policer to each class type. The policers distinguish between class types based on the EXP bits.

You configure LSP policers under the `family any` filter. The `family any` filter is used because the policer is applied to traffic entering the LSP. This traffic might be from different families: IPv6, MPLS, and so on. You do not need to know what sort of traffic is entering the LSP, as long as the match conditions apply to all types of traffic.

When configuring MPLS LSP policers, be aware of the following limitations:

- LSP policers are supported for packet LSPs only.
- LSP policers are supported for unicast next hops only. Multicast next hops are not supported.
- The LSP policer runs before any output filters.
- Traffic sourced from the Routing Engine (for example, ping traffic) does not take the same forwarding path as transit traffic. This type of traffic cannot be policed.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
19.2R1	Starting with Junos OS Release 19.2R1, you can apply an MPLS firewall filter to a loopback interface on a label switch router (LSR) on QFX5100, QFX5110, QFX5200, and QFX5210 switches.

System Log Messages and SNMP Traps for MPLS

Whenever an LSP makes a transition from up to down, or down to up, and whenever an LSP switches from one active path to another, the ingress router generates a system log message and sends an SNMP trap. The following shows a sample system log message:

```
RPD_MPLS_LSP_UP: MPLS LSP sheep1 up on primary(any) Route  192.168.1.1 192.168.1.2 192.168.1.3
RPD_MPLS_LSP_CHANGE: MPLS LSP sheep1 change on primary(any) Route  192.168.1.1 192.168.1.2
192.168.1.3
RPD_MPLS_LSP_DOWN: MPLS LSP sheep1 down on primary(any)
```

For information about the MPLS SNMP traps and the proprietary MPLS MIBs, see the Junos OS Network Management Administration Guide for Routing Devices.

System log messages for LSPs are generated by default. To disable the default logging of messages for LSPs, configure the `no-syslog` option under the `log-updown` statement:

```
log-updown {
    no-syslog;
}
```

To generate SNMP traps for LSPs, include the `trap` option to the `log-updown` statement:

```
log-updown {
    trap;
}
```

To generate SNMP traps whenever an LSP path goes down, include the `trap-path-down` option to the `log-updown` statement:

```
log-updown {
    trap-path-down;
}
```

To generate SNMP traps whenever an LSP path comes up, include the `trap-path-up` option to the `log-updown` statement:

```
log-updown {
    trap-path-up;
}
```

To disable the generation of system log messages, include the `no-syslog` option to the `log-updown` statement:

```
log-updown {
    no-syslog;
}
```

To disable the generation of SNMP traps, include the no-trap statement:

```
no-trap {
    mpls-lsp-traps;
    rfc3812-traps;
}
```

You can include this statement at the following hierarchy levels:

- [edit protocols mpls log-updown]
- [edit logical-systems *logical-system-name* protocols mpls log-updown]

For scalability reasons, only the ingress router generates SNMP traps. By default, MPLS issues traps for all configured LSPs. If you have many LSPs, the number of traps can become quite large. To disable the generation of SNMP traps, configure the no-trap statement.

The no-trap statement also includes the following options which allow you to block certain categories of MPLS SNMP traps:

- mpls-lsp-traps—Blocks the MPLS LSP traps defined in the **jnx-mpls.mib**, but allows the **rfc3812.mib** traps.
- rfc-3812-traps—Blocks the traps defined in the **rfc3812.mib**, but allows the MPLS LSP traps defined in the **jnx-mpls.mib**.

Load Balancing MPLS Traffic

IN THIS SECTION

- Configuring Load Balancing Based on MPLS Labels | **235**
- Example: Load-Balanced MPLS Network | **241**
- Router Configurations for the Load-Balanced MPLS Network | **242**
- Configuring Load Balancing Based on MPLS Labels on ACX Series Routers | **257**
- MPLS Encapsulated Payload Load-balancing Overview | **261**
- Configuring MPLS Encapsulated Payload for Load Balancing | **262**
- Policy-Based Multipath Routes Overview | **263**
- Understanding IP-Based Filtering and Selective Port Mirroring of MPLS Traffic | **269**

Configuring Load Balancing Based on MPLS Labels

Load balancing occurs on a per-packet basis for MPLS flows on supported platforms. Entropy, or random distribution, is essential for the uniform distribution of packets to their next hops. By default, when load balancing is used to help distribute traffic, Junos OS employs a hash algorithm to select a next-hop address to install into the forwarding table. Whenever the set of next hops for a destination changes, the next-hop address is reselected by means of the hash algorithm. You can configure how the hash algorithm is used to load-balance traffic across a set of equal-cost label switched paths (LSPs).

To ensure entropy for VPLS & VPWS traffic, Junos OS can create a hash based on data from the IP header and as many as three MPLS labels (the so-called top labels).

In some cases, as the number of network feature that use labels grows (such as MPLS Fast Reroute, and RFC 3107, RSVP and VPN) data in the top three labels can become static and thus not a sufficient source for entropy. Load balancing can become skewed as a result, or the incidence of out-of-order packet delivery may rise. For these cases, labels from the bottom of the label stack can be used (see Table 1, below for qualifications). Top labels and bottom labels cannot be used at the same time.

NOTE: MPC cards do not support the regular hash key configuration. For the MPC-based hash key configuration to be effective, you need an `enhanced-hash-key` configuration.

Load balancing is used to evenly distribute traffic when the following conditions apply:

- There are multiple equal-cost next hops over different interfaces to the same destination.
- There is a single next hop over an aggregated interface.

An LSP tends to load-balance its placement by randomly selecting one of the equal-cost next hops and using it exclusively. The random selection is made independently at each transit router, which compares Interior Gateway Protocol (IGP) metrics alone. No consideration is given to bandwidth or congestion levels.

This feature applies to aggregated Ethernet and aggregated SONET/SDH interfaces as well as multiple equal-cost MPLS next hops. In addition, on the T Series, MX Series, M120, and M320 routers only, you can configure load balancing for IPv4 traffic over Layer 2 Ethernet pseudowires. You can also configure load balancing for Ethernet pseudowires based on IP information. The option to include IP information in the hash key provides support for Ethernet circuit cross-connect (CCC) connections.

To load-balance based on the MPLS label information, configure the `family mpls` statement:

```
[edit forwarding-options hash-key]
family mpls {
    all-labels;
    bottom-label-1;
```

```
    bottom-label-2;
    bottom-label-3;
    label-1;
    label-2;
    label-3;
    no-labels;
    no-label-1-exp;
    payload {
        ether-pseudowire;
        ip {
            disable;
            layer-3-only;
            port-data {
                destination-lsb;
                destination-msb;
                source-lsb;
                source-msb;
            }
        }
    }
}
```

You can include this statement at the following hierarchy levels:

- `[edit forwarding-options hash-key]`

Table 8 on page 236 provides detailed information about all of the possible MPLS LSP load-balancing options.

Table 8: MPLS LSP Load Balancing Options

Statement	Supported Platforms	MPLS LSP Load Balancing Options
`all-labels`	MX Series and PTX Series	Prior to Junos OS Release 19.1R1, up to eight MPLS labels were included in the hash key to identify the uniqueness of a flow in the Packet Forwarding Engine. On PTX Series routers, this value is set by default. Starting in Junos OS Release 19.1R1, for MX Series routers with MPC and MIC interfaces, up to sixteen incoming MPLS labels are included in the hash key.

Table 8: MPLS LSP Load Balancing Options *(Continued)*

Statement	Supported Platforms	MPLS LSP Load Balancing Options
`bottom-label-1`	MX Series with DPC (I-Chip). Not supported on M10i, M7i, and M120.	Uses the bottom-most label for calculating the hash key, for example if the top labels do not provide sufficient variable for the required level of entropy.
`bottom-label-2`	MX Series with DPC (I-Chip). Not supported on M10i, M7i, and M120.	Uses the second label from the bottom for calculating the hash key, for example if the top labels do not provide sufficient variable for the required level of entropy.
`bottom-label-3`	MX Series with DPC (I-Chip). Not supported on M10i, M7i, and M120.	Uses the third label from the bottom for calculating the hash key, for example if the top labels do not provide sufficient variable for the required level of entropy.
`label-1`	M Series, MX Series, T Series	Include the first label in the hash key. Use this option for single label packets.
`label-2`	M Series, MX Series, T Series	Include the second label in the hash key. You must also configure the `label-1` option. The entire first label and the first 16 bits of the second label are used in the hash key.
`label-3`	M Series, MX Series, T Series	Include the third label in the hash key. You must also configure the `label-1` option and the `label-2` option.
`no-labels`	All	Excludes MPLS labels from the hash key.

Table 8: MPLS LSP Load Balancing Options *(Continued)*

Statement	Supported Platforms	MPLS LSP Load Balancing Options
`no-label-1-exp`	M Series, MX Series, T Series	Excludes the EXP bit of the top label from the hash key. You must also configure the `label-1` option. For Layer 2 VPNs, the router could encounter a packet reordering problem. When a burst of traffic pushes the customer traffic bandwidth to exceed its limits, the traffic might be affected in mid flow. Packets might be reordered as a result. By excluding the EXP bit from the hash key, you can avoid this reordering problem.
`payload`	All	Allows you to configure which parts of the IP packet payload to include in the hash key. For the PTX Series Packet Transport Router, this value is set by default.
`disable`	PTX Series	Exclude IP payload from the hash key.
`ether-pseudowire`	M120, M320, MX Series, T Series	Load-balance IPv4 traffic over Layer 2 Ethernet pseudowires.
`ip`	All	Include the IPv4 or IPv6 address in the hash key. You must also configure either `label-1` or `no-labels`.
`layer-3-only`	All	Include only the Layer 3 IP information in the hash key. Excludes all of the `port-data` bytes from the hash key.
`port-data`	M Series, MX Series, T Series	Include the source and destination port field information. By default, the most significant byte and least significant byte of the source and destination port fields are used in the hash key. To select specific bytes to use in the hash key, include one or more of the `source-msb`, `source-lsb`, `destination-msb`, and `destination-lsb` options at the `[edit forwarding-options hash-key family mpls payload ip port-data]` hierarchy level. To prevent all four bytes from being hashed, include the `layer-3-only` statement at the `[edit forwarding-options hash-key family mpls payload ip]` hierarchy level.

Table 8: MPLS LSP Load Balancing Options *(Continued)*

Statement	Supported Platforms	MPLS LSP Load Balancing Options
`destination-lsb`	M Series, MX Series, T Series	Include the least significant byte of the destination port in the hash key. Can be combined with any of the other `port-data` options.
`destination-msb`	M Series, MX Series, T Series	Include the most significant byte of the destination port in the hash key. Can be combined with any of the other `port-data` options.
`source-lsb`	M Series, MX Series, T Series	Include the least significant byte of the source port in the hash key. Can be combined with any of the other `port-data` options.
`source-msb`	M Series, MX Series, T Series	Include the most significant byte of the source port in the hash key. Can be combined with any of the other `port-data` options.

The following examples illustrate ways in which you can configure MPLS LSP load balancing:

- To include the IP address as well as the first label in the hash key:
  - For M Series, MX Series, and T Series routers, configure the `label-1` statement and the `ip` option for the `payload` statement at the `[edit forwarding-options hash-key family mpls]` hierarchy level:

```
[edit forwarding-options hash-key family mpls]
label-1;
payload {
    ip;
}
```

  - For PTX Series Packet Transport Routers, the `all-labels` and `ip payload` options are configured by default, so no configuration is necessary.

- (M320 and T Series routers only) To include the IP address as well as both the first and second labels in the hash key, configure the `label-1` and `label-2` options and the `ip` option for the `payload` statement at the `[edit forwarding-options hash-key family mpls]` hierarchy level:

```
[edit forwarding-options hash-key family mpls]
label-1;
label-2;
payload {
    ip;
}
```

NOTE: You can include this combination of statements on M320 and T Series routers only. If you include them on an M Series Multiservice Edge Router, only the first MPLS label and the IP payload are used in the hash key.

- For T Series routers, ensure proper load balancing by including the `label-1`, `label-2`, and `label-3` options at the `[edit forwarding-options hash-key family mpls]` hierarchy level:

```
[edit forwarding-options hash-key family mpls]
label-1;
label-2;
label-3;
```

- (M Series, MX Series, and T Series routers only) For Layer 2 VPNs, the router could encounter a packet reordering problem. When a burst of traffic pushes the customer traffic bandwidth to exceed its limits, the traffic might be affected in mid flow. Packets might be reordered as a result. By excluding the EXP bit from the hash key, you can avoid this reordering problem. To exclude the EXP bit of the first label from the hash calculations, include the `no-label-1-exp` statement at the `[edit forwarding-options hash-key family mpls]` hierarchy level:

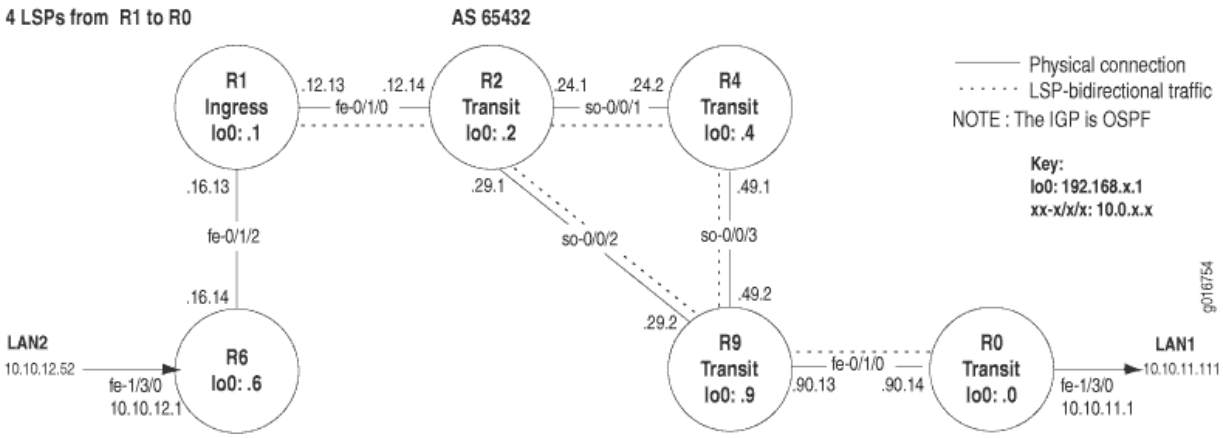
```
[edit forwarding-options hash-key family mpls]
label-1;
no-label-1-exp;
payload {
    ip;
}
```

Example: Load-Balanced MPLS Network

When you configure several RSVP LSPs to the same egress router, the LSP with the lowest metric is selected and carries all traffic. If all of the LSPs have the same metric, one of the LSPs is selected at random and all traffic is forwarded over it. To distribute traffic equally across all LSPs, you can configure load balancing on the ingress or transit routers, depending on the type of load balancing configured.

Figure 11 on page 241 illustrates an MPLS network with four LSPs configured to the same egress router (**R0**). Load balancing is configured on ingress router **R1**. The example network uses Open Shortest Path First (OSPF) as the interior gateway protocol (IGP) with OSPF area **0.0.0.0**. An IGP is required for the Constrained Shortest Path First (CSPF) LSP, which is the default for the Junos OS. In addition, the example network uses a policy to create BGP traffic.

Figure 11: Load-Balancing Network Topology



The network shown in Figure 11 on page 241 consists of the following components:

- A full-mesh interior BGP (IBGP) topology, using AS 65432
- MPLS and RSVP enabled on all routers
- A send-statics policy on routers **R1** and **R0** that allows a new route to be advertised into the network
- Four unidirectional LSPs between **R1** and **R0**, and one reverse direction LSP between **R0** and **R1**, which allows for bidirectional traffic
- Load balancing configured on ingress router **R1**

The network shown in Figure 11 on page 241 is a BGP full-mesh network. Since route reflectors and confederations are not used to propagate BGP learned routes, each router must have a BGP session with every other router running BGP.

Router Configurations for the Load-Balanced MPLS Network

IN THIS SECTION

- Purpose | 242
- Action | 242
- Sample Output 1 | 242
- Sample Output 2 | 244
- Sample Output 3 | 247
- Sample Output 4 | 250
- Sample Output 5 | 252
- Sample Output 6 | 254
- Meaning | 256

Purpose

The configurations in this topic are for the six load-balanced routers in the example network illustrated in "Load-Balancing Network Topology" on page 241.

Action

To display the configuration of a router, use the following Junos OS CLI operational mode command:

```
user@host> show configuration | no-more
```

Sample Output 1

The following configuration output is for edge router **R6**.

```
user@R6> show configuration | no-more
[...Output truncated...]
interfaces {
    fe-0/1/2 {
        unit 0 {
            family inet {
```

```
                address 10.0.16.14/30;
            }
             family mpls;  #MPLS enabled on relevant interfaces
        }
    }
    fe-1/3/0 {
        unit 0 {
            family inet {
                address 10.10.12.1/24;
            }
        }
    }
    fxp0 {
        unit 0 {
            family inet {
                address 192.168.70.148/21;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 192.168.6.1/32;
            }
        }
    }
}
routing-options {
    static {
[...Output truncated...]
     router-id 192.168.6.1; #Manually configured RID
    autonomous-system 65432; #Full mesh IBGP
    }
}
protocols {
    rsvp {
        interface fe-0/1/2.0;
        interface fxp0.0 {
            disable;
        }
    }
    mpls {
        interface fe-0/1/2.0;
```

```
        interface fxp0.0 {
            disable;
        }
    }
    bgp {
        group internal {
            type internal;
            local-address 192.168.6.1;
            neighbor 192.168.1.1;
            neighbor 192.168.2.1;
            neighbor 192.168.4.1;
            neighbor 192.168.9.1;
            neighbor 192.168.0.1;
        }
    }
     ospf  { #IGP enabled
        traffic-engineering;
        area 0.0.0.0 {
            interface fe-0/1/2.0;
            interface fe-1/3/0.0;
            interface  lo0.0 {
                passive; #Ensures protocols do not run over this interface
            }
        }
    }
}
```

Sample Output 2

The following configuration output is for ingress router **R1**.

```
user@R1> show configuration | no-more
[...Output truncated...]
interfaces {
    fe-0/1/0 {
        unit 0 {
            family inet {
                address 10.0.12.13/30;
            }
             family mpls; #MPLS enabled on relevant interfaces
        }
```

```
    }
    fe-0/1/2 {
        unit 0 {
            family inet {
                address 10.0.16.13/30;
            }
            family mpls;
        }
    }
    fxp0 {
        unit 0 {
            family inet {
                address 192.168.70.143/21;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 192.168.1.1/32;
            }
        }
    }
}
routing-options {
    static {
        [...Output truncated...]
     route 100.100.1.0/24 reject; #Static route for send-statics policy
}
     router-id 192.168.1.1; #Manually configured RID
     autonomous-system 65432; #Full mesh IBGP
    forwarding-table {
         export lbpp; #Routes exported to forwarding table
    }
}
protocols {
    rsvp {
        interface fe-0/1/0.0;
        interface fe-0/1/2.0;
        interface fxp0.0 {
            disable;
        }
    }
```

```
    mpls {
         label-switched-path lsp 1 { #First LSP
             to 192.168.0.1;  # Destination of the LSP
             install 10.0.90.14/32 active; # The prefix is installed in the
            primary via-r4;               # inet.0 routing table
        }
         label-switched-path lsp2  {
            to 192.168.0.1;
            install 10.0.90.14/32 active;
            primary via-r2;
        }
         label-switched-path lsp3  {
            to 192.168.0.1;
            install 10.0.90.14/32 active;
            primary via-r2;
        }
         label-switched-path lsp4  {
            to 192.168.0.1;
            install 10.0.90.14/32 active;
            primary via-r4;
        }
         path via-r2 { #Primary path to spread traffic across interfaces
            10.0.29.2 loose;
        }
         path via-r4  {
            10.0.24.2 loose;
        }
        interface fe-0/1/0.0;
        interface fe-0/1/2.0;
        interface fxp0.0 {
            disable;
        }
    }
    bgp {
         export send-statics; #Allows advertising of a new route
        group internal {
            type internal;
            local-address 192.168.1.1;
            neighbor 192.168.2.1;
            neighbor 192.168.4.1;
            neighbor 192.168.9.1;
            neighbor 192.168.6.1;
            neighbor 192.168.0.1;
```

```
        }
    }
     ospf  { #IGP enabled
        traffic-engineering;
        area 0.0.0.0 {
            interface fe-0/1/0.0;
            interface fe-0/1/2.0;
             interface  lo0.0 {
                 passive; #Ensures protocols do not run over this interface
            }
        }
    }
}
policy-options { #Load balancing policy
    policy-statement lbpp {
        then {
            load-balance per-packet;
        }
    }
    policy-statement send-statics {  #Static route policy
        term statics {
            from {
                route-filter 100.100.1.0/24 exact;
            }
            then accept;
        }
    }
}
```

Sample Output 3

The following configuration output is for transit router **R2**.

```
user@R2> show configuration | no-more
[...Output truncated...]
interfaces {
    so-0/0/1 {
        unit 0 {
            family inet {
                address 10.0.24.1/30;
            }
```

```
             family mpls; #MPLS enabled on relevant interfaces
        }
    }
    so-0/0/2 {
        unit 0 {
            family inet {
                address 10.0.29.1/30;
            }
            family mpls;
        }
    }
    fe-0/1/0 {
        unit 0 {
            family inet {
                address 10.0.12.14/30;
            }
            family mpls;
        }
    }
    fxp0 {
        unit 0 {
            family inet {
                address 192.168.70.144/21;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 192.168.2.1/32;
            }
        }
    }
}
routing-options {
    static {
        [...Output truncated...]
     router-id 192.168.2.1; #Manually configured RID
     autonomous-system 65432; #Full mesh IBGP
    }
}
protocols {
    rsvp {
```

```
        interface so-0/0/1.0;
        interface fe-0/1/0.0;
        interface so-0/0/2.0;
        interface fxp0.0 {
            disable;
        }
    }
    mpls {
        interface fe-0/1/0.0;
        interface so-0/0/1.0;
        interface so-0/0/2.0;
        interface fxp0.0 {
            disable;
        }
    }
    bgp {
        group internal {
            type internal;
            local-address 192.168.2.1;
            neighbor 192.168.1.1;
            neighbor 192.168.4.1;
            neighbor 192.168.9.1;
            neighbor 192.168.6.1;
            neighbor 192.168.0.1;
        }
    }
     ospf  { #IGP enabled
        traffic-engineering;
        area 0.0.0.0 {
            interface fe-0/1/0.0;
            interface so-0/0/1.0;
            interface so-0/0/2.0;
            interface  lo0.0  {
                 passive; #Ensures protocols do not run over this interface
            }
        }
    }
}
```

Sample Output 4

The following configuration output is for transit router **R4**.

```
user@R4> show configuration | no-more
[...Output truncated...]
interfaces {
    so-0/0/1 {
        unit 0 {
            family inet {
                address 10.0.24.2/30;
            }
             family mpls;  # MPLS enabled on relevant interfaces
        }
    }
    so-0/0/3 {
        unit 0 {
            family inet {
                address 10.0.49.1/30;
            }
            family mpls;
        }
    }
    fxp0 {
        unit 0 {
            family inet {
                address 192.168.70.146/21;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 192.168.4.1/32;
            }
        }
    }
}
routing-options {
    static {
        [...Output truncated...]
     router-id 192.168.4.1; #Manually configured RID
```

```
     autonomous-system 65432; #Full mesh IBGP
}
protocols {
    rsvp {
        interface so-0/0/1.0;
        interface so-0/0/3.0;
        interface fxp0.0 {
            disable;
        }
    }
    mpls {
        interface so-0/0/1.0;
        interface so-0/0/3.0;
        interface fxp0.0 {
            disable;
        }
    }
    bgp {
        group internal {
            type internal;
            local-address 192.168.4.1;
            neighbor 192.168.1.1;
            neighbor 192.168.2.1;
            neighbor 192.168.9.1;
            neighbor 192.168.6.1;
            neighbor 192.168.0.1;
        }
    }
     ospf  { #IGP enabled
        traffic-engineering;
        area 0.0.0.0 {
            interface so-0/0/1.0;
            interface so-0/0/3.0;
            interface  lo0.0  {
                 passive; #Ensures protocols do not run over this interface
            }
        }
    }
}
```

Sample Output 5

The following configuration output is for transit router **R9**.

```
user@R9> show configuration | no-more
[...Output truncated...]
interfaces {
    so-0/0/2 {
        unit 0 {
            family inet {
                address 10.0.29.2/30;
            }
             family mpls; #MPLS enabled on relevant interfaces
        }
    }
    so-0/0/3 {
        unit 0 {
            family inet {
                address 10.0.49.2/30;
            }
            family mpls;
        }
    }
    fe-0/1/0 {
        unit 0 {
            family inet {
                address 10.0.90.13/30;
            }
            family mpls;
        }
    }
    fxp0 {
        unit 0 {
            family inet {
                address 192.168.69.206/21;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 192.168.9.1/32;
```

```
            }
        }
    }
}
routing-options {
    static {
        [...Output truncated...]
     router-id 192.168.9. 1; #Manually configured RID
     autonomous-system 65432; #Full mesh IBGP
}
protocols {
    rsvp {
        interface so-0/0/2.0;
        interface so-0/0/3.0;
        interface fe-0/1/0.0;
        interface fxp0.0 {
            disable;
        }
    }
    mpls {
        interface so-0/0/2.0;
        interface so-0/0/3.0;
        interface fe-0/1/0.0;
        interface fxp0.0 {
            disable;
        }
    }
    bgp {
        group internal {
            type internal;
            local-address 192.168.9.1;
            neighbor 192.168.1.1;
            neighbor 192.168.2.1;
            neighbor 192.168.4.1;
            neighbor 192.168.0.1;
            neighbor 192.168.6.1;
        }
    }
     ospf  { #IGP enabled
        traffic-engineering;
        area 0.0.0.0 {
            interface so-0/0/2.0;
            interface so-0/0/3.0;
```

```
            interface fe-0/1/0.0;
            interface  lo0.0 {
                 passive; #Ensures protocols do not run over this interface
            }
        }
    }
}
```

Sample Output 6

The following configuration output is for egress router **R0**.

```
user@R0> show configuration | no-more
[...Output truncated...]
interfaces {
    fe-0/1/0 {
        unit 0 {
            family inet {
                address 10.0.90.14/30;
            }
             family mpls; #MPLS enabled on relevant interfaces
        }
    }
    fe-1/3/0 {
        unit 0 {
            family inet {
                address 10.10.11.1/24;
        }
    }
    fxp0 {
        unit 0 {
            family inet {
                address 192.168.69.207/21;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 192.168.0.1/32;
            }
```

```
        }
    }
}
routing-options {
    static {
        [...Output truncated...]
         route 100.100.10.0/24 reject; #Static route for send-statics policy
    }
     router-id 192.168.0.1; #Manually configured RID
     autonomous-system 65432; #Full mesh IBGP
}
protocols {
    rsvp {
        interface fe-0/1/0.0;
        interface fe-1/3/0.0;
        interface fxp0.0 {
            disable;
        }
    }
    mpls {
         label-switched-path r0-r6 {
            to 192.168.6.1;
        }
        interface fe-0/1/0.0;
        interface fe-1/3/0.0;
        interface fxp0.0 {
            disable;
        }
    }
    bgp {
        group internal {
            type internal;
            local-address 192.168.0.1;
             export send-statics; #Allows advertising of a new route
            neighbor 192.168.9.1;
            neighbor 192.168.6.1;
            neighbor 192.168.1.1;
            neighbor 192.168.2.1;
            neighbor 192.168.4.1;
        }
    }
     ospf  { #IGP enabled
        traffic-engineering;
```

```
        area 0.0.0.0 {
            interface fe-0/1/0.0;
            interface fe-1/3/0.0;
            interface  lo0.0 {
                 passive; #Ensures protocols do not run over this interface
            }
        }
    }
}
policy-options {
     policy-statement send-statics {
        term statics {
            from {
                route-filter 100.100.10.0/24 exact;
            }
            then accept;
        }
    }
}
```

Meaning

Sample Outputs 1 through 6 show the base interfaces, routing options, protocols, and policy options configurations for all six routers in the example network illustrated in "Example: Load-Balanced MPLS Network" on page 241.

All routers in the network have MPLS, RSVP, and BGP enabled. OSPF is configured as the IGP, and relevant interfaces have basic IP information and MPLS support.

In addition, all routers have the router ID (RID) configured manually at the `[edit routing-options]` hierarchy level to avoid duplicate RID problems. The `passive` statement is included in the OSPF configuration to ensure that protocols are not run over the loopback (**lo0**) interface and that the loopback (**lo0**) interface is advertised correctly throughout the network.

Sample Outputs 1, 3, 4, and 5 for **R6**, **R2**, **R4**, and **R9** show the base configuration for transit label-switched routers. The base configuration includes all interfaces enabled for MPLS, the RID manually configured, and the relevant protocols (RSVP, MPLS, BGP, and OSPF).

Sample Output 2 from ingress router **R1** shows the base configuration plus four LSPs (**lsp1** through **lsp4)** configured to **R0**. The four LSPs are configured with different primary paths that specify a loose hop through **R4** for **lsp1** and **lsp4**, and through **R2** for **lsp2** and **lsp3**.

To create traffic, **R1** has a static route (**100.100.1.0/24**) configured at the `[edit routing-options static route]` hierarchy level. The prefix is included in the send-statics policy at the `[edit policy-options send statics]` hierarchy level so the routes can become BGP routes.

In addition, on the ingress router **R1**, load balancing is configured using the **per-packet** option, and the policy is exported at the `[edit routing-options forwarding-table]` hierarchy level.

Sample Output 6 from egress router **R0** shows one LSP (**r0-r6**) to **R6** used to create bidirectional traffic. OSPF requires bidirectional LSP reachability before it will advertise the LSP into the IGP. Although the LSP is advertised into the IGP, no hello messages or routing updates occur over the LSP—only user traffic is sent over the LSP. The router uses its local copy of the IGP database to verify bidirectional reachability.

In addition, **R0** has a static route (**100.100.10.0/24**) configured at the `[edit routing-options static route]` hierarchy level. The prefix is included in the send-statics policy at the `[edit policy-options send statics]` hierarchy level so the routes can become BGP routes.

Configuring Load Balancing Based on MPLS Labels on ACX Series Routers

Table 9 on page 259 provides detailed information about all of the possible MPLS LSP load-balancing options.

ACX Series routers can load-balance on a per-packet basis in MPLS. Load balancing can be performed on information in both the IP header and on up to three MPLS labels, providing a more uniform distribution of MPLS traffic to next hops. This feature is enabled on supported platforms by default and requires no configuration.

Load balancing is used to evenly distribute traffic when there is a single next hop over an aggregated interface or a LAG bundle. Load balancing using MPLS labels is supported only for LAG interfaces and not for equal-cost multipath (ECMP) links.

By default, when load balancing is used to help distribute traffic, Junos OS employs a hash algorithm to select a next-hop address to install into the forwarding table. Whenever the set of next hops for a destination changes in any way, the next-hop address is reselected by means of the hash algorithm. You can configure how the hash algorithm is used to load-balance traffic across interfaces in an aggregated Ethernet (ae) interface.

An LSP tends to load-balance its placement by randomly selecting one of the interfaces in an `ae-` interface bundle and using it exclusively. The random selection is made independently at each transit router, which compares Interior Gateway Protocol (IGP) metrics alone. No consideration is given to bandwidth or congestion levels.

To load-balance based on the MPLS label information, configure the `family mpls` statement:

```
[edit forwarding-options hash-key]
family mpls {
```

```
    all-labels;
    label-1;
    label-2;
    label-3;
    no-labels;
    payload {
        ether-pseudowire;
        ip {
            layer-3-only;
            port-data {
                destination-lsb;
                destination-msb;
                source-lsb;
                source-msb;
            }
        }
    }
}
```

You can include this statement at the `[edit forwarding-options hash-key]` hierarchy level.

NOTE: When you configure payload ip (`user@host#` **set forwarding-options hash-key family mpls payload ip**), configuring `layer-3-only` and `port-data` is mandatory.

Load balancing functionality, without proper hash-keys configuration, may result in an unpredictable behavior.

The ACX7000 series of devices do not support `port-data`.

For Layer 2 VPN/pseudowire tunnel termination, upto two labels are used for hashing and payload MAC destination and source addresses can be optionally selected. These controls can be used to support ether-pseudowire knob in family mpls under hash-key configuration shown above. However, since ACX2000 and ACX4000 also support TDM pseudowires, the ether-pseudowire knobs needs to be used only when TDM pseudowires are not being used.

For Layer 3 VPN tunnel termination, upto two labels are used for hasing and payload IP source and destination addresses and Layer 4 source and destination ports can be optionally selected. These controls can be used for supporting ip port-data knobs in family mpls under hash-key configuration shown above. However, since Layer 4 port MSB and LSB cannot be individually selected, one of destination-lsb or destination-msb knobs or one of source-lsb or source-msb knobs would select Layer 4 destination or source ports, respectively.

For LSR case, upto three labels are used for hashing. If a BOS is seen when parsing the first three labels, BCM examines the first nibble of payload - if the nibble is 4, the payload is treated as IPv4 and if the

first nibble is 6, the payload is treated as IPv6 and in such cases payload source and destination IP addresses can be speculatively used for hashing. These controls can be used for supporting ip port-data knobs in family mpls under hash-key configuration. However, Layer 4 ports cannot be used for hashing in LSR case, and only layer-3-only knob is applicable. BCM does not claim support for hashing on fields beyond the three MPLS labels. Load Balancing for a single pseudowire session does not take place in case of LSR as all the traffic specific to that session will carry the same set of MPLS labels.

Load balancing on LSR AE interfaces can be achieved for a higher number of MPLS sessions, that is minimum of 10 sessions. This is applicable for CCC/VPLS/L3VPN. In case of Layer 3 VPN, the traffic may not be equally distributed across the member links as the layer 3 addresses also get accounted for (along with the labels) for the hash input function.

For LER scenarios, in case of ACX5048 and ACX5096, hashing based on Layer 3 and Layer 4 fields is possible by configuring the payload option under the “family mpls” hierarchy. Hashing on the LER is not be based on Labels. For Layer 3 service, it is mandatory to mention the payload as “layer-3-only” and specify “port-data” in case of Layer 4 service. You can also mention the label count while configuring hash-keys on LER routers.

NOTE: LER and LSR load balancing behavior is applicable for CCC/VPLS/Layer 3 VPN and other IP MPLS scenarios.

This feature applies to aggregated Ethernet and aggregated SONET/SDH interfaces. In addition, you can configure load balancing for IPv4 traffic over Layer 2 Ethernet pseudowires. You can also configure load balancing for Ethernet pseudowires based on IP information. The option to include IP information in the hash key provides support for Ethernet circuit cross-connect (CCC) connections.

Table 9: MPLS LSP Load Balancing Options

Statement	MPLS LSP Load Balancing Options
`label-1`	Include the first label in the hash key. Use this option for single label packets.
`label-2`	Include the second label in the hash key. You must also configure the `label-1` option. The entire first label and the first 16 bits of the second label are used in the hash key.
`label-3`	Include the third label in the hash key. You must also configure the `label-1` option and the `label-2` option.
`no-labels`	Excludes MPLS labels from the hash key.

Table 9: MPLS LSP Load Balancing Options *(Continued)*

Statement	MPLS LSP Load Balancing Options
payload	Allows you to configure which parts of the IP packet payload to include in the hash key. For the PTX Series Packet Transport Switch, this value is set by default.
disable	Exclude IP payload from the hash key.
ether-pseudowire	Load-balance IPv4 traffic over Layer 2 Ethernet pseudowires.
ip	Include the IPv4 or IPv6 address in the hash key. You must also configure either label-l or no-labels.
layer-3-only	Include only the Layer 3 IP information in the hash key. Excludes all of the port-data bytes from the hash key.
port-data	Include the source and destination port field information. By default, the most significant byte and least significant byte of the source and destination port fields are used in the hash key. To select specific bytes to use in the hash key, include one or more of the source-msb, source-lsb, destination-msb, and destination-lsb options at the [edit forwarding-options hash-key family mpls payload ip port-data] hierarchy level. To prevent all four bytes from being hashed, include the layer-3-only statement at the [edit forwarding-options hash-key family mpls payload ip] hierarchy level.
destination-lsb	Include the least significant byte of the destination port in the hash key. Can be combined with any of the other port-data options.
destination-msb	Include the most significant byte of the destination port in the hash key. Can be combined with any of the other port-data options.
source-lsb	Include the least significant byte of the source port in the hash key. Can be combined with any of the other port-data options.
source-msb	Include the most significant byte of the source port in the hash key. Can be combined with any of the other port-data options.

To include the IP address as well as the first label in the hash key, configure the label-1 statement and the ip option for the payload statement at the [edit forwarding-options hash-key family mpls] hierarchy level:

```
[edit forwarding-options hash-key family mpls]
label-1;
payload {
    ip;
}
```

To include the IP address as well as both the first and second labels in the hash key, configure the label-1 and label-2 options and the ip option for the payload statement at the [edit forwarding-options hash-key family mpls] hierarchy level:

```
[edit forwarding-options hash-key family mpls]
label-1;
label-2;
payload {
    ip;
}
```

Ensure proper load balancing by including the label-1, label-2, and label-3 options at the [edit forwarding-options hash-key family mpls] hierarchy level:

```
[edit forwarding-options hash-key family mpls]
label-1;
label-2;
label-3;
```

MPLS Encapsulated Payload Load-balancing Overview

Routers can load-balance on a per-packet basis in MPLS. Load balancing can be performed on the information in both the IP header and on up to three MPLS labels, providing a more uniform distribution of MPLS traffic to next hops.

Load balancing is used to evenly distribute traffic when the following conditions apply:

- There are multiple equal-cost next hops over different interfaces to the same destination.
- There is a single next hop over an aggregated interface.

By default, when load balancing is used to help distribute traffic, a hash algorithm is used to select a next-hop address to install into the forwarding table. Whenever the set of next hops for a destination changes in any way, the next-hop address is reselected by means of the hash algorithm.

In case of multiple transport layer networks such as Ethernet over MPLS or Ethernet pseudowire, the hash algorithm needs to look beyond the outer header of the payload and into the inner headers to generate an even distribution. To determine the inner encapsulation, the PFE relies on the presence of certain codes or numbers at fixed payload offets; for example the presence of payload type 0X800 or the presence of protocol number 4 for an IPv4 packet. In Junos OS, you can configure `zero-control-word` option to indicate the start of an Ethernet frame in an MPLS ether-pseudowire payload. On seeing this control word, which is four bytes having a numerical value of all zeros, the hash generator assumes the start of an Ethernet frame at the end of the control word in an MPLS ether-pseudowire packet.

NOTE: For DPC I-chip-based cards, configure the `zero-control-word` option at the `[edit forwarding-options hash-key family mpls ether-pseudowire]` hierarchy level; and for MPC cards, configure the `zero-control-word` option at the `[edit forwarding-options enhanced-hash-key family mpls ether-pseudowire]` hierarchy level.

Configuring MPLS Encapsulated Payload for Load Balancing

By default, when load balancing is used to help distribute traffic, a hash algorithm is used to select a next-hop address to install into the forwarding table. Whenever the set of next hops for a destination changes in any way, the next-hop address is reselected by means of the hash algorithm. Configure the `zero-control-word` option to indicate the start of an Ethernet frame in an MPLS ether-pseudowire payload. On seeing this control word, four bytes having a numerical value of all zeros, the hash generator assumes the start of the Ethernet frame at the end of the control word in an MPLS ether-pseudowire packet.

Before you begin to configure MPLS encapsulated payload for load balancing, configure routing and signaling protocols.

To configure MPLS encapsulated payload for load balancing:

Configure the `zero-control-word` option to indicate the start of an Ethernet frame in an MPLS ether-pseudowire payload.

- For DPC I-chip-based cards, configure the `zero-control-word` option at the `[edit forwarding-options hash-key family mpls ether-pseudowire]` hierarchy level.

```
[edit forwarding-options hash-key family mpls ether-pseudowire]
user@host# set zero-control-word
```

- For MPC cards, configure the `zero-control-word` option at the `[edit forwarding-options enhanced-hash-key family mpls ether-pseudowire]` hierarchy level.

```
[edit forwarding-options enhanced-hash-key family mpls ether-pseudowire]
user@host# set zero-control-word
```

Policy-Based Multipath Routes Overview

IN THIS SECTION

- Understanding Policy-Based Multipath Routes | **263**
- Benefits of Policy-Based Multipath Routes | **264**
- Policy-Based Multipath Routes for Route Resolution | **264**
- Sample Route Resolution Using Policy-Based Multipath Routes | **265**
- Enhancement to Class-of-Service (CoS) Forwarding-Policy | **267**
- Enhancements to Policy Match Protocol | **268**
- Impact of Configuring Policy-Based Multipath Route on Network Performance | **269**

Segment routing networks can have multiple transport protocols in the core. You can combine segment routing SR-TE LDP or RSVP routes and SR-TE IP routes and install a multipath route in the routing information base (also known as routing table). You can then steer selective service traffic using the mutlipath route through policy configuration.

Understanding Policy-Based Multipath Routes

There are different transport protocols in a network, such as IGP, labelled IGP, RSVP, LDP, and segment routing traffic-engineering (SR-TE) protocols, that are used to resolve service traffic. However, you could not use a combination of the transport protocols to resolve the service traffic. With the introduction of the policy-based multipath feature, you can combine segment routing traffic-engineered (SR-TE) LDP or RSVP routes and SR-TE IP routes to create a multipath route that is installed in the routing information base. You can resolve BGP service routes over the mutlipath route through policy configuration and steer traffic differently for different prefixes.

A multipath route has combined next hops of route entries that are used for load balancing. All the supporting routes of the multipath route entry must be in same routing information base. When the supporting routes are under different routing information base, you can use the `rib-group` configuration statement to add route entries to a particular routing information base.

You can configure a multipath route using a policy to select the list of routes whose next hops is to be combined together. When you include the `policy-multipath` statement along with the `policy` statement at the `[edit routing-options rib routing-table-name]` hierarchy level, a policy-based multipath route is created.

The policy-based multipath feature is supported for both IP and IPv6 protocols, and can be configured under the `[edit routing-instances]` hierarchy level.

For example:

```
[edit routing-options]
user@host# set rib inet.3 policy-multipath policy example-policy

[edit policy-options]
user@host# set policy-statement example-policy from example-conditions
user@host# set policy-options policy-statement example-policy then accept
```

The configured policy is applied to each route entry for a given prefix. The multipath route is created only when more than one route (including active route) passes the policy. Any action commands configured in the policy, such as apply, is evaluated using the active route. For non-active routes, the policy is applied to check if the routes can participate in the multipath route or not. Multipath routes inherit all attributes of the active route. These attributes can be modified using the multipath policy configuration.

Benefits of Policy-Based Multipath Routes

- Provides flexibility to combine core network protocols to steer selective traffic.
- Optimizes network performance with weighted equal-cost multipath using multipath routes.

Policy-Based Multipath Routes for Route Resolution

You can combine segment routing traffic-engineered (SR-TE) LDP or RSVP routes and SR-TE IP routes and install a multipath route in the routing information base. The policy-based multipath routes are not active entries in the routing information base. When a multipath route is generated by configuration of policy, it is used for resolving protocol next hops instead of active routes. A multipath route next hop is created by merging gateways of next hops of each constituent route.

Take the following into consideration when configuring policy-based multipath routes for route resolution:

- If the member route of a multipath route points to a next hop other than the router next hop or an indirect next hop with forwarding next hop to the router next hop, such next hops are ignored.

- If the constituent routes point to indirect next hop, then gateways from the forwarding-next hop are merged and the indirect next hop is ignored.
- If total number of gateways exceeds the `maximum-ecmp` supported on the device, then only the `maximum-ecmp` gateways are retained and all other gateways are ignored.
- Gateways with lower weights are given preference. When one of the member route has unilist of indirect next hops and each of the next hop is pointing to a forwarding next hop, there can be weight values both at the indirect next hop and at forwarding next hop. In such cases, weight value of gateways is updated to reflect the combined effect of weights at both levels.

Sample Route Resolution Using Policy-Based Multipath Routes

Taking as an example, let us assume there are segment routing traffic-engineered LSPs, label IS-IS routes, and LDP LSPs for a destination 10.1.1.1/32, as displayed in the output below:

```
10.1.1.1/32         *[SPRING-TE/8] 00:00:58, metric 1, metric2 30
                    > to 10.13.1.2 via ge-0/0/1.1, Push 33333, Push 801005, Push 801006(top)
                    [L-ISIS/14] 1w0d 00:15:57, metric 10
                    > to 10.12.1.1 via ge-0/0/0.1
                      to 10.22.1.1 via ge-0/0/0.2
                      to 10.23.1.1 via ge-0/0/0.3
                      to 10.24.1.1 via ge-0/0/0.4
                      to 10.25.1.1 via ge-0/0/0.5
                      to 10.13.1.2 via ge-0/0/1.1, Push 801001, Push 801005(top)
                    [LDP/19] 1w0d 00:09:27, metric 1
                    > to 10.12.1.1 via ge-0/0/0.1
                      to 10.22.1.1 via ge-0/0/0.2
                      to 10.23.1.1 via ge-0/0/0.3
                      to 10.24.1.1 via ge-0/0/0.4
                      to 10.25.1.1 via ge-0/0/0.5
                      to 10.13.1.2 via ge-0/0/1.1, Push 801001, Push 801005(top)
```

Here, segment routing LSP is the active route entry to the 10.1.1.1 destination, and by default, only this route is used to resolve any services resolving over 10.1.1.1.

When there is a requirement to use more than one protocols for resolving service routes, you can achieve this by configuring *policy-multipath* to combine the protocols. For instance, if segment routing and LDP paths are required for service resolution, you must configure `policy-multipath` combining the segment routing and LDP routes for prefix 10.1.1.1.

For example:

```
[edit policy-options]
user@host# set rib inet.3 policy-multipath policy example-policy
user@host# set policy-statement abc term 1 from protocol spring-te
user@host# set policy-statement abc term 1 from protocol ldp
user@host# set policy-statement abc term 1 from route-filter 10.1.1.1/32 exact
user@host# set policy-statement abc term 1 then accept
```

With this configuration, you create a policy-based multipath route for prefix 10.1.1.1/32 that uses constituent route entries of segment routing and LDP protocols.

You can view the multipath route using the `show route` command output, as follows:

```
10.1.1.1/32         *[SPRING-TE/8] 00:10:28, metric 1, metric2 30
                    > to 10.13.1.2 via ge-0/0/1.1, Push 33333, Push 801005, Push 801006(top)
                    [L-ISIS/14] 1w0d 00:25:27, metric 10
                    > to 10.12.1.1 via ge-0/0/0.1
                      to 10.22.1.1 via ge-0/0/0.2
                      to 10.23.1.1 via ge-0/0/0.3
                      to 10.24.1.1 via ge-0/0/0.4
                      to 10.25.1.1 via ge-0/0/0.5
                      to 10.13.1.2 via ge-0/0/1.1, Push 801001, Push 801005(top)
                    [LDP/19] 1w0d 00:18:57, metric 1
                    > to 10.12.1.1 via ge-0/0/0.1
                      to 10.22.1.1 via ge-0/0/0.2
                      to 10.23.1.1 via ge-0/0/0.3
                      to 10.24.1.1 via ge-0/0/0.4
                      to 10.25.1.1 via ge-0/0/0.5
                      to 10.13.1.2 via ge-0/0/1.1, Push 801001, Push 801005(top)
                    [Multipath/8] 00:03:13, metric 1, metric2 30
                    > to 10.12.1.1 via ge-0/0/0.1
                      to 10.22.1.1 via ge-0/0/0.2
                      to 10.23.1.1 via ge-0/0/0.3
                      to 10.24.1.1 via ge-0/0/0.4
                      to 10.25.1.1 via ge-0/0/0.5
                      to 10.13.1.2 via ge-0/0/1.1, Push 33333, Push 801005, Push 801006(top)
                      to 10.13.1.2 via ge-0/0/1.1, Push 801001, Push 801005(top)
```

You can see form the command output that the multipath route combines next hops of segment routing and LDP paths. The multipath route it is not active, and by default, the route preference and metric is the same as that of active route.

NOTE: You can use the following combinations for the poilcy-based multipath route: However we cannot create multipath of LDP/L-ISIS as active-route is not part of multipath.

- Segment routing traffic-engineered LSPs and LDP LSPs.
- Segment routing traffic-engineered LSPs, and label IS-IS paths.
- Segment routing traffic-engineered LSPs, LDP LSPs, and label IS-IS paths.

However, you cannot create multipath route of LDP and label IS-IS, as the active route is not part of the multipath route.

With the same configuration, assuming that there is a static route 1.2.3.4/32 configured with a protocol next hop of 10.1.1.1, this route is resolved using the multipath route over both segment routing traffic-engineered LSPs and LDP LSPs.

For example:

```
10.1.3.4/32        *[Static/5] 00:00:12, metric2 1
                     to 10.12.1.1 via ge-0/0/0.1
                   > to 10.22.1.1 via ge-0/0/0.2
                     to 10.23.1.1 via ge-0/0/0.3
                     to 10.24.1.1 via ge-0/0/0.4
                     to 10.25.1.1 via ge-0/0/0.5
                     to 10.13.1.2 via ge-0/0/1.1, Push 33333, Push 801005, Push 801006(top)
                     to 10.13.1.2 via ge-0/0/1.1, Push 801001, Push 801005(top)
```

Enhancement to Class-of-Service (CoS) Forwarding-Policy

For class-of-service-based forwarding, you must use the `forwarding-policy next-hop-map` configuration statement.

Prior to Junos OS Release 19.1R1, the match conditions supported under class-of-service-based forwarding included:

- **next-hop**—Match next hop based on outgoing interface or next hop address.
- **lsp-next-hop**—Match named LSPs using regular expression of LSP name.
- **non-lsp-next-hop**—Match all LSPs without an LSP name.

With the policy-based multipath route feature, you can also match all next hops without a label for certain prefixes. To do this, you must enable the `non-labelled-next-hop` option at the `[edit class-of-service forwarding-policy next-hop-map` *`map-name`* `forwarding-class` *`forwarding-class-name`* hierarchy level.

For example:

```
[edit]
class-of-service {
    forwarding-policy {
        next-hop-map abc {
            forwarding-class best-effort {
                                non-labelled-next-hop;
            }
        }
    }
}
```

Enhancements to Policy Match Protocol

Prior to Junos OS Release 19.1R1, when you used a policy to match protocol using the `from protocol` statement at the `[edit policy-options policy-statement` *`statement-name`*`]` hierarchy level, all protocol routes (labeled and unlabeled) were matched. With the policy-based multipath route feature, you can match labeled protocol routes specifically.

The options for matching labeled protocols) are:

- **l-isis**—Match labeled IS-IS routes. The `isis` option matches IS-IS routes, excluding label IS-IS routes.
- **l-ospf**—Match labled OSPF routes. The `ospf` option matches all OSPF routes, including OSPFv2, OSPFv3 and label OSPF.

For example:

```
[edit]
policy-options {
    policy-statement abc {
        from protocol [ l-ospf l-isis ];
    }
}
```

Impact of Configuring Policy-Based Multipath Route on Network Performance

When you configure policy-based multipath route, a change of route in the routing information base results in the evaluation of the policy to check if a multipath route needs to be created. Because this feature requires that member routes must be in the same routing information base, the `rib-group` statement is used to merge routes from different routing information base. Configuring the `rib-group` statement at the application level increases number of routes in the system.

When there are a number of routes in the routing information base, constant change of routes leads to reevaluation of the multipath policy. This could impact network performance. It is recommended to configure the policy-based multipath route feature only when required.

Understanding IP-Based Filtering and Selective Port Mirroring of MPLS Traffic

IN THIS SECTION

- IP-Based Filtering of MPLS Traffic | **269**
- Selective Port Mirroring of MPLS Traffic | **270**
- Sample Configurations | **271**

In an MPLS packet, the IP header comes immediately after the MPLS header. The IP-based filtering feature provides a deep inspection mechanism, where a maximum of upto eight MPLS labels of the inner payload can be inspected to enable filtering of MPLS traffic based on IP parameters. The filtered MPLS traffic can also be port mirrored to a monitoring device to offer network-based services in the core MPLS network.

IP-Based Filtering of MPLS Traffic

Prior to Junos OS Release 18.4R1, filtering based on IP parameters was not supported for MPLS family filter. With the introduction of the IP-based filtering feature, you can apply inbound and outbound filters for MPLS-tagged IPv4 and IPv6 packets based on IP parameters, such as source and destination addresses, Layer 4 protocol type, and source and destination ports.

The IP-based filtering feature enables you to filter MPLS packets at the ingress of an interface, where the filtering is done using match conditions on the inner payload of the MPLS packet. The selective MPLS traffic can then be port mirrored to a remote monitoring device using logical tunnels.

To support IP-based filtering, additional match conditions are added that allow MPLS packets to be deep inspected to parse the inner payload with Layer 3 and Layer 4 headers before the appropriate filters are applied.

NOTE: The IP-based filtering feature is supported only for MPLS-tagged IPv4 and IPv6 packets. In other words, the MPLS filters match IP parameters only when the IP payload comes immediately after the MPLS labels.

In other scenarios, where the MPLS payload includes pseudowires, protocols other than inet and inet6, or other encapsulations like Layer 2 VPN or VPLS, the IP-based filtering feature is not supported.

The following match conditions are added for the IP-based filtering of MPLS traffic:

- IPv4 source address
- IPv4 destination address
- IPv6 source address
- IPv6 destination address
- Protocol
- Source port
- Destination port
- Source IPv4 prefix list
- Destination IPv4 prefix list
- Source IPv6 prefix list
- Destination IPv6 prefix list

NOTE: The following match combinations are supported for the IP-based filtering of MPLS traffic:

- Source and destination address match conditions with IPv4 and IPv6 prefix lists.
- Source and destination port address and protocol types match conditions with IPv4 and IPv6 prefix lists.

Selective Port Mirroring of MPLS Traffic

Port mirroring is the capability of mirroring a packet to a configured destination, in addition to the normal processing and forwarding of the packets. Port mirroring is applied as an action for a firewall filter, which is applied at the ingress or egress of any interface. Similarly, the selective port mirroring

feature provides the capability to mirror MPLS traffic, which is filtered based on IP parameters, to a mirrored destination using logical tunnels.

To enable selective port mirroring, additional actions are configured at the `[edit firewall family mpls filter filter-nameterm term-name then]` hierarchy level, in addition to the existing `counter`, `accept`, and `discard` actions:

- `port-mirror`
- `port-mirror-instance`

Port Mirroring

The `port-mirror` action enables port mirroring globally on the device, which applies to all Packet Forwarding Engines (PFEs) and associated interfaces.

For MPLS family filter, the `port-mirror` action is enabled for global port mirroring.

Port Mirroring Instance

The `port-mirror-instance` action enables you to customize each instance with different properties for input sampling and port mirroring output destinations, instead of having to use a single system-wide configuration for port mirroring.

You can configure only two port mirroring instances per Flexible PIC Concentrator (FPC) by including the `instance port-mirror-instance-name` statement at the `[edit forwarding-options port-mirror]` hierarchy level. You can then associate individual port mirroring instances with an FPC, PIC, or (Forwarding Engine Board (FEB) depending on the device hardware.

For MPLS family filter, the `port-mirror-instance` action is enabled only for the port-mirroring instance.

NOTE: For both `port-mirror` and `port-mirror-instance` actions, the output interface must be enabled with Layer 2 family and not family MPLS (Layer 3) for the selective port mirroring feature to work.

Sample Configurations

IP-Based Filtering Configuration

```
[edit firewall family mpls filter mpls-filter]
term ipv4-term {
    from {
        ip-version {
            ipv4 {
```

```
                source-address {
                    10.10.10.10/24;
                }
                destination-address {
                    20.20.20.20/24;
                }
                protocol tcp {
                    source-port 100;
                    destination-port 200;
                }
                soure-prefix-list ipv4-source-users;
                destination-prefix-list ipv4-destination-users;
            }
        }
        exp 1;
        }
        then port-mirror;
        then accept;
    then count;
}
term ipv6-term {
    from {
        ip-version {
            ipv6 {
                source-address {
                    2000::1/128;
                }
                destination-address {
                    3000::1/128;
                }
                protocol tcp {
                    source-port 100;
                    destination-port 200;
                }
                source-prefix-list ipv6-source-users;
                destination-prefix-list ipv6-destination-users;
            }
        }
        exp 1;
    }
    then port-mirror-instance port-mirror-instance1;
    then accept;
```

```
    then count;
}
```

```
[edit policy-options]
prefix-list ipv4-source-users {
    172.16.1.16/28;
    172.16.2.16/28;
}
prefix-list ipv6-source-users {
    2001::1/128;
    3001::1/128;
}
```

```
[edit interfaces]
xe-0/0/1 {
    unit 0 {
        family inet {
            address 100.100.100.1/30;
        }
        family mpls {
            filter {
                input mpls-filter;
            }
        }
    }
}
```

Selective Port Mirroring Configuration

```
[edit forwarding-options]
port-mirroring {
    input {
        rate 2;
        run-length 4;
        maximum-packet-length 500;
    }
    family any {
        output {
```

```
            interface xe-2/0/2.0;
        }
    }
}
```

```
[edit forwarding-options]
port-mirroring {
    instance {
        port-mirror-instance1 {
            input {
                rate 3;
                run-length 5;
                maximum-packet-length 500;
            }
            family any {
                output {
                    interface xe-2/0/2.0;
                }
            }
        }
    }
}
```

NOTE: The output interface `xe-2/0/2.0` is configured for Layer 2 family and not family MPLS.

For both `port-mirror` and `port-mirror-instance` actions, the output interface must be enabled with Layer 2 family and not family MPLS (Layer 3) for the selective port mirroring feature to work.

Mirrored Destination Configuration

```
[edit interfaces]
xe-2/0/2 {
    vlan-tagging;
    encapsulation extended-vlan-bridge;
    unit 0 {
        vlan-id 600;
```

```
    }
}
```

```
[edit bridge-domains]
bd {
    domain-type bridge;
    interface xe-2/0/2.0;
}
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
19.1R1	Starting in Junos OS Release 19.1R1, for MX Series routers with MPC and MIC interfaces, up to sixteen incoming MPLS labels are included in the hash key.

RELATED DOCUMENTATION

Configuring Load Balancing for Ethernet Pseudowires | 1931

Shared Risk Link Groups for MPLS

IN THIS SECTION

- SRLG Overview | 276
- Example: Configuring SRLG | 276
- Example: Excluding SRLG Links Completely for the Secondary LSP | 289
- Example: Configuring SRLG with Link Protection | 298
- Example: Configuring SRLG with Link Protection with the exclude-srlg Option | 329

SRLG Overview

In MPLS traffic engineering, a Shared Risk Link Group (SRLG) is a set of links sharing a common resource, which affects all links in the set if the common resource fails. These links share the same risk of failure and are therefore considered to belong to the same SRLG. For example, links sharing a common fiber are said to be in the same SRLG because a fault with the fiber might cause all links in the group to fail.

An SRLG is represented by a 32-bit number unique within an IGP (OSPFv2 and IS-IS) domain. A link might belong to multiple SRLGs. The SRLG of a path in a label-switched path (LSP) is the set of SRLGs for all the links in the path. When computing the secondary path for an LSP, it is preferable to find a path such that the secondary and primary paths do not have any links in common in case the SRLGs for the primary and secondary paths are disjoint. This ensures that a single point of failure on a particular link does not bring down both the primary and secondary paths in the LSP.

When the SRLG is configured, the device uses the Constrained Shortest Path First (CSPF) algorithm and tries to keep the links used for the primary and secondary paths mutually exclusive. If the primary path goes down, the CSPF algorithm computes the secondary path by trying to avoid links that share any SRLG with the primary path. In addition, when computing the path for a bypass LSP, CSPF tries to avoid links that share any SRLG with the protected links.

When the SRLG is not configured, CSPF only takes into account the costs of the links when computing the secondary path.

Any change in link SRLG information triggers the IGP to send LSP updates for the new link SRLG information. CSPF recomputes the paths during the next round of reoptimization.

Junos OS Release 11.4 and later supports SRLG based on the following RFCs:

- RFC 4203, *OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)*.
- RFC 5307, *IS-IS Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)*.

NOTE: Currently, the “Fate Sharing” feature continues to be supported with the SRLG feature.

Example: Configuring SRLG

IN THIS SECTION

- Requirements | **277**
- Overview | **277**
- Configuration | **279**

- Verification | 286

This example shows how to configure Shared Risk Link Groups (SRLGs) on a device.

Requirements

This example uses the following hardware and software components:

- MX Series routers
- Junos OS Release 11.4 or later running on all the devices

Overview

IN THIS SECTION

- Topology | 278

Junos OS Release 11.4 and later support SRLG configuration in an IGP (OSPFv2 and IS-IS) domain. In this example, you configure SRLG and associate it with the MPLS interface on a device.

The device uses the SRLG cost parameter for the Constrained Shortest Path First (CSPF) algorithm and tries to keep the links used for the primary and secondary paths mutually exclusive by avoiding links that share any SRLG with the primary path.

To configure the SRLG, you first define the SRLG parameters at the `[edit routing-options srlg srlg-name]` hierarchy level and then associate the SRLG with an MPLS interface at the `[edit mpls interface interface-name]` hierarchy level.

The `srlg srlg-name` statement has the following options:

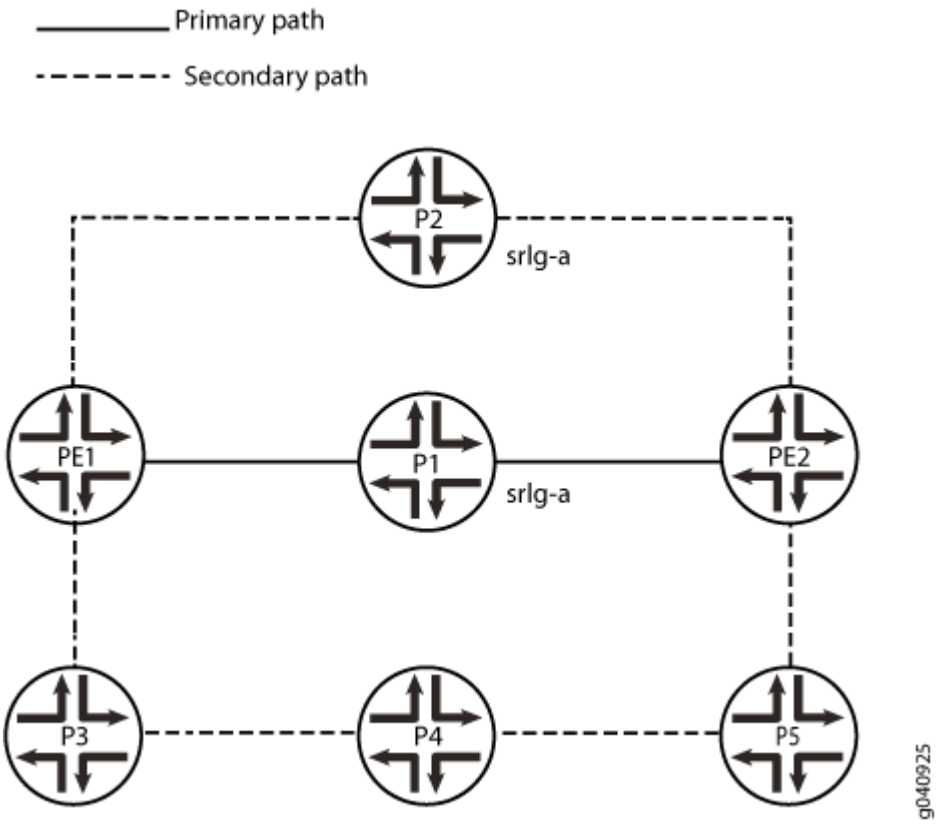
- `srlg-cost`—Include a cost for the SRLG ranging from 1 through 65535. The cost of the SRLG determines the level of impact this SRLG has on the CSPF algorithm for path computations. The higher the cost, the less likely it is for a secondary path to share the same SRLG as the primary path. By default, the `srlg-cost` is 1.
- `srlg-value`—Include a group ID for the SRLG ranging from 1 through 4294967295.

In this example:

- PE1 is the ingress router and PE2 is the egress router.
- P1, P2, and P3, P4, and P5 are transit routers.
  - P1 has direct primary path connections to both the PE1 ingress and PE2 egress routers.
  - P2 has direct secondary path connections to PE1 and PE2.
  - P3 has a direct secondary path connection to PE1, and an indirect secondary path through P4 and P5 to PE2.
  - P4 has indirect secondary paths to PE1 through P3 and to PE2 through P5.
  - P5 has an indirect path through P4 and P3 to PE1 and a direct secondary path to PE2.

OSPF is configured on all the routers as the interior gateway protocol (IGP). SRLG is configured on all seven routers. The primary path includes SRLG `srlg-a`. For the standby secondary path, the link P2>PE2 belongs to SRLG `srlg-a`. The effective link metric, with the added `srlg-cost` of 10, becomes 11. Therefore, the computed secondary path is PE1>P3>P4>P5>PE2 with a CSPF link metric of 4.

Topology



Configuration

IN THIS SECTION

- CLI Quick Configuration | 279
- Procedure | 283

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the `[edit]` hierarchy level.

Router PE1

```
set interfaces ge-0/0/1 unit 0 family inet address 192.168.12.1/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 192.168.13.1/24
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 family inet address 192.168.14.1/24
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.0.1/32
set routing-options srlg srlg-a srlg-value 101
set routing-options srlg srlg-a srlg-cost 10
set protocols rsvp interface ge-0/0/1.0
set protocols rsvp interface ge-0/0/2.0
set protocols rsvp interface ge-0/0/3.0
set protocols mpls optimize-timer 120
set protocols mpls label-switched-path pe1-pe2 to 10.255.0.7
set protocols mpls label-switched-path pe1-pe2 primary via-p1
set protocols mpls label-switched-path pe1-pe2 secondary path2 standby
set protocols mpls path via-p1 10.255.0.2 strict
set protocols mpls path path2
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols mpls interface ge-0/0/3.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
```

```
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

Router P1

```
set interfaces ge-0/0/1 unit 0 family inet address 192.168.12.2/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 192.168.27.2/24
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.0.2/32
set routing-options srlg srlg-a srlg-value 101
set routing-options srlg srlg-a srlg-cost 10
set protocols rsvp interface ge-0/0/1.0
set protocols rsvp interface ge-0/0/2.0
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0 srlg srlg-a
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

Router P2

```
set interfaces ge-0/0/1 unit 0 family inet address 192.168.13.3/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 192.168.37.3/24
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.0.3/32
set routing-options srlg srlg-a srlg-value 101
set routing-options srlg srlg-a srlg-cost 10
set protocols rsvp interface ge-0/0/1.0
set protocols rsvp interface ge-0/0/2.0
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0 srlg srlg-a
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

Router P3

```
set interfaces ge-0/0/1 unit 0 family inet address 192.168.14.4/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 192.168.45.4/24
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.0.4/32
set routing-options srlg srlg-a srlg-value 101
set routing-options srlg srlg-a srlg-cost 10
set protocols rsvp interface ge-0/0/1.0
set protocols rsvp interface ge-0/0/2.0
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

Router P4

```
set interfaces ge-0/0/1 unit 0 family inet address 192.168.45.5/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 192.168.56.5/24
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.0.5/32
set routing-options srlg srlg-a srlg-value 101
set routing-options srlg srlg-a srlg-cost 10
set protocols rsvp interface ge-0/0/1.0
set protocols rsvp interface ge-0/0/2.0
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

Router P5

```
set interfaces ge-0/0/1 unit 0 family inet address 192.168.56.6/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 192.168.67.6/24
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.0.6/32
set routing-options srlg srlg-a srlg-value 101
set routing-options srlg srlg-a srlg-cost 10
set protocols rsvp interface ge-0/0/1.0
set protocols rsvp interface ge-0/0/2.0
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

Router PE2

```
set interfaces ge-0/0/1 unit 0 family inet address 192.168.27.7/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 192.168.37.7/24
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 family inet address 192.168.67.7/24
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.0.7/32
set routing-options srlg srlg-a srlg-value 101
set routing-options srlg srlg-a srlg-cost 10
set protocols rsvp interface ge-0/0/1.0
set protocols rsvp interface ge-0/0/2.0
set protocols rsvp interface ge-0/0/3.0
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols mpls interface ge-0/0/3.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
```

```
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see the Junos OS CLI User Guide.

To configure the ingress router PE1:

1. Configure the device interfaces.

```
[edit interfaces]
user@PE1# set ge-0/0/1 unit 0 family inet address 192.168.12.1/24
user@PE1# set ge-0/0/1 unit 0 family mpls
user@PE1# set ge-0/0/2 unit 0 family inet address 192.168.13.1/24
user@PE1# set ge-0/0/2 unit 0 family mpls
user@PE1# set ge-0/0/3 unit 0 family inet address 192.168.14.1/24
user@PE1# set ge-0/0/3 unit 0 family mpls
user@PE1# set lo0 unit 0 family inet address 10.255.0.1/32
```

2. Configure OSPF on the interfaces.

```
[edit protocols ospf]
user@PE1# set traffic-engineering
user@PE1# set area 0.0.0.0 interface ge-0/0/1.0
user@PE1# set area 0.0.0.0 interface ge-0/0/2.0
user@PE1# set area 0.0.0.0 interface ge-0/0/3.0
user@PE1# set area 0.0.0.0 interface lo0.0
```

3. Configure the SRLG definitions.

```
[edit routing-options]
user@PE1# set srlg srlg-a srlg-value 101
user@PE1# set srlg srlg-a srlg-cost 10
```

4. Configure MPLS and the LSPs.

```
[edit protocols mpls]
user@PE1# set interface ge-0/0/1.0
user@PE1# set interface ge-0/0/2.0
user@PE1# set interface ge-0/0/3.0
user@PE1# set optimize-timer 120
user@PE1# set label-switched-path pe1-pe2 to 10.255.0.7
user@PE1# set label-switched-path pe1-pe2 primary via-p1
user@PE1# set label-switched-path pe1-pe2 secondary path2 standby
user@PE1# set path via-p1 10.255.0.2 strict
user@PE1# set path path2
```

5. Enable RSVP on the interfaces.

```
[edit protocols rsvp]
user@PE1# set interface ge-0/0/1.0
user@PE1# set interface ge-0/0/2.0
user@PE1# set interface ge-0/0/3.0
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols ospf`, `show routing-options`, `show protocols mpls`, and `show protocols rsvp` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@PE1# show interfaces
interfaces {
    ge-0/0/1 {
        unit 0 {
            family inet {
                address 192.168.12.1/24;
            }
            family mpls;
        }
    }
    ge-0/0/2 {
        unit 0 {
            family inet {
```

```
                address 192.168.13.1/24;
            }
            family mpls;
        }
    }
    ge-0/0/3 {
        unit 0 {
            family inet {
                address 192.168.14.1/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.0.1/32;
            }
        }
    }
}
```

```
user@PE1# show protocols ospf
traffic-engineering;
area 0.0.0.0 {
    interface ge-0/0/1.0;;
    interface ge-0/0/2.0;
    interface ge-0/0/3.0;
    interface lo0.0;
}
```

```
user@PE1# show protocols mpls
optimize-timer 120;
label-switched-path pe1-pe2 {
    to 10.255.0.7;
    primary via-p1;
    secondary path2 {
        standby;
    }
}
```

```
path via-p1 {
    10.255.0.2 strict;
}
path path2;
interface ge-0/0/1.0;
interface ge-0/0/2.0;
interface ge-0/0/3.0;
```

```
user@PE1# show protocols rsvp
interface ge-0/0/1.0;
interface ge-0/0/2.0;
interface ge-0/0/3.0;
```

```
user@PE1# show routing-options
routing-options {
    srlg {
        srlg-a {
            srlg-value 101;
            srlg-cost 10;
        }
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

NOTE: Repeat this procedure for every Juniper Networks router in the IGP domain, after modifying the appropriate interface names, addresses, and any other parameters for each router.

Verification

IN THIS SECTION

- Verifying SRLG Definitions | **287**
- Verify TE-Link SRLG | **287**
- Verify Standby Secondary Path | **288**

Confirm that the configuration is working properly.

Verifying SRLG Definitions

Purpose

Verify SRLG-to-value mappings and SRLG cost.

Action

```
user@PE1> show mpls srlg

SRLG             Value      Cost
srlg-a             101        10
```

Verify TE-Link SRLG

Purpose

Verify the traffic engineering link SRLG association.

Action

```
user@PE1> show ted link detail
...
10.255.0.2->192.168.27.7-1, Local: 192.168.27.2, Remote: 0.0.0.0
  Local interface index: 0, Remote interface index: 0
  LocalPath: 1, Metric: 1, StaticBW: 1000Mbps, AvailBW: 1000Mbps
      Color: 0 <none>
  SRLGs: srlg-a
  localBW [0] 0bps  [1] 0bps  [2] 0bps  [3] 0bps
  localBW [4] 0bps  [5] 0bps  [6] 0bps  [7] 0bps

...
10.255.0.3->192.168.37.7-1, Local: 192.168.37.3, Remote: 0.0.0.0
  Local interface index: 0, Remote interface index: 0
  LocalPath: 0, Metric: 1, StaticBW: 1000Mbps, AvailBW: 1000Mbps
      Color: 0 <none>
  SRLGs: srlg-a
  localBW [0] 0bps  [1] 0bps  [2] 0bps  [3] 0bps
```

```
  localBW [4] 0bps  [5] 0bps  [6] 0bps  [7] 0bps

...
```

Meaning

Links P1-PE2 and P2-PE2 are associated with SRLG `srlg-a`.

Verify Standby Secondary Path

Purpose

Check the SRLG link cost and its impact on the CSPF computation of the standby secondary path link.

Action

```
user@PE1> show mpls lsp ingress extensive
Ingress LSP: 1 sessions

10.255.0.7
  From: 10.255.0.1, State: Up, ActiveRoute: 0, LSPname: pe1-pe2
  ActivePath: via-p1 (primary)
  LSPtype: Static Configured
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary   via-p1           State: Up
    Priorities: 7 0
    OptimizeTimer: 120
    SmartOptimizeTimer: 180
    SRLG: srlg-a
    Reoptimization in 110 second(s).
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 2)
 192.168.12.2 S 192.168.27.7 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
          192.168.12.2 192.168.27.7
    7 Oct 13 15:17:11.310 CSPF: computation result ignored, new path no benefit
    6 Oct 13 15:15:14.959 Selected as active path
    5 Oct 13 15:15:14.958 Record Route:  192.168.12.2 192.168.27.7
    4 Oct 13 15:15:14.954 Up
    3 Oct 13 15:15:14.793 Originate Call
```

```
    2 Oct 13 15:15:14.793 CSPF: computation result accepted  192.168.12.2 192.168.27.7
    1 Oct 13 15:14:46.214 CSPF failed: no route toward 10.255.0.2
  Standby   path2            State: Up
    Priorities: 7 0
    OptimizeTimer: 120
    SmartOptimizeTimer: 180
    Reoptimization in 115 second(s).
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 4)
 192.168.14.4 S 192.168.45.5 S 192.168.56.6 S 192.168.67.7 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
          192.168.14.4 192.168.45.5 192.168.56.6 192.168.67.7
   10 Oct 13 15:17:11.929 Record Route:  192.168.14.4 192.168.45.5 192.168.56.6 192.168.67.7
    9 Oct 13 15:17:11.929 Up
    8 Oct 13 15:17:11.729 Originate Call
    7 Oct 13 15:17:11.729 Clear Call
    6 Oct 13 15:17:11.729 CSPF: computation result accepted  192.168.14.4 192.168.45.5
192.168.56.6 192.168.67.7
    5 Oct 13 15:17:11.729 CSPF: Reroute due to re-optimization
    4 Oct 13 15:15:14.984 Record Route:  192.168.13.3 192.168.37.7
    3 Oct 13 15:15:14.984 Up
    2 Oct 13 15:15:14.830 Originate Call
    1 Oct 13 15:15:14.830 CSPF: computation result accepted  192.168.13.3 192.168.37.7
  Created: Thu Oct 13 15:13:46 2011
Total 1 displayed, Up 1, Down 0
```

Meaning

Check the standby secondary path. The effective link cost for P2>PE2 is 11 (with the added `srlg-cost` of 10). CSPF computes the secondary path as PE1>P3>P4>P5>PE2 with a CSPF link metric of 4.

Example: Excluding SRLG Links Completely for the Secondary LSP

IN THIS SECTION

- Requirements | **290**
- Overview | **290**
- Configuration | **291**
- Verification | **296**

This example shows how to configure the `exclude-srlg` option to exclude Shared Risk Link Group (SRLG) links for the secondary label-switched path (LSP).

Requirements

This example uses the following hardware and software components:

- MX Series devices
- Junos OS Release 11.4 or later running on all the devices

Overview

IN THIS SECTION

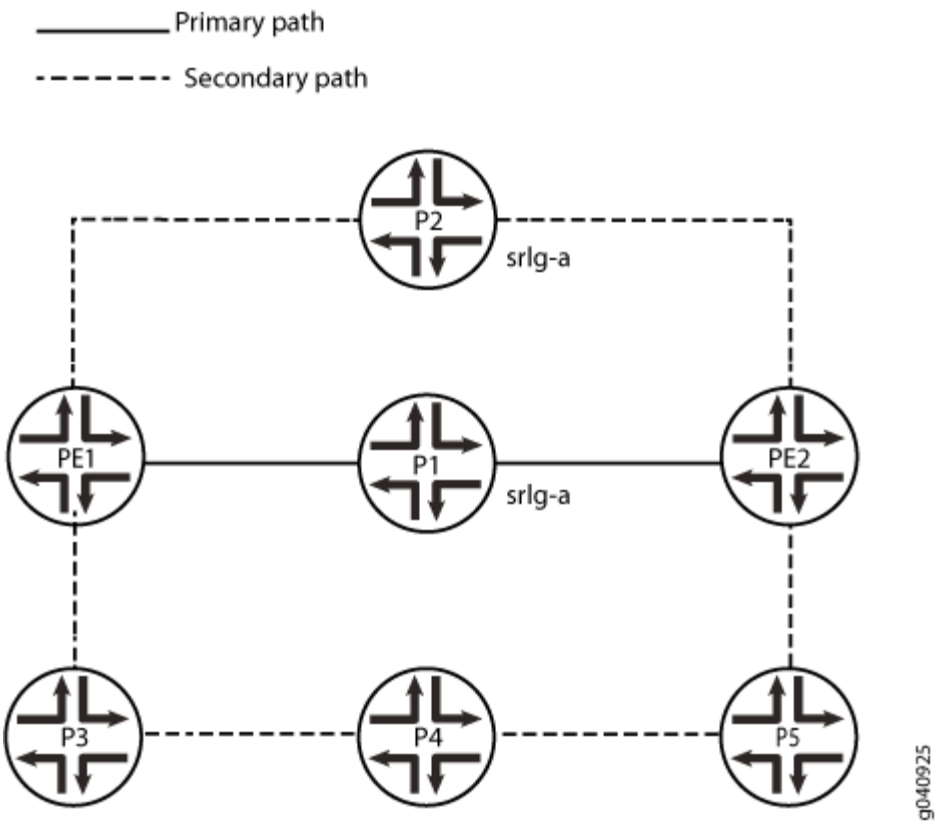
- Topology | **291**

For critical links where it is imperative to keep the secondary and primary paths completely disjoint from any common SRLG, you can optionally configure the `exclude-srlg` statement at the `[edit protocols mpls]` or `[edit protocols mpls label-switched-path` *`path-name`*`]` hierarchy levels. For logical systems, you configure the `exclude-srlg` statement at the `edit logical-systems protocols mpls` `[edit logical-systems` *`logical-system-name`* `protocols mpls label-switched-path` *`path-name`*`]` hierarchy level.

If `exclude-srlg` is configured, the Constrained Shortest Path First (CSPF) algorithm excludes any link belonging to the set of SRLGs in the primary path. If `exclude-srlg` is not configured, and if a link belongs to the set of SRLGs in the primary path, CSPF adds the SRLG cost to the metric, but still accepts the link for computing the path.

In this example, PE1 is the ingress router and PE2 is the egress router. P1, P2, and P3, P4, and P5 are transit routers. OSPF is configured on all the routers as the interior gateway protocol (IGP). SRLG is configured on all seven routers. The primary path includes SRLG `srlg-a`. For the standby secondary path, the link P2>PE2 belongs to SRLG `srlg-a`. Because `exclude-srlg` is configured, CSPF rejects link P2>PE2 as the link belongs to the SRLG `srlg-a`. Therefore, the computed standby secondary path is PE1>P3>P4>P5>PE2.

Topology



Configuration

IN THIS SECTION

- CLI Quick Configuration | **291**
- Procedure | **292**

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the `[edit]` hierarchy level.

Router PE1

```
set interfaces ge-0/0/1 unit 0 family inet address 192.168.12.1/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 192.168.13.1/24
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 family inet address 192.168.14.1/24
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.0.1/32
set routing-options srlg srlg-a srlg-value 101
set protocols rsvp interface ge-0/0/1.0
set protocols rsvp interface ge-0/0/2.0
set protocols rsvp interface ge-0/0/3.0
set protocols mpls optimize-timer 120
set protocols mpls exclude-srlg
set protocols mpls label-switched-path pe1-pe2 to 10.255.0.7
set protocols mpls label-switched-path pe1-pe2 primary via-p1
set protocols mpls label-switched-path pe1-pe2 secondary path2 standby
set protocols mpls path via-p1 10.255.0.2 strict
set protocols mpls path path2
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols mpls interface ge-0/0/3.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see the Junos OS CLI User Guide.

1. Configure the device interfaces.

```
[edit interfaces]
user@PE1# set ge-0/0/1 unit 0 family inet address 192.168.12.1/24
```

```
user@PE1# set ge-0/0/1 unit 0 family mpls
user@PE1# set ge-0/0/2 unit 0 family inet address 192.168.13.1/24
user@PE1# set ge-0/0/2 unit 0 family mpls
user@PE1# set ge-0/0/3 unit 0 family inet address 192.168.14.1/24
user@PE1# set ge-0/0/3 unit 0 family mpls
user@PE1# set lo0 unit 0 family inet address 10.255.0.1/32
```

2. Configure OSPF on the interfaces.

```
[edit protocols ospf]
user@PE1# set traffic-engineering
user@PE1# set area 0.0.0.0 interface ge-0/0/1.0
user@PE1# set area 0.0.0.0 interface ge-0/0/2.0
user@PE1# set area 0.0.0.0 interface ge-0/0/3.0
user@PE1# set area 0.0.0.0 interface lo0.0
```

3. Configure the SRLG definitions.

```
[edit routing-options]
user@PE1# set routing-options srlg srlg-a srlg-value 101
```

4. Configure MPLS and the LSPs.

```
[edit protocols mpls]
user@PE1# set interface ge-0/0/1.0
user@PE1# set interface ge-0/0/2.0
user@PE1# set interface ge-0/0/3.0
user@PE1# set optimize-timer 120
user@PE1# set exclude-srlg
user@PE1# set label-switched-path pe1-pe2 to 10.255.0.7
user@PE1# set label-switched-path pe1-pe2 primary via-p1
user@PE1# set label-switched-path pe1-pe2 secondary path2 standby
user@PE1# set path via-p1 10.255.0.2 strict
user@PE1# set path path2
```

5. Configure the `exclude-srlg` statement to forcibly keep the links for the secondary path completely disjoint from the primary LSP path.

```
user@PE1 set protocols mpls exclude-srlg
```

6. Enable RSVP on the interfaces.

```
[edit protocols rsvp]
user@PE1# set interface ge-0/0/1.0
user@PE1# set interface ge-0/0/2.0
user@PE1# set interface ge-0/0/3.0
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols ospf`, `show routing-options`, `show protocols mpls`, and `show protocols rsvp` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@PE1# show interfaces
interfaces {
    ge-0/0/1 {
        unit 0 {
            family inet {
                address 192.168.12.1/24;
            }
            family mpls;
        }
    }
    ge-0/0/2 {
        unit 0 {
            family inet {
                address 192.168.13.1/24;
            }
            family mpls;
        }
    }
    ge-0/0/3 {
        unit 0 {
            family inet {
```

```
                address 192.168.14.1/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.0.1/32;
            }
        }
    }
}
```

```
user@PE1# show protocols ospf
traffic-engineering;
area 0.0.0.0 {
    interface ge-0/0/1.0;;
    interface ge-0/0/2.0;
    interface ge-0/0/3.0;
    interface lo0.0;
}
```

```
user@PE1# show protocols mpls
optimize-timer 120;
label-switched-path pe1-pe2 {
    to 10.255.0.7;
    primary via-p1;
    secondary path2 {
        standby;
    }
}
path via-p1 {
    10.255.0.2 strict;
}
path path2;
interface ge-0/0/1.0;
```

```
interface ge-0/0/2.0;
interface ge-0/0/3.0;
```

```
user@PE1# show protocols rsvp
interface ge-0/0/1.0;
interface ge-0/0/2.0;
interface ge-0/0/3.0;
```

```
user@PE1# show routing-options
routing-options {
    srlg {
        srlg-a srlg-value 101;
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

NOTE: Repeat this procedure for every Juniper Networks router in the IGP domain, after modifying the appropriate interface names, addresses, and any other parameters for each router.

Verification

IN THIS SECTION

- Verifying the Secondary Path Link for the LSP | **296**

Confirm that the configuration is working properly.

Verifying the Secondary Path Link for the LSP

Purpose

Verify that the link for the secondary path is completely disjoint from the primary path.

Action

```
user@PE1> show mpls lsp detail
Ingress LSP: 1 sessions

10.255.0.7
  From: 10.255.0.1, State: Up, ActiveRoute: 0, LSPname: pe1-pe2
  ActivePath: via-p1 (primary)
  LSPtype: Static Configured
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary   via-p1           State: Up
    Priorities: 7 0
    OptimizeTimer: 120
    SmartOptimizeTimer: 180
    SRLG: srlg-a
    Reoptimization in 77 second(s).
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 2)
 192.168.12.2 S 192.168.27.7 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
          192.168.12.2 192.168.27.7
  Standby   path2            State: Up
    Priorities: 7 0
    OptimizeTimer: 120
    SmartOptimizeTimer: 180
    Reoptimization in 106 second(s).
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 4)
 192.168.14.4 S 192.168.45.5 S 192.168.56.6 S 192.168.67.7 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
          192.168.14.4 192.168.45.5 192.168.56.6 192.168.67.7
Total 1 displayed, Up 1, Down 0

Link P1->PE2: SRLG srlg-a
Link P2->PE2: SRLG srlg-a

Primary path:      PE1-P1-PE2       (CSPF metric: 2)
Standby secondary: PE1-P3-P4-P5-PE2 (CSPF metric: 4)
```

Meaning

Primary path includes SRLG `srlg-a`. For the standby secondary path, the link P2>PE2 belongs to SRLG `srlg-a`. CSPF rejects link P2>PE2 because the link belongs to the SRLG `srlg-a`.

Example: Configuring SRLG with Link Protection

IN THIS SECTION

- Requirements | **298**
- Overview | **298**
- Configuration | **299**
- Verification | **327**

This example shows how to configure SRLG with link protection without the `exclude-srlg` option.

Requirements

This example uses the following hardware and software components:

- MX Series devices
- Junos OS Release 11.4 or later running on all the devices

Overview

IN THIS SECTION

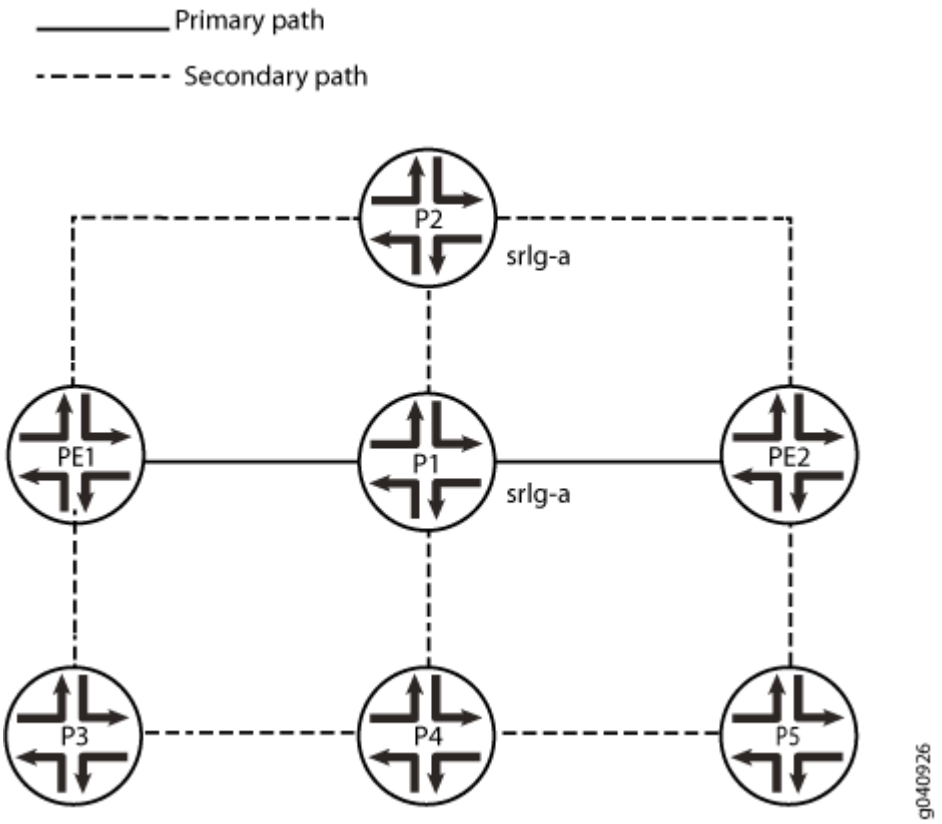
- Topology | **299**

In this example, PE1 is the ingress router and PE2 is the egress router. P1, P2, and P3, P4, and P5 are transit routers. OSPF is configured on all the routers as the interior gateway protocol (IGP). SRLG is configured on all seven routers. The link P1>PE2 (primary path) and the link P2>PE2 belong to SRLG srlg-a.

You configure link protection for the interface P1>PE2 by including the `link-protection` statement.

When SRLG srlg-a is configured on the link P1>PE2 and P2>PE2, the bypass takes the longer path P1>P4>P5>PE2, not selecting the link P2>PE2 because of the added SRLG cost for srlg-a.

Topology



Configuration

IN THIS SECTION

- CLI Quick Configuration | **300**
- Configuring Device PE1 | **304**
- Configuring Device P1 | **307**
- Configuring Device P2 | **311**
- Configuring Device P3 | **314**
- Configuring Device P4 | **317**
- Configuring Device P5 | **320**
- Configuring Device PE2 | **323**

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the `[edit]` hierarchy level.

Router PE1

```
set interfaces ge-0/0/1 unit 0 family inet address 192.168.12.1/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 192.168.13.1/24
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 family inet address 192.168.14.1/24
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.0.1/32
set routing-options srlg srlg-a srlg-value 101
set routing-options srlg srlg-a srlg-cost 10
set protocols rsvp interface ge-0/0/1.0
set protocols rsvp interface ge-0/0/2.0
set protocols rsvp interface ge-0/0/3.0
set protocols mpls optimize-timer 120
set protocols mpls label-switched-path pe1-pe2 to 10.255.0.7
set protocols mpls label-switched-path pe1-pe2 link-protection
set protocols mpls label-switched-path pe1-pe2 primary via-p1
set protocols mpls label-switched-path pe1-pe2 secondary path2 standby
set protocols mpls path via-p1 10.255.0.2 strict
set protocols mpls path path2
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols mpls interface ge-0/0/3.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

Router P1

```
set interfaces ge-0/0/1 unit 0 family inet address 192.168.12.2/24
set interfaces ge-0/0/1 unit 0 family mpls
```

```
set interfaces ge-0/0/2 unit 0 family inet address 192.168.27.2/24
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 family inet address 192.168.23.2/24
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces ge-0/0/4 unit 0 family inet address 192.168.25.2/24
set interfaces ge-0/0/4 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.0.2/32
set routing-options srlg srlg-a srlg-value 101
set routing-options srlg srlg-a srlg-cost 10
set protocols rsvp interface ge-0/0/1.0
set protocols rsvp interface ge-0/0/2.0 link-protection
set protocols rsvp interface ge-0/0/3.0
set protocols rsvp interface ge-0/0/4.0
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0 srlg srlg-a
set protocols mpls interface ge-0/0/3.0
set protocols mpls interface ge-0/0/4.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0
set protocols ospf area 0.0.0.0 interface ge-0/0/4.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

Router P2

```
set interfaces ge-0/0/1 unit 0 family inet address 192.168.13.3/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 192.168.37.3/24
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 family inet address 192.168.23.3/24
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.0.3/32
set routing-options srlg srlg-a srlg-value 101
set routing-options srlg srlg-a srlg-cost 10
set protocols rsvp interface ge-0/0/1.0
set protocols rsvp interface ge-0/0/2.0
set protocols rsvp interface ge-0/0/3.0
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0 srlg srlg-a
set protocols mpls interface ge-0/0/3.0
```

```
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

Router P3

```
set interfaces ge-0/0/1 unit 0 family inet address 192.168.14.4/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 192.168.45.4/24
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.0.4/32
set routing-options srlg srlg-a srlg-value 101
set routing-options srlg srlg-a srlg-cost 10
set protocols rsvp interface ge-0/0/1.0
set protocols rsvp interface ge-0/0/2.0
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

Router P4

```
set interfaces ge-0/0/1 unit 0 family inet address 192.168.45.5/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 192.168.56.5/24
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 family inet address 192.168.25.5/24
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.0.5/32
set routing-options srlg srlg-a srlg-value 101
set routing-options srlg srlg-a srlg-cost 10
set protocols rsvp interface ge-0/0/1.0
set protocols rsvp interface ge-0/0/2.0
set protocols rsvp interface ge-0/0/3.0
set protocols mpls interface ge-0/0/1.0
```

```
set protocols mpls interface ge-0/0/2.0
set protocols mpls interface ge-0/0/3.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

Router P5

```
set interfaces ge-0/0/1 unit 0 family inet address 192.168.56.6/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 192.168.67.6/24
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.0.6/32
set routing-options srlg srlg-a srlg-value 101
set routing-options srlg srlg-a srlg-cost 10
set protocols rsvp interface ge-0/0/1.0
set protocols rsvp interface ge-0/0/2.0
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

Router PE2

```
set interfaces ge-0/0/1 unit 0 family inet address 192.168.27.7/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 192.168.37.7/24
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 family inet address 192.168.67.7/24
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.0.7/32
set routing-options srlg srlg-a srlg-value 101
set routing-options srlg srlg-a srlg-cost 10
set protocols rsvp interface ge-0/0/1.0
set protocols rsvp interface ge-0/0/2.0
```

```
set protocols rsvp interface ge-0/0/3.0
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols mpls interface ge-0/0/3.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

Configuring Device PE1

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see the Junos OS CLI User Guide.

To configure the ingress router PE1:

1. Configure the device interfaces.

```
[edit interfaces]
user@PE1# set ge-0/0/1 unit 0 family inet address 192.168.12.1/24
user@PE1# set ge-0/0/1 unit 0 family mpls
user@PE1# set ge-0/0/2 unit 0 family inet address 192.168.13.1/24
user@PE1# set ge-0/0/2 unit 0 family mpls
user@PE1# set ge-0/0/3 unit 0 family inet address 192.168.14.1/24
user@PE1# set ge-0/0/3 unit 0 family mpls
user@PE1# set lo0 unit 0 family inet address 10.255.0.1/32
```

2. Configure OSPF on the interfaces.

```
[edit protocols ospf]
user@PE1# set traffic-engineering
user@PE1# set area 0.0.0.0 interface ge-0/0/1.0
user@PE1# set area 0.0.0.0 interface ge-0/0/2.0
user@PE1# set area 0.0.0.0 interface ge-0/0/3.0
user@PE1# set area 0.0.0.0 interface lo0.0
```

3. Configure the SRLG definitions.

```
[edit routing-options]
user@PE1# set srlg srlg-a srlg-value 101
user@PE1# set srlg srlg-a srlg-cost 10
```

4. Configure MPLS and the LSPs and configure link protection for the `pe1-pe2` LSP.

```
[edit protocols mpls]
user@PE1# set interface ge-0/0/1.0
user@PE1# set interface ge-0/0/2.0
user@PE1# set interface ge-0/0/3.0
user@PE1# set optimize-timer 120
user@PE1# set label-switched-path pe1-pe2 to 10.255.0.7
user@PE1# set protocols mpls label-switched-path pe1-pe2 link-protection
user@PE1# set label-switched-path pe1-pe2 primary via-p1
user@PE1# set label-switched-path pe1-pe2 secondary path2 standby
user@PE1# set path via-p1 10.255.0.2 strict
user@PE1# set path path2
```

5. Enable RSVP on the interfaces.

```
[edit protocols rsvp]
user@PE1# set interface ge-0/0/1.0
user@PE1# set interface ge-0/0/2.0
user@PE1# set interface ge-0/0/3.0
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols ospf`, `show routing-options`, `show protocols mpls`, and `show protocols rsvp` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@PE1# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            address 192.168.12.1/24;
```

```
        }
        family mpls;
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 192.168.13.1/24;
        }
        family mpls;
    }
}
ge-0/0/3 {
    unit 0 {
        family inet {
            address 192.168.14.1/24;
        }
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.255.0.1/32;
        }
    }
}
}
```

```
user@PE1# show protocols ospf
traffic-engineering;
area 0.0.0.0 {
    interface ge-0/0/1.0;
    interface ge-0/0/2.0;
    interface ge-0/0/3.0;
    interface lo0.0;
}
```

```
user@PE1# show protocols mpls
optimize-timer 120;
```

```
label-switched-path pe1-pe2 {
    to 10.255.0.7;
    link-protection;
    primary via-p1;
    secondary path2 {
        standby;
    }
}
path via-p1 {
    10.255.0.2 strict;
}
path path2;
interface ge-0/0/1.0;
interface ge-0/0/2.0;
interface ge-0/0/3.0;
```

```
user@PE1# show protocols rsvp
interface ge-0/0/1.0;
interface ge-0/0/2.0;
interface ge-0/0/3.0;
```

```
user@PE1# show routing-options
srlg {
    srlg-a {
        srlg-value 101;
        srlg-cost 10;
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Configuring Device P1

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see the Junos OS CLI User Guide.

To configure device P1:

1. Configure the device interfaces.

```
[edit interfaces]
user@P1# set ge-0/0/1 unit 0 family inet address 192.168.12.2/24
user@P1# set ge-0/0/1 unit 0 family mpls
user@P1# set ge-0/0/2 unit 0 family inet address 192.168.27.2/24
user@P1# set ge-0/0/2 unit 0 family mpls
user@P1# set ge-0/0/3 unit 0 family inet address 192.168.23.2/24
user@P1# set ge-0/0/3 unit 0 family mpls
user@P1# set ge-0/0/4 unit 0 family inet address 192.168.25.2/24
user@P1# set ge-0/0/4 unit 0 family mpls
user@P1# set lo0 unit 0 family inet address 10.255.0.2/32
```

2. Configure OSPF on the interfaces.

```
[edit protocols ospf]
user@P1# set traffic-engineering
user@P1# set area 0.0.0.0 interface ge-0/0/1.0
user@P1# set area 0.0.0.0 interface ge-0/0/2.0
user@P1# set area 0.0.0.0 interface ge-0/0/3.0
user@P1# set area 0.0.0.0 interface ge-0/0/4.0
user@P1# set area 0.0.0.0 interface lo0.0
```

3. Configure the SRLG definitions.

```
[edit routing-options]
user@P1# set srlg srlg-a srlg-value 101
user@P1# set srlg srlg-a srlg-cost 10
```

4. Configure MPLS on the interfaces and associate the SRLG `srlg-a` with interface **ge-0/0/2.0** for the P1>PE2 link.

```
[edit protocols mpls]
user@P1# set interface ge-0/0/1.0
user@P1# set interface ge-0/0/2.0 srlg srlg-a
user@P1# set interface ge-0/0/3.0
user@P1# set interface ge-0/0/4.0
```

5. Enable RSVP on the interfaces and configure `link-protection` for interface `ge-0/0/2.0`.

```
[edit protocols rsvp]
user@P1# set interface ge-0/0/1.0
user@P1# set interface ge-0/0/2.0 link-protection
user@P1# set interface ge-0/0/3.0
user@P1# set interface ge-0/0/4.0
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols ospf`, `show protocols mpls`, `show protocols rsvp`, and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@P1# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            address 192.168.12.2/24;
        }
        family mpls;
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 192.168.27.2/24;
        }
        family mpls;
    }
}
ge-0/0/3 {
    unit 0 {
        family inet {
            address 192.168.23.2/24;
        }
        family mpls;
    }
}
ge-0/0/4 {
```

```
    unit 0 {
        family inet {
            address 192.168.25.2/24;
        }
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.255.0.2/32;
        }
    }
}
```

```
user@P1# show protocols ospf
traffic-engineering;
area 0.0.0.0 {
    interface ge-0/0/1.0;
    interface ge-0/0/2.0;
    interface ge-0/0/3.0;
    interface ge-0/0/4.0;
    interface lo0.0;
}
```

```
user@P1# show protocols mpls
interface ge-0/0/1.0;
interface ge-0/0/2.0 {
    srlg srlg-a;
}
interface ge-0/0/3.0;
interface ge-0/0/4.0;
```

```
user@P1# show protocols rsvp
interface ge-0/0/1.0;
interface ge-0/0/2.0 {
    link-protection;
}
```

```
interface ge-0/0/3.0;
interface ge-0/0/4.0;
```

```
user@P1# show routing-options
srlg {
    srlg-a {
        srlg-value 101;
        srlg-cost 10;
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Configuring Device P2

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see the Junos OS CLI User Guide.

To configure P2:

1. Configure the device interfaces.

```
[edit interfaces]
user@P2# set ge-0/0/1 unit 0 family inet address 192.168.13.3/24
user@P2# set ge-0/0/1 unit 0 family mpls
user@P2# set ge-0/0/2 unit 0 family inet address 192.168.37.3/24
user@P2# set ge-0/0/2 unit 0 family mpls
user@P2# set ge-0/0/3 unit 0 family inet address 192.168.23.3/24
user@P2# set ge-0/0/3 unit 0 family mpls
user@P2# set lo0 unit 0 family inet address 10.255.0.3/32
```

2. Configure OSPF on the interfaces.

```
[edit protocols ospf]
user@P2# set traffic-engineering
user@P2# set area 0.0.0.0 interface ge-0/0/1.0
user@P2# set area 0.0.0.0 interface ge-0/0/2.0
```

```
user@P2# set area 0.0.0.0 interface ge-0/0/3.0
user@P2# set area 0.0.0.0 interface lo0.0
```

3. Configure the SRLG definitions.

```
[edit routing-options]
user@P2# set srlg srlg-a srlg-value 101
user@P2# set srlg srlg-a srlg-cost 10
```

4. Configure MPLS on the interfaces and associate the SRLG `srlg-a` with interface **ge-0/0/2.0** for the P2>PE2 link.

```
[edit protocols mpls]
user@P2# set interface ge-0/0/1.0
user@P2# set interface ge-0/0/2.0 srlg srlg-a
user@P2# set interface ge-0/0/3.0
```

5. Enable RSVP on the interfaces.

```
[edit protocols rsvp]
user@P2# set interface ge-0/0/1.0
user@P2# set interface ge-0/0/2.0
user@P2# set interface ge-0/0/3.0
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols ospf`, `show protocols mpls`, `show protocols rsvp`, and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@P2# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            address 192.168.13.3/24;
        }
        family mpls;
    }
```

```
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 192.168.37.3/24;
        }
        family mpls;
    }
}
ge-0/0/3 {
    unit 0 {
        family inet {
            address 192.168.23.3/24;
        }
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.255.0.3/32;
        }
    }
}
}
```

```
user@P2# show protocols ospf
traffic-engineering;
area 0.0.0.0 {
    interface ge-0/0/1.0;
    interface ge-0/0/2.0;
    interface ge-0/0/3.0;
    interface lo0.0;
}
```

```
user@P2# show protocols mpls
interface ge-0/0/1.0;
interface ge-0/0/2.0 {
    srlg srlg-a;
}
```

```
interface ge-0/0/3.0;
}
```

```
user@P2# show protocols rsvp
interface ge-0/0/1.0;
interface ge-0/0/2.0;
interface ge-0/0/3.0;
```

```
user@P2# show routing-options
srlg {
    srlg-a {
        srlg-value 101;
        srlg-cost 10;
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Configuring Device P3

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see the Junos OS CLI User Guide.

To configure P3:

1. Configure the device interfaces.

```
[edit interfaces]
user@P3# set ge-0/0/1 unit 0 family inet address 192.168.14.4/24
user@P3# set ge-0/0/1 unit 0 family mpls
user@P3# set ge-0/0/2 unit 0 family inet address 192.168.45.4/24
user@P3# set ge-0/0/2 unit 0 family mpls
user@P3# set lo0 unit 0 family inet address 10.255.0.4/32
```

2. Configure OSPF on the interfaces.

```
[edit protocols ospf]
user@P3# set traffic-engineering
user@P3# set area 0.0.0.0 interface ge-0/0/1.0
user@P3# set area 0.0.0.0 interface ge-0/0/2.0
user@P3# set area 0.0.0.0 interface lo0.0
```

3. Configure the SRLG definitions.

```
[edit routing-options]
user@P3# set srlg srlg-a srlg-value 101
user@P3# set srlg srlg-a srlg-cost 10
```

4. Configure MPLS on the interfaces.

```
[edit protocols mpls]
user@P3# set interface ge-0/0/1.0
user@P3# set interface ge-0/0/2.0
```

5. Enable RSVP on the interfaces.

```
[edit protocols rsvp]
user@P3# set interface ge-0/0/1.0
user@P3# set interface ge-0/0/2.0
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols ospf`, `show protocols mpls`, `show protocols rsvp`, and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@P3# show interfaces
interfaces {
    ge-0/0/1 {
        unit 0 {
            family inet {
```

```
                address 192.168.14.4/24;
            }
            family mpls;
        }
    }
    ge-0/0/2 {
        unit 0 {
            family inet {
                address 192.168.45.4/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.0.4/32;
            }
        }
    }
}
```

```
user@P3# show protocols ospf
traffic-engineering;
area 0.0.0.0 {
    interface ge-0/0/1.0;
    interface ge-0/0/2.0;
```

```
    interface lo0.0;
}
```

```
user@P3# show protocols mpls
interface ge-0/0/1.0;
interface ge-0/0/2.0;
```

```
user@P3# show protocols rsvp
interface ge-0/0/1.0;
interface ge-0/0/2.0;
```

```
user@P3# show routing-options
srlg {
    srlg-a {
        srlg-value 101;
        srlg-cost 10;
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Configuring Device P4

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see the Junos OS CLI User Guide.

To configure P4:

1. Configure the device interfaces.

```
[edit interfaces]
user@P4# set ge-0/0/1 unit 0 family inet address 192.168.45.5/24
user@P4# set ge-0/0/1 unit 0 family mpls
user@P4# set ge-0/0/2 unit 0 family inet address 192.168.56.5/24
user@P4# set ge-0/0/2 unit 0 family mpls
user@P4# set ge-0/0/3 unit 0 family inet address 192.168.25.5/24
```

```
user@P4# set ge-0/0/3 unit 0 family mpls
user@P4# set lo0 unit 0 family inet address 10.255.0.5/32
```

2. Configure OSPF on the interfaces.

```
[edit protocols ospf]
user@P4# set traffic-engineering
user@P4# set area 0.0.0.0 interface ge-0/0/1.0
user@P4# set area 0.0.0.0 interface ge-0/0/2.0
user@P4# set area 0.0.0.0 interface ge-0/0/3.0
user@P4# set area 0.0.0.0 interface lo0.0
```

3. Configure the SRLG definitions.

```
[edit routing-options]
user@P4# set srlg srlg-a srlg-value 101
user@P4# set srlg srlg-a srlg-cost 10
```

4. Configure MPLS on the interfaces.

```
[edit protocols mpls]
user@P4# set interface ge-0/0/1.0
user@P4# set interface ge-0/0/2.0
user@P4# set interface ge-0/0/3.0
```

5. Enable RSVP on the interfaces.

```
[edit protocols rsvp]
user@P4# set interface ge-0/0/1.0
user@P4# set interface ge-0/0/2.0
user@P4# set interface ge-0/0/3.0
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols ospf`, `show protocols mpls`, `show protocols rsvp`, and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@P4# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            address 192.168.45.5/24;
        }
        family mpls;
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 192.168.56.5/24;
        }
        family mpls;
    }
}
ge-0/0/3 {
    unit 0 {
        family inet {
            address 192.168.25.5/24;
        }
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.255.0.5/32;
        }
    }
}
```

```
user@P4# show protocols ospf
traffic-engineering;
```

```
area 0.0.0.0 {
    interface ge-0/0/1.0;
    interface ge-0/0/2.0;
    interface ge-0/0/3.0;
    interface lo0.0;
}
```

```
user@P4# show protocols mpls
interface ge-0/0/1.0;
interface ge-0/0/2.0;
interface ge-0/0/3.0;
```

```
user@P4# show protocols rsvp
interface ge-0/0/1.0;
interface ge-0/0/2.0;
interface ge-0/0/3.0;
```

```
user@P4# show routing-options
srlg {
    srlg-a {
        srlg-value 101;
        srlg-cost 10;
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Configuring Device P5

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see the Junos OS CLI User Guide.

To configure P5:

1. Configure the device interfaces.

```
[edit interfaces]
user@P5# set ge-0/0/1 unit 0 family inet address 192.168.56.6/24
user@P5# set ge-0/0/1 unit 0 family mpls
user@P5# set ge-0/0/2 unit 0 family inet address 192.168.67.6/24
user@P5# set ge-0/0/2 unit 0 family mpls
user@P5# set lo0 unit 0 family inet address 10.255.0.6/32
```

2. Configure OSPF on the interfaces.

```
[edit protocols ospf]
user@P5# set traffic-engineering
user@P5# set area 0.0.0.0 interface ge-0/0/1.0
user@P5# set area 0.0.0.0 interface ge-0/0/2.0
user@P5# set area 0.0.0.0 interface lo0.0
```

3. Configure the SRLG definitions.

```
[edit routing-options]
user@P5# set srlg srlg-a srlg-value 101
user@P5# set srlg srlg-a srlg-cost 10
```

4. Configure MPLS on the interfaces.

```
[edit protocols mpls]
user@P5# set interface ge-0/0/1.0
user@P5# set interface ge-0/0/2.0
```

5. Enable RSVP on the interfaces.

```
[edit protocols rsvp]
user@P5# set interface ge-0/0/1.0
user@P5# set interface ge-0/0/2.0
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols ospf`, `show protocols mpls`, `show protocols rsvp`, and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@P5# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            address 192.168.56.6/24;
        }
        family mpls;
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 192.168.67.6/24;
        }
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.255.0.6/32;
        }
    }
}
```

```
user@P5# show protocols ospf
traffic-engineering;
area 0.0.0.0 {
    interface ge-0/0/1.0;
    interface ge-0/0/2.0;
```

```
    interface lo0.0;
}
```

```
user@P5# show protocols mpls
interface ge-0/0/1.0;
interface ge-0/0/2.0;
```

```
user@P5# show protocols rsvp
interface ge-0/0/1.0;
interface ge-0/0/2.0;
```

```
user@P5# show routing-options
srlg {
    srlg-a {
        srlg-value 101;
        srlg-cost 10;
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Configuring Device PE2

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see the Junos OS CLI User Guide.

To configure PE2:

1. Configure the device interfaces.

```
                        [edit interfaces]
user@PE2# set ge-0/0/1 unit 0 family inet address 192.168.27.7/24
user@PE2# set ge-0/0/1 unit 0 family mpls
user@PE2# set ge-0/0/2 unit 0 family inet address 192.168.37.7/24
```

```
user@PE2# set ge-0/0/2 unit 0 family mpls
user@PE2# set ge-0/0/3 unit 0 family inet address 192.168.67.7/24
user@PE2# set ge-0/0/3 unit 0 family mpls
user@PE2# set lo0 unit 0 family inet address 10.255.0.7/32
```

2. Configure OSPF on the interfaces.

```
[edit protocols ospf]
user@PE2# set traffic-engineering
user@PE2# set area 0.0.0.0 interface ge-0/0/1.0
user@PE2# set area 0.0.0.0 interface ge-0/0/2.0
user@PE2# set area 0.0.0.0 interface ge-0/0/3.0
user@PE2# set area 0.0.0.0 interface lo0.0
```

3. Configure the SRLG definitions.

```
[edit routing-options]
user@PE2# set srlg srlg-a srlg-value 101
user@PE2# set srlg srlg-a srlg-cost 10
```

4. Configure MPLS on the interfaces.

```
[edit protocols mpls]
user@PE2# set interface ge-0/0/1.0
user@PE2# set interface ge-0/0/2.0
user@PE2# set interface ge-0/0/3.0
```

5. Enable RSVP on the interfaces.

```
[edit protocols rsvp]
user@PE2# set interface ge-0/0/1.0
user@PE2# set interface ge-0/0/2.0
user@PE2# set interface ge-0/0/3.0
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols ospf`, `show protocols mpls`, `show protocols rsvp`, and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@PE2# show interfaces
interfaces {
    ge-0/0/1 {
        unit 0 {
            family inet {
                address 192.168.27.7/24;
            }
            family mpls;
        }
    }
    ge-0/0/2 {
        unit 0 {
            family inet {
                address 192.168.37.7/24;
            }
            family mpls;
        }
    }
    ge-0/0/3 {
        unit 0 {
            family inet {
                address 192.168.67.7/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.0.7/32;
            }
        }
```

```
    }
}
```

```
user@PE2# show protocols ospf
traffic-engineering;
area 0.0.0.0 {
    interface ge-0/0/1.0;
    interface ge-0/0/2.0;
    interface ge-0/0/3.0;
    interface lo0.0;
}
```

```
user@PE2# show protocols mpls
interface ge-0/0/1.0;
interface ge-0/0/2.0;
interface ge-0/0/3.0;
```

```
user@PE2# show protocols rsvp
interface ge-0/0/1.0;
interface ge-0/0/2.0;
interface ge-0/0/3.0;
```

```
user@PE2# show routing-options
srlg {
    srlg-a {
        srlg-value 101;
        srlg-cost 10;
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

IN THIS SECTION

- Verifying the SRLG Cost Is Added to the TE Link | 327

Confirm that the configuration is working properly.

Verifying the SRLG Cost Is Added to the TE Link

Purpose

Verify that the SRLG cost is added to the TE link if it belongs to the SRLG of the protected link. Issue the `show ted link detail` and `show rsvp session extensive bypass` commands on device P1.

Action

```
user@P1> show ted link detail

...
10.255.0.2->192.168.27.7-1, Local: 192.168.27.2, Remote: 0.0.0.0
  Local interface index: 0, Remote interface index: 0
  LocalPath: 0, Metric: 1, StaticBW: 1000Mbps, AvailBW: 1000Mbps
      Color: 0 <none>
  SRLGs: srlg-a
  localBW [0] 0bps  [1] 0bps  [2] 0bps  [3] 0bps
  localBW [4] 0bps  [5] 0bps  [6] 0bps  [7] 0bps
[...]
10.255.0.3->192.168.37.7-1, Local: 192.168.37.3, Remote: 0.0.0.0
  Local interface index: 0, Remote interface index: 0
  LocalPath: 0, Metric: 1, StaticBW: 1000Mbps, AvailBW: 1000Mbps
      Color: 0 <none>
  SRLGs: srlg-a
  localBW [0] 0bps  [1] 0bps  [2] 0bps  [3] 0bps
```

```
  localBW [4] 0bps  [5] 0bps  [6] 0bps  [7] 0bps
...
```

```
user@P1> show rsvp session extensive bypass

Ingress RSVP: 1 sessions

10.255.0.7
  From: 10.255.0.2, LSPstate: Up, ActiveRoute: 0
  LSPname: Bypass->192.168.27.7
  LSPtype: Static Configured
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 299776
  Resv style: 1 SE, Label in: -, Label out: 299776
  Time left:    -, Since: Fri Oct 21 13:19:21 2011
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 52081 protocol 0
  Type: Bypass LSP
    Number of data route tunnel through: 1
    Number of RSVP session tunnel through: 0
  PATH rcvfrom: localclient
  Adspec: sent MTU 1500
  Path MTU: received 1500
  PATH sentto: 192.168.25.5 (ge-0/0/4.0) 26 pkts
  RESV rcvfrom: 192.168.25.5 (ge-0/0/4.0) 26 pkts
  Explct route: 192.168.25.5 192.168.56.6 192.168.67.7
  Record route: <self> 192.168.25.5 192.168.56.6 192.168.67.7
Total 1 displayed, Up 1, Down 0
```

Meaning

The shortest path for the bypass protecting the link P1->PE2 would have been P1->P2->PE2. Because the links P1>PE2 and P2>PE2 both belong to SRLG `srlg-a`, the SRLG cost of 10 for `srlg-a` is added to the metric for the link P2>PE2. This makes the metric for the link P2>PE2 too high to be selected for the shortest path. Therefore, the CSPF result for the computed path for the bypass becomes P1>P4>P5>PE2.

Example: Configuring SRLG with Link Protection with the exclude-srlg Option

IN THIS SECTION

- Requirements | 329
- Overview | 329
- Configuration | 330
- Verification | 358

This example shows how to configure SRLG with link protection with the `exclude-srlg` option.

Requirements

This example uses the following hardware and software components:

- MX Series devices
- Junos OS Release 11.4 or later running on all the devices

Overview

IN THIS SECTION

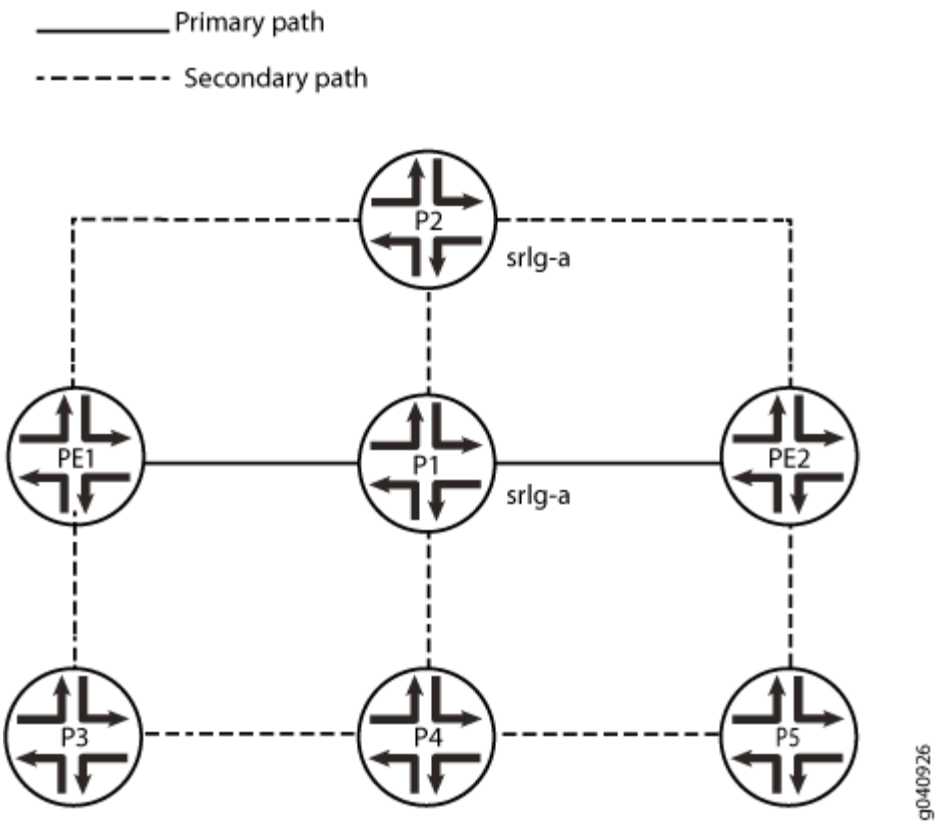
- Topology | 330

In this example, PE1 is the ingress router and PE2 is the egress router. P1, P2, and P3, P4, and P5 are transit routers. OSPF is configured on all the routers as the interior gateway protocol (IGP). SRLG is configured on all seven routers. The link P1>PE2 (primary path) and the link P2>PE2 belong to SRLG `srlg-a`.

You configure link protection for the interface P1>PE2 by including the `link-protection` statement along with the `exclude-srlg` option. This makes the bypass LSP and the protected link completely disjoint in any SRLG.

When SRLG `srlg-a` is configured on the link P1>PE2 and P2>PE2, the link P2>PE2 is rejected for CSPF consideration due to the `exclude-srlg` configuration. Therefore, the computed path for the bypass becomes P1>P4>P5>PE2.

Topology



Configuration

IN THIS SECTION

- CLI Quick Configuration | **331**
- Configuring Device PE1 | **335**
- Configuring Device P1 | **338**
- Configuring Device P2 | **342**
- Configuring Device P3 | **345**
- Configuring Device P4 | **348**
- Configuring Device P5 | **351**
- Configuring Device PE2 | **354**

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the `[edit]` hierarchy level.

Router PE1

```
set interfaces ge-0/0/1 unit 0 family inet address 192.168.12.1/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 192.168.13.1/24
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 family inet address 192.168.14.1/24
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.0.1/32
set routing-options srlg srlg-a srlg-value 101
set protocols rsvp interface ge-0/0/1.0
set protocols rsvp interface ge-0/0/2.0
set protocols rsvp interface ge-0/0/3.0
set protocols mpls optimize-timer 120
set protocols mpls label-switched-path pe1-pe2 to 10.255.0.7
set protocols mpls label-switched-path pe1-pe2 link-protection
set protocols mpls label-switched-path pe1-pe2 primary via-p1
set protocols mpls label-switched-path pe1-pe2 secondary path2 standby
set protocols mpls path via-p1 10.255.0.2 strict
set protocols mpls path path2
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols mpls interface ge-0/0/3.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

Router P1

```
set interfaces ge-0/0/1 unit 0 family inet address 192.168.12.2/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 192.168.27.2/24
```

```
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 family inet address 192.168.23.2/24
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces ge-0/0/4 unit 0 family inet address 192.168.25.2/24
set interfaces ge-0/0/4 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.0.2/32
set routing-options srlg srlg-a srlg-value 101
set routing-options srlg srlg-a srlg-cost 10
set protocols rsvp interface ge-0/0/1.0
set protocols rsvp interface ge-0/0/2.0 link-protection exclude-srlg
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0 srlg srlg-a
set protocols mpls interface ge-0/0/3.0
set protocols mpls interface ge-0/0/4.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0
set protocols ospf area 0.0.0.0 interface ge-0/0/4.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

Router P2

```
set interfaces ge-0/0/1 unit 0 family inet address 192.168.13.3/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 192.168.37.3/24
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 family inet address 192.168.23.3/24
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.0.3/32
set routing-options srlg srlg-a srlg-value 101
set routing-options srlg srlg-a srlg-cost 10
set protocols rsvp interface ge-0/0/1.0
set protocols rsvp interface ge-0/0/2.0
set protocols rsvp interface ge-0/0/3.0
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0 srlg srlg-a
set protocols mpls interface ge-0/0/3.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
```

```
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

Router P3

```
set interfaces ge-0/0/1 unit 0 family inet address 192.168.14.4/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 192.168.45.4/24
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.0.4/32
set routing-options srlg srlg-a srlg-value 101
set routing-options srlg srlg-a srlg-cost 10
set protocols rsvp interface ge-0/0/1.0
set protocols rsvp interface ge-0/0/2.0
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

Router P4

```
set interfaces ge-0/0/1 unit 0 family inet address 192.168.45.5/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 192.168.56.5/24
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 family inet address 192.168.25.5/24
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.0.5/32
set routing-options srlg srlg-a srlg-value 101
set routing-options srlg srlg-a srlg-cost 10
set protocols rsvp interface ge-0/0/1.0
set protocols rsvp interface ge-0/0/2.0
set protocols rsvp interface ge-0/0/3.0
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols mpls interface ge-0/0/3.0
set protocols ospf traffic-engineering
```

```
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

Router P5

```
set interfaces ge-0/0/1 unit 0 family inet address 192.168.56.6/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 192.168.67.6/24
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.0.6/32
set routing-options srlg srlg-a srlg-value 101
set routing-options srlg srlg-a srlg-cost 10
set protocols rsvp interface ge-0/0/1.0
set protocols rsvp interface ge-0/0/2.0
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

Router PE2

```
set interfaces ge-0/0/1 unit 0 family inet address 192.168.27.7/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 192.168.37.7/24
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 family inet address 192.168.67.7/24
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.0.7/32
set routing-options srlg srlg-a srlg-value 101
set routing-options srlg srlg-a srlg-cost 10
set protocols rsvp interface ge-0/0/1.0
set protocols rsvp interface ge-0/0/2.0
set protocols rsvp interface ge-0/0/3.0
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
```

```
set protocols mpls interface ge-0/0/3.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

Configuring Device PE1

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see the Junos OS CLI User Guide.

To configure the ingress router PE1:

1. Configure the device interfaces.

```
[edit interfaces]
user@PE1# set ge-0/0/1 unit 0 family inet address 192.168.12.1/24
user@PE1# set ge-0/0/1 unit 0 family mpls
user@PE1# set ge-0/0/2 unit 0 family inet address 192.168.13.1/24
user@PE1# set ge-0/0/2 unit 0 family mpls
user@PE1# set ge-0/0/3 unit 0 family inet address 192.168.14.1/24
user@PE1# set ge-0/0/3 unit 0 family mpls
user@PE1# set lo0 unit 0 family inet address 10.255.0.1/32
```

2. Configure OSPF on the interfaces.

```
[edit protocols ospf]
user@PE1# set traffic-engineering
user@PE1# set area 0.0.0.0 interface ge-0/0/1.0
user@PE1# set area 0.0.0.0 interface ge-0/0/2.0
user@PE1# set area 0.0.0.0 interface ge-0/0/3.0
user@PE1# set area 0.0.0.0 interface lo0.0
```

3. Configure the SRLG definitions.

```
[edit routing-options]
user@PE1# set routing-options srlg srlg-a srlg-value 101
user@PE1# set routing-options srlg srlg-a srlg-cost 10
```

4. Configure MPLS and the LSPs and configure link protection for the `pe1-pe2` LSP.

```
[edit protocols mpls]
user@PE1# set interface ge-0/0/1.0
user@PE1# set interface ge-0/0/2.0
user@PE1# set interface ge-0/0/3.0
user@PE1# set optimize-timer 120
user@PE1# set label-switched-path pe1-pe2 to 10.255.0.7
user@PE1# set protocols mpls label-switched-path pe1-pe2 link-protection
user@PE1# set label-switched-path pe1-pe2 primary via-p1
user@PE1# set label-switched-path pe1-pe2 secondary path2 standby
user@PE1# set path via-p1 10.255.0.2 strict
user@PE1# set path path2
```

5. Enable RSVP on the interfaces.

```
[edit protocols rsvp]
user@PE1# set interface ge-0/0/1.0
user@PE1# set interface ge-0/0/2.0
user@PE1# set interface ge-0/0/3.0
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols ospf`, `show routing-options`, `show protocols mpls`, and `show protocols rsvp` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@PE1# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            address 192.168.12.1/24;
```

```
        }
        family mpls;
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 192.168.13.1/24;
        }
        family mpls;
    }
}
ge-0/0/3 {
    unit 0 {
        family inet {
            address 192.168.14.1/24;
        }
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.255.0.1/32;
        }
    }
}
}
```

```
user@PE1# show protocols ospf
traffic-engineering;
area 0.0.0.0 {
    interface ge-0/0/1.0;
    interface ge-0/0/2.0;
    interface ge-0/0/3.0;
    interface lo0.0;
}
```

```
user@PE1# show protocols mpls
optimize-timer 120;
```

```
label-switched-path pe1-pe2 {
    to 10.255.0.7;
    link-protection;
    primary via-p1;
    secondary path2 {
        standby;
    }
}
path via-p1 {
    10.255.0.2 strict;
}
path path2;
interface ge-0/0/1.0;
interface ge-0/0/2.0;
interface ge-0/0/3.0;
```

```
user@PE1# show protocols rsvp
interface ge-0/0/1.0;
interface ge-0/0/2.0;
interface ge-0/0/3.0;
```

```
user@PE1# show routing-options
srlg {
    srlg-a {
        srlg-value 101;
        srlg-cost 10;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Device P1

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see the Junos OS CLI User Guide.

To configure device P1:

1. Configure the device interfaces.

```
[edit interfaces]
user@P1# set ge-0/0/1 unit 0 family inet address 192.168.12.2/24
user@P1# set ge-0/0/1 unit 0 family mpls
user@P1# set ge-0/0/2 unit 0 family inet address 192.168.27.2/24
user@P1# set ge-0/0/2 unit 0 family mpls
user@P1# set ge-0/0/3 unit 0 family inet address 192.168.23.2/24
user@P1# set ge-0/0/3 unit 0 family mpls
user@P1# set ge-0/0/4 unit 0 family inet address 192.168.25.2/24
user@P1# set ge-0/0/4 unit 0 family mpls
user@P1# set lo0 unit 0 family inet address 10.255.0.2/32
```

2. Configure OSPF on the interfaces.

```
[edit protocols ospf]
user@P1# set traffic-engineering
user@P1# set area 0.0.0.0 interface ge-0/0/1.0
user@P1# set area 0.0.0.0 interface ge-0/0/2.0
user@P1# set area 0.0.0.0 interface ge-0/0/3.0
user@P1# set area 0.0.0.0 interface ge-0/0/4.0
user@P1# set area 0.0.0.0 interface lo0.0
```

3. Configure the SRLG definitions.

```
[edit routing-options]
user@P1# set routing-options srlg srlg-a srlg-value 101
user@P1# set routing-options srlg srlg-a srlg-cost 10
```

4. Configure MPLS on the interfaces and associate the SRLG with interface **ge-0/0/2.0** for the P1>PE2 link.

```
[edit protocols mpls]
user@P1# set interface ge-0/0/1.0
user@P1# set interface ge-0/0/2.0 srlg srlg-a
user@P1# set interface ge-0/0/3.0
user@P1# set interface ge-0/0/4.0
```

5. Enable RSVP on the interfaces and include the `link-protection` statement with the `exclude-srlg` option for interface `ge-0/0/2.0`.

```
[edit protocols rsvp]
user@P1# set interface ge-0/0/1.0
user@P1# set interface ge-0/0/2.0 link-protection exclude-srlg
user@P1# set interface ge-0/0/3.0
user@P1# set interface ge-0/0/4.0
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols ospf`, `show protocols mpls`, `show protocols rsvp`, and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@P1# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            address 192.168.12.2/24;
        }
        family mpls;
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 192.168.27.2/24;
        }
        family mpls;
    }
}
ge-0/0/3 {
    unit 0 {
        family inet {
            address 192.168.23.2/24;
        }
        family mpls;
    }
}
```

```
ge-0/0/4 {
    unit 0 {
        family inet {
            address 192.168.25.2/24;
        }
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.255.0.2/32;
        }
    }
}
```

```
user@P1# show protocols ospf
traffic-engineering;
area 0.0.0.0 {
    interface ge-0/0/1.0;
    interface ge-0/0/2.0;
    interface ge-0/0/3.0;
    interface ge-0/0/4.0;
    interface lo0.0;
}
```

```
user@P1# show protocols mpls
interface ge-0/0/1.0;
interface ge-0/0/2.0 {
    srlg srlg-a;
}
interface ge-0/0/3.0;
interface ge-0/0/4.0;
```

```
user@P1# show protocols rsvp
interface ge-0/0/1.0;
interface ge-0/0/2.0 {
    link-protection {
        exclude-srlg;
```

```
    }
interface ge-0/0/3.0;
interface ge-0/0/4.0;
}
```

```
user@P1# show routing-options
srlg {
    srlg-a {
        srlg-value 101;
        srlg-cost 10;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Device P2

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see the Junos OS CLI User Guide.

To configure P2:

1. Configure the device interfaces.

```
[edit interfaces]
user@P2# set ge-0/0/1 unit 0 family inet address 192.168.13.3/24
user@P2# set ge-0/0/1 unit 0 family mpls
user@P2# set ge-0/0/2 unit 0 family inet address 192.168.37.3/24
user@P2# set ge-0/0/2 unit 0 family mpls
user@P2# set ge-0/0/3 unit 0 family inet address 192.168.23.3/24
user@P2# set ge-0/0/3 unit 0 family mpls
user@P2# set lo0 unit 0 family inet address 10.255.0.3/32
```

2. Configure OSPF on the interfaces.

```
[edit protocols ospf]
user@P2# set traffic-engineering
user@P2# set area 0.0.0.0 interface ge-0/0/1.0
```

```
user@P2# set area 0.0.0.0 interface ge-0/0/2.0
user@P2# set area 0.0.0.0 interface ge-0/0/3.0
user@P2# set area 0.0.0.0 interface lo0.0
```

3. Configure the SRLG definitions.

```
[edit routing-options]
user@P2# set routing-options srlg srlg-a srlg-value 101
user@P2# set routing-options srlg srlg-a srlg-cost 10
```

4. Configure MPLS on the interfaces and associate the SRLG with interface **ge-0/0/2.0** for the P2>PE2 link.

```
[edit protocols mpls]
user@P2# set interface ge-0/0/1.0
user@P2# set interface ge-0/0/2.0 srlg srlg-a
user@P2# set interface ge-0/0/3.0
```

5. Enable RSVP on the interfaces.

```
[edit protocols rsvp]
user@P2# set interface ge-0/0/1.0
user@P2# set interface ge-0/0/2.0
user@P2# set interface ge-0/0/3.0
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols ospf`, `show protocols mpls`, `show protocols rsvp`, and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@P2# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            address 192.168.13.3/24;
        }
        family mpls;
```

```
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 192.168.37.3/24;
        }
        family mpls;
    }
}
ge-0/0/3 {
    unit 0 {
        family inet {
            address 192.168.23.3/24;
        }
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.255.0.3/32;
        }
    }
}
}
```

```
user@P2# show protocols ospf
traffic-engineering;
area 0.0.0.0 {
    interface ge-0/0/1.0;
    interface ge-0/0/2.0;
    interface ge-0/0/3.0;
    interface lo0.0;
}
```

```
user@P2# show protocols mpls
interface ge-0/0/1.0;
interface ge-0/0/2.0 {
    srlg srlg-a;
```

```
}
interface ge-0/0/3.0;
}
```

```
user@P2# show protocols rsvp
interface ge-0/0/1.0;
interface ge-0/0/2.0;
interface ge-0/0/3.0;
```

```
user@P2# show routing-options
srlg {
    srlg-a {
        srlg-value 101;
        srlg-cost 10;
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Configuring Device P3

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see the Junos OS CLI User Guide.

To configure P3:

1. Configure the device interfaces.

```
[edit interfaces]
user@P3# set ge-0/0/1 unit 0 family inet address 192.168.14.4/24
user@P3# set ge-0/0/1 unit 0 family mpls
user@P3# set ge-0/0/2 unit 0 family inet address 192.168.45.4/24
user@P3# set ge-0/0/2 unit 0 family mpls
user@P3# set lo0 unit 0 family inet address 10.255.0.4/32
```

2. Configure OSPF on the interfaces.

```
[edit protocols ospf]
user@P3# set traffic-engineering
user@P3# set area 0.0.0.0 interface ge-0/0/1.0
user@P3# set area 0.0.0.0 interface ge-0/0/2.0
user@P3# set area 0.0.0.0 interface lo0.0
```

3. Configure the SRLG definitions.

```
[edit routing-options]
user@P3# set routing-options srlg srlg-a srlg-value 101
user@P3# set routing-options srlg srlg-a srlg-cost 10
```

4. Configure MPLS on the interfaces.

```
[edit protocols mpls]
user@P3# set interface ge-0/0/1.0
user@P3# set interface ge-0/0/2.0
```

5. Enable RSVP on the interfaces.

```
[edit protocols rsvp]
user@P3# set interface ge-0/0/1.0
user@P3# set interface ge-0/0/2.0
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols ospf`, `show protocols mpls`, `show protocols rsvp`, and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@P3# show interfaces
interfaces {
    ge-0/0/1 {
        unit 0 {
            family inet {
```

```
                address 192.168.14.4/24;
            }
            family mpls;
        }
    }
    ge-0/0/2 {
        unit 0 {
            family inet {
                address 192.168.45.4/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.0.4/32;
            }
        }
    }
}
```

```
user@P3# show protocols ospf
traffic-engineering;
area 0.0.0.0 {
    interface ge-0/0/1.0;
    interface ge-0/0/2.0;
```

```
    interface lo0.0;
}
```

```
user@P3# show protocols mpls
interface ge-0/0/1.0;
interface ge-0/0/2.0;
```

```
user@P3# show protocols rsvp
interface ge-0/0/1.0;
interface ge-0/0/2.0;
```

```
user@P3# show routing-options
srlg {
    srlg-a {
        srlg-value 101;
        srlg-cost 10;
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Configuring Device P4

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see the Junos OS CLI User Guide.

To configure P4:

1. Configure the device interfaces.

```
[edit interfaces]
user@P4# set ge-0/0/1 unit 0 family inet address 192.168.45.5/24
user@P4# set ge-0/0/1 unit 0 family mpls
user@P4# set ge-0/0/2 unit 0 family inet address 192.168.56.5/24
user@P4# set ge-0/0/2 unit 0 family mpls
user@P4# set ge-0/0/3 unit 0 family inet address 192.168.25.5/24
```

```
user@P4# set ge-0/0/3 unit 0 family mpls
user@P4# set lo0 unit 0 family inet address 10.255.0.5/32
```

2. Configure OSPF on the interfaces.

```
[edit protocols ospf]
user@P4# set traffic-engineering
user@P4# set area 0.0.0.0 interface ge-0/0/1.0
user@P4# set area 0.0.0.0 interface ge-0/0/2.0
user@P4# set area 0.0.0.0 interface ge-0/0/3.0
user@P4# set area 0.0.0.0 interface lo0.0
```

3. Configure the SRLG definitions.

```
[edit routing-options]
user@P4# set routing-options srlg srlg-a srlg-value 101
user@P4# set routing-options srlg srlg-a srlg-cost 10
```

4. Configure MPLS on the interfaces.

```
[edit protocols mpls]
user@P4# set interface ge-0/0/1.0
user@P4# set interface ge-0/0/2.0
user@P4# set interface ge-0/0/3.0
```

5. Enable RSVP on the interfaces.

```
[edit protocols rsvp]
user@P4# set interface ge-0/0/1.0
user@P4# set interface ge-0/0/2.0
user@P4# set interface ge-0/0/3.0
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols ospf`, `show protocols mpls`, `show protocols rsvp`, and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@P4# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            address 192.168.45.5/24;
        }
        family mpls;
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 192.168.56.5/24;
        }
        family mpls;
    }
}
ge-0/0/3 {
    unit 0 {
        family inet {
            address 192.168.25.5/24;
        }
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.255.0.5/32;
        }
    }
}
```

```
user@P4# show protocols ospf
traffic-engineering;
```

```
area 0.0.0.0 {
    interface ge-0/0/1.0;
    interface ge-0/0/2.0;
    interface ge-0/0/3.0;
    interface lo0.0;
}
```

```
user@P4# show protocols mpls
interface ge-0/0/1.0;
interface ge-0/0/2.0;
interface ge-0/0/3.0;
```

```
user@P4# show protocols rsvp
interface ge-0/0/1.0;
interface ge-0/0/2.0;
interface ge-0/0/3.0;
```

```
user@P4# show routing-options
srlg {
    srlg-a {
        srlg-value 101;
        srlg-cost 10;
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Configuring Device P5

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see the Junos OS CLI User Guide.

To configure P5:

1. Configure the device interfaces.

```
[edit interfaces]
user@P5# set ge-0/0/1 unit 0 family inet address 192.168.56.6/24
user@P5# set ge-0/0/1 unit 0 family mpls
user@P5# set ge-0/0/2 unit 0 family inet address 192.168.67.6/24
user@P5# set ge-0/0/2 unit 0 family mpls
user@P5# set lo0 unit 0 family inet address 10.255.0.6/32
```

2. Configure OSPF on the interfaces.

```
[edit protocols ospf]
user@P5# set traffic-engineering
user@P5# set area 0.0.0.0 interface ge-0/0/1.0
user@P5# set area 0.0.0.0 interface ge-0/0/2.0
user@P5# set area 0.0.0.0 interface lo0.0
```

3. Configure the SRLG definitions.

```
[edit routing-options]
user@P5# set routing-options srlg srlg-a srlg-value 101
user@P5# set routing-options srlg srlg-a srlg-cost 10
```

4. Configure MPLS on the interfaces.

```
[edit protocols mpls]
user@P5# set interface ge-0/0/1.0
user@P5# set interface ge-0/0/2.0
```

5. Enable RSVP on the interfaces.

```
[edit protocols rsvp]
user@P5# set interface ge-0/0/1.0
user@P5# set interface ge-0/0/2.0
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols ospf`, `show protocols mpls`, `show protocols rsvp`, and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@P5# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            address 192.168.56.6/24;
        }
        family mpls;
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 192.168.67.6/24;
        }
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.255.0.6/32;
        }
    }
}
```

```
user@P5# show protocols ospf
traffic-engineering;
area 0.0.0.0 {
    interface ge-0/0/1.0;
    interface ge-0/0/2.0;
```

```
    interface lo0.0;
}
```

```
user@P5# show protocols mpls
interface ge-0/0/1.0;
interface ge-0/0/2.0;
```

```
user@P5# show protocols rsvp
interface ge-0/0/1.0;
interface ge-0/0/2.0;
```

```
user@P5# show routing-options
srlg {
    srlg-a {
        srlg-value 101;
        srlg-cost 10;
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Configuring Device PE2

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see the Junos OS CLI User Guide.

To configure PE2:

1. Configure the device interfaces.

```
[edit interfaces]
user@PE2# set ge-0/0/1 unit 0 family inet address 192.168.27.7/24
user@PE2# set ge-0/0/1 unit 0 family mpls
user@PE2# set ge-0/0/2 unit 0 family inet address 192.168.37.7/24
user@PE2# set ge-0/0/2 unit 0 family mpls
user@PE2# set ge-0/0/3 unit 0 family inet address 192.168.67.7/24
```

```
user@PE2# set ge-0/0/3 unit 0 family mpls
user@PE2# set lo0 unit 0 family inet address 10.255.0.7/32
```

2. Configure OSPF on the interfaces.

```
[edit protocols ospf]
user@PE2# set traffic-engineering
user@PE2# set area 0.0.0.0 interface ge-0/0/1.0
user@PE2# set area 0.0.0.0 interface ge-0/0/2.0
user@PE2# set area 0.0.0.0 interface ge-0/0/3.0
user@PE2# set area 0.0.0.0 interface lo0.0
```

3. Configure the SRLG definitions.

```
[edit routing-options]
user@PE2# set routing-options srlg srlg-a srlg-value 101
user@PE2# set routing-options srlg srlg-a srlg-cost 10
```

4. Configure MPLS on the interfaces.

```
[edit protocols mpls]
user@PE2# set interface ge-0/0/1.0
user@PE2# set interface ge-0/0/2.0
user@PE2# set interface ge-0/0/3.0
```

5. Enable RSVP on the interfaces.

```
[edit protocols rsvp]
user@PE2# set interface ge-0/0/1.0
user@PE2# set interface ge-0/0/2.0
user@PE2# set interface ge-0/0/3.0
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols ospf`, `show protocols mpls`, `show protocols rsvp`, and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@PE2# show interfaces
interfaces {
    ge-0/0/1 {
        unit 0 {
            family inet {
                address 192.168.27.7/24;
            }
            family mpls;
        }
    }
    ge-0/0/2 {
        unit 0 {
            family inet {
                address 192.168.37.7/24;
            }
            family mpls;
        }
    }
    ge-0/0/3 {
        unit 0 {
            family inet {
                address 192.168.67.7/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.0.7/32;
            }
        }
```

```
    }
}
```

```
user@PE2# show protocols ospf
traffic-engineering;
area 0.0.0.0 {
    interface ge-0/0/1.0;
    interface ge-0/0/2.0;
    interface ge-0/0/3.0;
    interface lo0.0;
}
```

```
user@PE2# show protocols mpls
interface ge-0/0/1.0;
interface ge-0/0/2.0;
interface ge-0/0/3.0;
```

```
user@PE2# show protocols rsvp
interface ge-0/0/1.0;
interface ge-0/0/2.0;
interface ge-0/0/3.0;
```

```
user@PE2# show routing-options
srlg {
    srlg-a {
        srlg-value 101;
        srlg-cost 10;
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

IN THIS SECTION

- Verifying the SRLG Cost Is Added to the TE Link | 358

Confirm that the configuration is working properly.

Verifying the SRLG Cost Is Added to the TE Link

Purpose

Verify that the TE link is excluded if it belongs to the SRLG of the protected link when `link-protection` is configured with `exclude-srlg`. Issue the `show ted link detail` and `show rsvp session extensive bypass` commands on device P1.

Action

```
user@P1> show ted link detail

...
10.255.0.2->192.168.27.7-1, Local: 192.168.27.2, Remote: 0.0.0.0
  Local interface index: 0, Remote interface index: 0
  LocalPath: 0, Metric: 1, StaticBW: 1000Mbps, AvailBW: 1000Mbps
      Color: 0 <none>
  SRLGs: srlg-a
  localBW [0] 0bps  [1] 0bps  [2] 0bps  [3] 0bps
  localBW [4] 0bps  [5] 0bps  [6] 0bps  [7] 0bps
[...]
10.255.0.3->192.168.37.7-1, Local: 192.168.37.3, Remote: 0.0.0.0
  Local interface index: 0, Remote interface index: 0
  LocalPath: 0, Metric: 1, StaticBW: 1000Mbps, AvailBW: 1000Mbps
      Color: 0 <none>
  SRLGs: srlg-a
  localBW [0] 0bps  [1] 0bps  [2] 0bps  [3] 0bps
```

```
  localBW [4] 0bps  [5] 0bps  [6] 0bps  [7] 0bps
...
```

```
user@P1> show rsvp session extensive bypass

 Ingress RSVP: 1 sessions

10.255.0.7
  From: 10.255.0.2, LSPstate: Up, ActiveRoute: 0
  LSPname: Bypass->192.168.27.7
  LSPtype: Static Configured
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 299776
  Resv style: 1 SE, Label in: -, Label out: 299776
  Time left:    -, Since: Fri Oct 21 13:19:21 2011
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 52081 protocol 0
  Type: Bypass LSP
    Number of data route tunnel through: 1
    Number of RSVP session tunnel through: 0
  PATH rcvfrom: localclient
  Adspec: sent MTU 1500
  Path MTU: received 1500
  PATH sentto: 192.168.25.5 (ge-0/0/4.0) 63 pkts
  RESV rcvfrom: 192.168.25.5 (ge-0/0/4.0) 63 pkts
  Explct route: 192.168.25.5 192.168.56.6 192.168.67.7
  Record route: <self> 192.168.25.5 192.168.56.6 192.168.67.7
Total 1 displayed, Up 1, Down 0
```

Meaning

The shortest path for the bypass protecting the link P1>PE2 would have been P1>P2>PE2. Because the links P1>PE2 and P2>PE2 both belong to SRLG `srlg-a`, the link P2>PE2 is rejected for CSPF consideration due to the `exclude-srlg` constraint. Therefore, the computed path for the bypass becomes P1>P4>P5>PE2.

RELATED DOCUMENTATION

Basic MPLS Configuration | 46

CHAPTER 6

Protecting MPLS Traffic

IN THIS CHAPTER

- Node and Path Protection for MPLS LSPs | 360
- Link Protection for MPLS LSPs | 459

Node and Path Protection for MPLS LSPs

IN THIS SECTION

- MPLS and Traffic Protection | 360
- Node-Link Protection Overview | 361
- Path Protection Overview | 363
- Configuring Path Protection in an MPLS Network (CLI Procedure) | 364
- Preventing Use of a Path That Previously Failed | 367
- Configuring MPLS Inter-AS Link-Node Protection with Labeled BGP | 368
- Configuring Egress Protection Service Mirroring for BGP Signaled Layer 2 Services | 389
- Example: Configuring MPLS Egress Protection Service Mirroring for BGP Signaled Layer 2 Services | 394
- Example: Configuring Layer 3 VPN Egress Protection with PLR as Protector | 419
- Understanding MPLS and Path Protection on EX Series Switches | 453
- Verifying Path Protection in an MPLS Network | 454

MPLS and Traffic Protection

Typically, when an LSP fails, the router immediately upstream from the failure signals the outage to the ingress router. The ingress router calculates a new path to the egress router, establishes the new LSP, and then directs the traffic from the failed path to the new path. This rerouting process can be time-

consuming and prone to failure. For example, the outage signals to the ingress router might get lost, or the new path might take too long to come up, resulting in significant packet drops. The Junos OS provides several complementary mechanisms for protecting against LSP failures:

- Standby secondary paths—You can configure primary and secondary paths. You configure secondary paths with the `standby` statement. To activate traffic protection, you need to configure these standby paths only on the ingress router. If the primary path fails, the ingress router immediately reroutes traffic from the failed path to the standby path, thereby eliminating the need to calculate a new route and signal a new path. For information about configuring standby LSPs, see "Configuring Hot Standby of Secondary Paths for LSPs" on page 689.

- Fast reroute—You configure fast reroute on an LSP to minimize the effect of a failure in the LSP. Fast reroute enables a router upstream from the failure to route around the failure quickly to the router downstream of the failure. The upstream router then signals the outage to the ingress router, thereby maintaining connectivity before a new LSP is established. For a detailed overview of fast reroute, see "Fast Reroute Overview" on page 577. For information about configuring fast reroute, see "Configuring Fast Reroute" on page 580.

- Link protection—You can configure link protection to help ensure that traffic traversing a specific interface from one router to another can continue to reach its destination in the event that this interface fails. When link protection is configured for an interface and configured for an LSP that traverses this interface, a bypass LSP is created that handles this traffic if the interface fails. The bypass LSP uses a different interface and path to reach the same destination. For information about configuring link protection, see "Configuring Link Protection on Interfaces Used by LSPs" on page 468.

When standby secondary path, and fast reroute or link protection are configured on an LSP, full traffic protection is enabled. When a failure occurs in an LSP, the router upstream from the failure routes traffic around the failure and notifies the ingress router of the failure. This rerouting keeps the traffic flowing while waiting for the notification to be processed at the ingress router. After receiving the failure notification, the ingress router immediately reroutes the traffic from the patched primary path to the more optimal standby path.

Fast reroute and link protection provide a similar type of traffic protection. Both features provide a quick transfer service and employ a similar design. Fast reroute and link protection are both described in RFC 4090, *Fast Reroute Extensions to RSVP-TE for LSP Tunnels*. However, you need to configure only one or the other. Although you can configure both, there is little, if any, benefit in doing so.

Node-Link Protection Overview

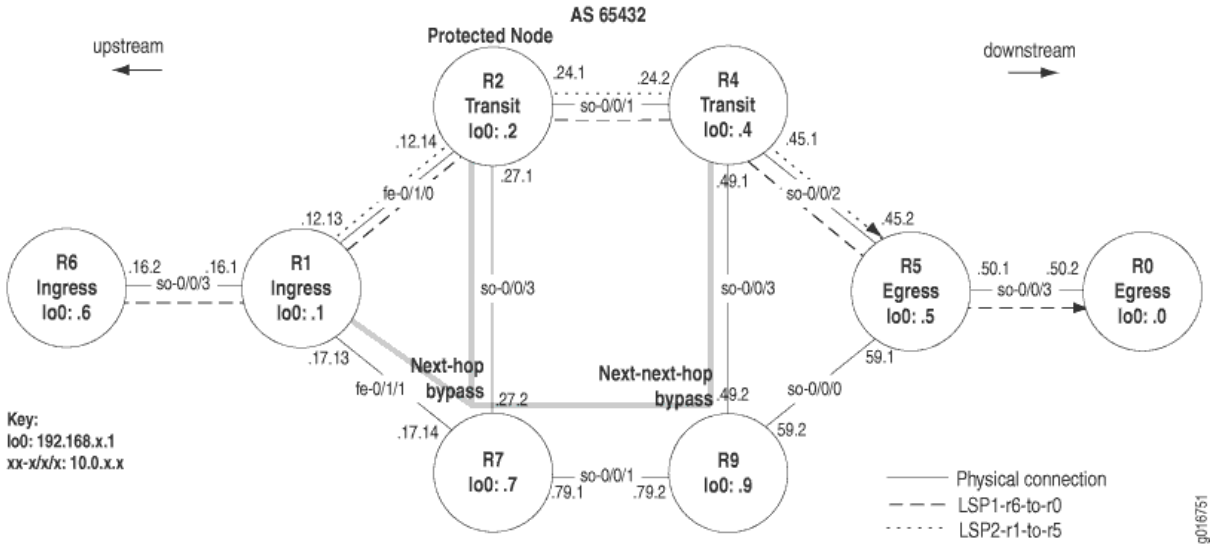
Node-link protection (many-to-one or facility backup) extends the capabilities of link protection and provides slightly different protection from fast reroute. While link protection is useful for selecting an alternate path to the same router when a specific link fails, and fast reroute protects interfaces or nodes along the entire path of an LSP, node-link protection establishes a bypass path that avoids a particular node in the LSP path.

When you enable node-link protection for an LSP, you must also enable link protection on all RSVP interfaces in the path. Once enabled, the following types of bypass paths are established:

- Next-hop bypass LSP—Provides an alternate route for an LSP to reach a neighboring router. This type of bypass path is established when you enable either node-link protection or link protection.
- Next-next-hop bypass LSP—Provides an alternate route for an LSP through a neighboring router en route to the destination router. This type of bypass path is established exclusively when node-link protection is configured.

Figure 12 on page 362 illustrates the example MPLS network topology used in this topic. The example network uses OSPF as the interior gateway protocol (IGP) and a policy to create traffic.

Figure 12: Node-Link Protection



The MPLS network in Figure 12 on page 362 illustrates a router-only network that consists of unidirectional LSPs between **R1** and **R5**, (**lsp2-r1-to-r5**) and between **R6** and **R0** (**lsp1-r6-to-r0**). Both LSPs have strict paths configured that go through interface **fe-0/1/0**.

In the network shown in Figure 12 on page 362, both types of bypass paths are preestablished around the protected node (**R2**)**.** A next-hop bypass path avoids interface **fe-0/1/0** by going through **R7**, and a next-next-hop bypass path avoids **R2** altogether by going through **R7** and **R9** to **R4**. Both bypass paths are shared by all protected LSPs traversing the failed link or node (many LSPs protected by one bypass path).

Node-link protection (many-to-one or facility backup) allows a router immediately upstream from a node failure to use an alternate node to forward traffic to its downstream neighbor. This is accomplished by preestablishing a bypass path that is shared by all protected LSPs traversing the failed link.

When an outage occurs, the router immediately upstream from the outage switches protected traffic to the bypass node, and then signals the failure to the ingress router. Like fast reroute, node-link protection provides local repair, restoring connectivity faster than the ingress router can establish a standby secondary path or signal a new primary LSP.

Node-link protection is appropriate in the following situations:

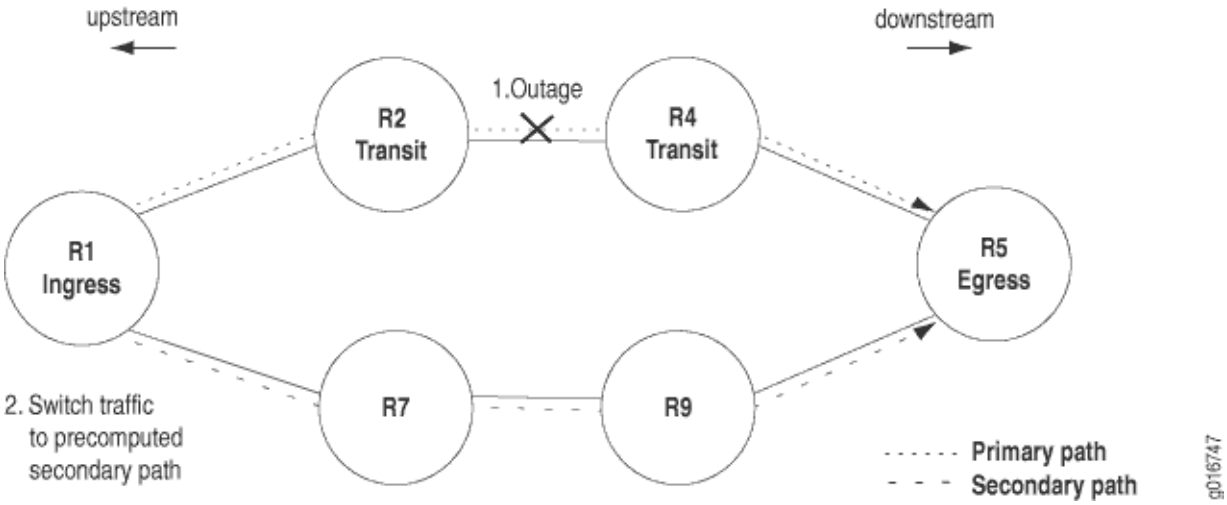
- Protection of the downstream link and node is required.
- The number of LSPs to be protected is large.
- Satisfying path selection criteria (priority, bandwidth, and link coloring) for bypass paths is less critical.
- Control at the granularity of individual LSPs is not required.

Path Protection Overview

The main advantages of path protection are control over where the traffic goes after a failure and minimum packet loss when combined with fast reroute (one-to-one backup or link protection). Path protection is the configuration, within a label-switched path (LSP), of two types of paths: a primary path, used in normal operations, and a secondary path used when the primary fails, as shown in Figure 13 on page 363.

In Figure 13 on page 363, an MPLS network consisting of eight routers has a primary path between **R1** and **R5** which is protected by the secondary path between **R1** and **R5**. When a failure is detected, such as an interface down event, an Resource Reservation Protocol (RSVP) error message is sent to the ingress router which switches traffic to the secondary path, maintaining traffic flow.

Figure 13: Path Protection



If the secondary path is pre-signaled or on standby, recovery time from a failure is faster than if the secondary path is not pre-signaled. When the secondary path is not pre-signaled a call-setup delay occurs during which the new physical path for the LSP is established, extending the recovery time. If the failure in the primary path is corrected, and after a few minutes of hold time, the ingress router switches traffic back from the secondary path to the primary path.

Because path protection is provided by the ingress router for the entire path, there can be some disadvantages, for example, double-booking of resources and unnecessary protection of links. By protecting a single resource at a time, local protection can remedy these disadvantages.

Configuring Path Protection in an MPLS Network (CLI Procedure)

IN THIS SECTION

- Configuring the Primary Path | **365**
- Configuring the Secondary Path | **366**
- Configuring the Revert Timer | **367**

The Junos OS implementation of MPLS on EX Series switches provides path protection as a mechanism for protecting against label switched path (LSP) failures. Path protection reduces the time required to recalculate a route in case of a failure within the MPLS tunnel. You configure path protection on the ingress provider edge switch in your MPLS network. You do not configure the egress provider edge switch or the provider switches for path protection. You can explicitly specify which provider switches are used for the primary and secondary paths, or you can let the software calculate the paths automatically.

Before you configure path protection, be sure you have:

- Configured an ingress provider edge switch and an egress provider edge switch. See "Configuring MPLS on Provider Edge Switches Using IP-Over-MPLS" on page 83 or "Configuring MPLS on Provider Edge EX8200 and EX4500 Switches Using Circuit Cross-Connect" on page 89.
- Configured at least one provider (transit) switch. See "Configuring MPLS on EX8200 and EX4500 Provider Switches" on page 93.
- Verified the configuration of your MPLS network.

To configure path protection, complete the following tasks on the ingress provider edge switch:

Configuring the Primary Path

The `primary` statement creates the primary path, which is the LSP's preferred path. The `secondary` statement creates an alternative path if the primary path can no longer reach the egress provider edge switch.

In the tasks described in this topic, the ***lsp-name*** has already been configured on the ingress provider edge switch as **lsp_to_240** and the loopback interface address on the remote provider edge switch has already been configured as **127.0.0.8**.

When the software switches from the primary to a secondary path, it continuously attempts to revert to the primary path, switching back to it when it is again reachable but no sooner than the time specified in the `revert-timer` statement.

You can configure zero primary paths or one primary path. If you do not configure a primary path, the first secondary path (if a secondary path has been configured) is selected as the path. If you do not specify any named paths, or if the path that you specify is empty, the software makes all routing decisions necessary for the packets to reach the egress provider edge switch.

To configure a primary path:

1. Create the primary path for the LSP:

```
[edit protocols mpls                          label-switched-path lsp_to_240 to 127.0.0.8]
user@switch# set primary primary_path_lsp_to_240
```

2. Configure an explicit route for the primary path by specifying the IP address of the loopback interface or the switch IP address or hostname of each switch used in the MPLS tunnel. You can specify the link types as either **strict** or **loose** in each `path` statement. If the link type is **strict**, the LSP must go to the next address specified in the `path` statement without traversing other switches. If the link type is **loose**, the LSP can traverse through other switches before reaching this switch. This configuration uses the default **strict** designation for the paths.

NOTE: You can enable path protection without specifying which provider switches are used. If you do not list the specific provider switches to be used for the MPLS tunnel, the switch calculates the route.

TIP: Do not include the ingress provider edge switch in these statements. List the IP address of the loopback interface or switch address or hostname of all other switch hops in sequence, ending with the egress provider edge switch.

```
[edit protocols mpls label-switched-path lsp_to_240 to 127.0.0.8]
user@switch# set path primary_path_lsp_to_240 127.0.0.2
user@switch# set path primary_path_lsp_to_240 127.0.0.3
user@switch# set path primary_path_lsp_to_240 127.0.0.8
```

Configuring the Secondary Path

You can configure zero or more secondary paths. All secondary paths are equal, and the software tries them in the order that they are listed in the configuration. The software does not attempt to switch among secondary paths. If the first secondary path in the configuration is not available, the next one is tried, as so on. To create a set of equal paths, specify secondary paths without specifying a primary path. If you do not specify any named paths, or if the path that you specify is empty, the software makes all routing decisions necessary to reach the egress provider edge switch.

To configure the secondary path:

1. Create a secondary path for the LSP:

```
[edit protocols mpls label-switched-path lsp_to_240 to 127.0.0.8]
user@switch# set secondary secondary_path_lsp_to_240
standby
```

2. Configure an explicit route for the secondary path by specifying the IP address of the loopback interface or the switch IP address or hostname of each switch used in the MPLS tunnel. You can specify the link types as either **strict** or **loose** in each `path` statement. This configuration uses the default **strict** designation for the paths.

TIP: Do not include the ingress provider edge switch in these statements. List the IP address of the loopback interface or switch address or hostname of all other switch hops in sequence, ending with the egress provider edge switch.

```
[edit protocols mpls label-switched-path lsp_to_240 to 127.0.0.8]
user@switch# set path secondary_path_lsp_to_240 127.0.0.4
user@switch# set path primary_path_lsp_to_240 127.0.0.8
```

Configuring the Revert Timer

For LSPs configured with both primary and secondary paths, you can optionally configure a revert timer. If the primary path goes down and traffic is switched to the secondary path, the revert timer specifies the amount of time (in seconds) that the LSP must wait before it can revert traffic back to the primary path. If the primary path experiences any connectivity problems or stability problems during this time, the timer is restarted.

TIP: If you do not explicitly configure the revert timer, it is set by default to 60 seconds.

To configure the revert timer for LSPs configured with primary and secondary paths:

- For all LSPs on the switch:

```
[edit protocols mpls]
user@switch#  set revert-timer 120
```

- For a specific LSP on the switch:

```
[edit protocols mpls label-switched-path]
user@switch#  set lsp_to_240 revert-timer 120
```

Preventing Use of a Path That Previously Failed

If you configure an alternate path through the network in case the active path fails, you may not want traffic to revert back to the failed path, even if it is no longer failing. When you configure a primary path, the traffic switches over to the secondary path during a failure, and reverts back to the primary path when it returns.

At times, switching traffic back to a primary path that has previously failed may not be a particularly sound idea. In this case, only configure secondary paths, resulting in the next configured secondary path establishing when the first secondary path fails. Later, if the first secondary path becomes operational, the Junos OS will not revert to it, but will continue using the second secondary path.

Configuring MPLS Inter-AS Link-Node Protection with Labeled BGP

IN THIS SECTION

- Understanding MPLS Inter-AS Link Protection | **368**
- Example: Configuring MPLS Inter-AS Link-Node Protection | **370**

Understanding MPLS Inter-AS Link Protection

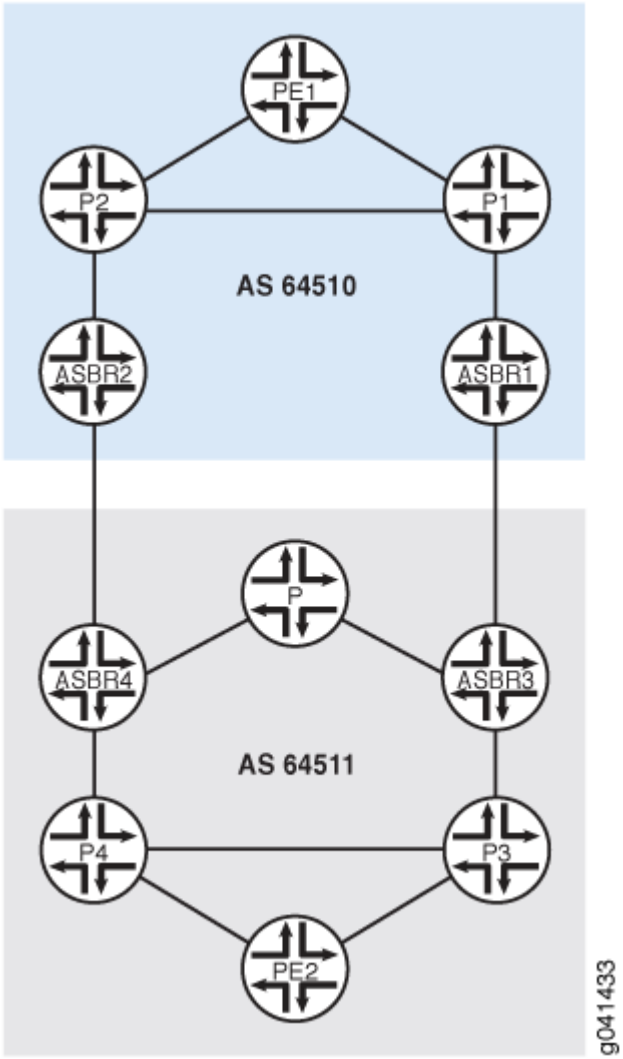
Link protection is essential in an MPLS network to ensure traffic restoration in case of an interface failure. The ingress router chooses an alternate link through another interface to send traffic to its destination.

In Figure 3, autonomous system border routers (ASBRs) run external BGP (EBGP) to ASBRs in another autonomous system (AS) to exchange labels for /32 IPv4 routes. Inside the ASs, internal BGP (IBGP) propagates the routes to provider edge (PE) devices. If the link from Device ASBR3 to Device ASBR1 goes down, until Device ASBR3 reinstalls the new next hop, all traffic going toward AS 64510 from AS 64511 through the ASBR3-ASBR1 link is dropped. A fast traffic restoration can be achieved if Device ASBR3 preprograms a backup path either through Device ASBR4 or through a direct path to Device ASBR2 if one exists (not shown in the diagram). This assumes that Device ASBR3 learns a loop-free MPLS path for routes that need to protected either through IBGP or EBGP.

This solution does not handle a failure on Device ASBR3 for traffic going toward AS 64511 from AS 64510 through the ASBR3-ASBR1 link. This solution is limited to downstream inter-AS link-node protection with labeled BGP. This solution does not support service restoration between provider (P) and ASBR routers when there is an ASBR failure. For example, this solution does not handle a failure on the P3-ASBR3 link.

This supported functionality is similar to BGP multipath, except only one next hop is used for active forwarding, and a second path is in protected mode.

Figure 14: MPLS Inter-AS Link-Node Protection Conceptual Topology



In an MPLS inter-AS environment, link protection can be enabled when `labeled-unicast` is used to send traffic between ASs. Hence, MPLS inter-AS link protection is configured on the link between two routers in different ASs.

To configure link protection on an interface, use the `protection` statement at the `[edit protocols bgp group` *`group-name`* `family inet labeled-unicast]` hierarchy level:

```
protocols {
    bgp {
        group test1 {
            type external;
            local-address 192.168.1.2;
            family inet {
```

```
                labeled-unicast {
                    protection;
                }
            }
        }
    }
}
```

NOTE: MPLS inter-AS link protection is supported only with `labeled-unicast` and external peers in a master routing instance.

The link on which protection is configured is known as the protection path. A protection path is selected only after the best path selection and is not selected in the following cases:

- The best path is a non-BGP path.
- Multiple next hops are active, as in BGP multipath.

Example: Configuring MPLS Inter-AS Link-Node Protection

IN THIS SECTION

- Requirements | 370
- Overview | 370
- Configuration | 373
- Verification | 386

This example shows how to configure tail-end protection in an inter-AS deployment with Layer 3 VPNs.

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

IN THIS SECTION

- Topology | 372

In Figure 15 on page 372. autonomous system border routers (ASBRs) run external BGP (EBGP) to ASBRs in another autonomous system (AS) to exchange labels for /32 IPv4 routes. Inside the ASs, internal BGP (IBGP) propagates the routes to provider edge (PE) devices.

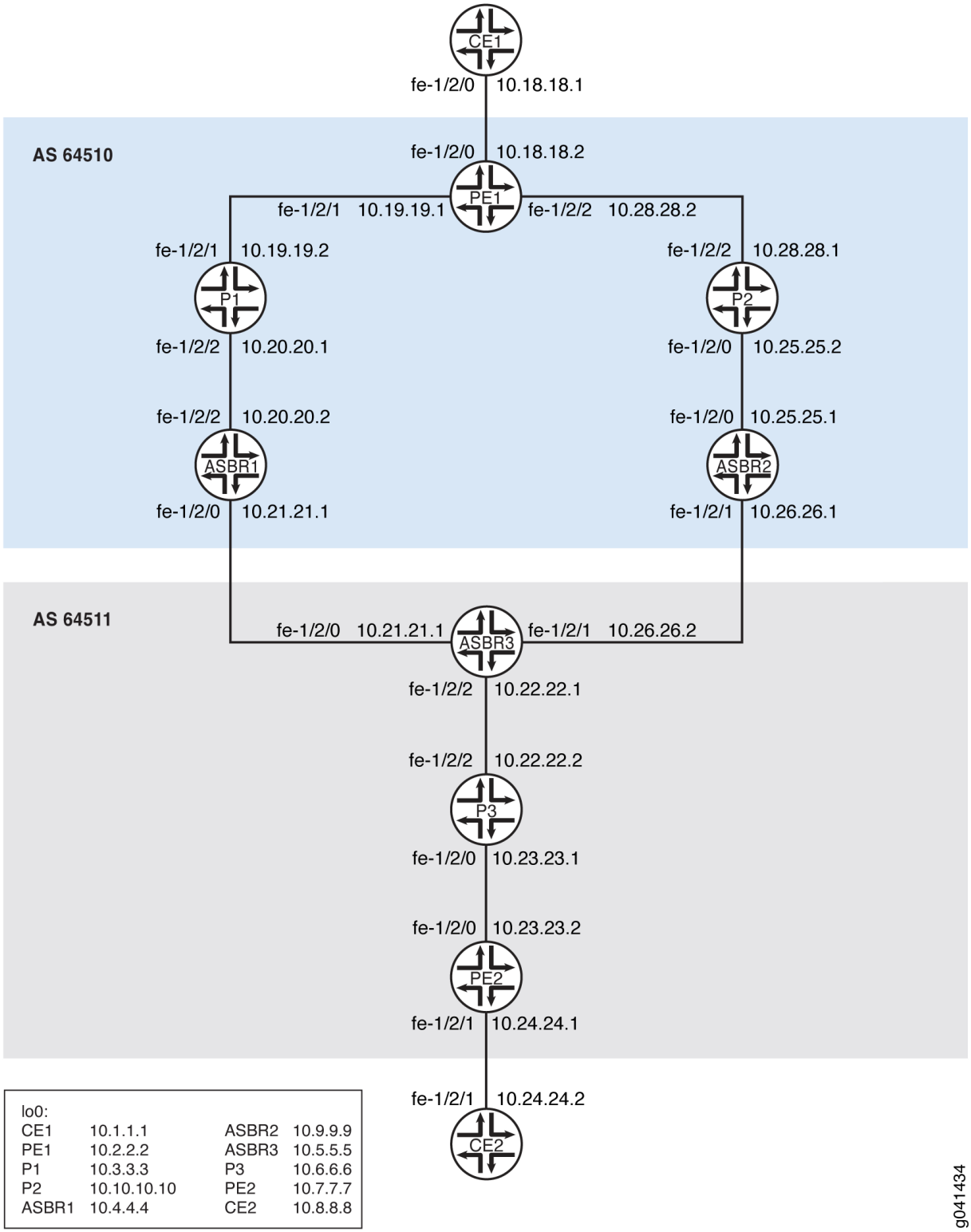
If the link from Device ASBR3 to Device ASBR1 goes down, until ASBR3 reinstalls the new next hop, all traffic going toward AS 64510 from AS 64511 through the ASBR3-ASBR1 link is dropped.

This example shows how to achieve fast traffic restoration by configuring Device ASBR3 to preprogram a backup path through Device ASBR2.

NOTE: This solution does not handle the Device P3 to Device ASBR3 failure. Nor does it handle a failure on Device ASBR3 for traffic going toward AS 645111 from AS 64510 through the ASBR3-ASBR1 link. This traffic is dropped.

Topology

Figure 15: MPLS Inter-AS Link-Node Protection Example Topology



Configuration

IN THIS SECTION

- CLI Quick Configuration | 373
- Procedure | 380

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the `[edit]` hierarchy level.

Device ASBR1

```
set interfaces fe-1/2/2 unit 0 family inet address 10.20.20.2/30
set interfaces fe-1/2/2 unit 0 family mpls
set interfaces fe-1/2/0 unit 0 family inet address 10.21.21.1/30
set interfaces fe-1/2/0 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.4.4.4/32
set protocols rsvp interface fe-1/2/2.0
set protocols rsvp interface lo0.0
set protocols mpls traffic-engineering bgp-igp-both-ribs
set protocols mpls label-switched-path To_PE1 to 10.2.2.2
set protocols mpls interface fe-1/2/2.0
set protocols mpls interface fe-1/2/0.0
set protocols mpls interface lo0.0
set protocols bgp group To-PE1 type internal
set protocols bgp group To-PE1 local-address 10.4.4.4
set protocols bgp group To-PE1 family inet unicast
set protocols bgp group To-PE1 family inet labeled-unicast
set protocols bgp group To-PE1 export next-hop-self
set protocols bgp group To-PE1 neighbor 10.2.2.2 family inet labeled-unicast
set protocols bgp group To-ASBR3 type external
set protocols bgp group To-ASBR3 family inet labeled-unicast
set protocols bgp group To-ASBR3 export To-ASBR3
set protocols bgp group To-ASBR3 neighbor 10.21.21.2 peer-as 64511
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface fe-1/2/2.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

```
set policy-options policy-statement To-ASBR3 term 1 from route-filter 10.2.2.2/32 exact
set policy-options policy-statement To-ASBR3 term 1 then accept
set policy-options policy-statement To-ASBR3 term 2 then reject
set policy-options policy-statement next-hop-self then next-hop self
set routing-options autonomous-system 64510
```

Device ASBR2

```
set interfaces fe-1/2/0 unit 0 description to-P2
set interfaces fe-1/2/0 unit 0 family inet address 10.25.25.1/30
set interfaces fe-1/2/0 unit 0 family mpls
set interfaces fe-1/2/1 unit 0 description to-ASBR3
set interfaces fe-1/2/1 unit 0 family inet address 10.26.26.1/30
set interfaces fe-1/2/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.9.9.9/32
set protocols rsvp interface fe-1/2/0.0
set protocols rsvp interface lo0.0
set protocols mpls traffic-engineering bgp-igp-both-ribs
set protocols mpls label-switched-path To_PE1 to 10.2.2.2
set protocols mpls interface fe-1/2/0.0
set protocols mpls interface fe-1/2/1.0
set protocols mpls interface lo0.0
set protocols bgp group To-PE1 type internal
set protocols bgp group To-PE1 local-address 10.9.9.9
set protocols bgp group To-PE1 family inet unicast
set protocols bgp group To-PE1 family inet labeled-unicast
set protocols bgp group To-PE1 export next-hop-self
set protocols bgp group To-PE1 neighbor 10.2.2.2 family inet labeled-unicast
set protocols bgp group To-ASBR3 type external
set protocols bgp group To-ASBR3 family inet labeled-unicast
set protocols bgp group To-ASBR3 export To-ASBR3
set protocols bgp group To-ASBR3 neighbor 10.26.26.2 peer-as 64511
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface fe-1/2/0.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set policy-options policy-statement To-ASBR3 term 1 from route-filter 10.2.2.2/32 exact
set policy-options policy-statement To-ASBR3 term 1 then accept
set policy-options policy-statement To-ASBR3 term 2 then reject
set policy-options policy-statement next-hop-self then next-hop self
set routing-options autonomous-system 64510
```

Device ASBR3

```
set interfaces fe-1/2/0 unit 0 description to-ASBR1
set interfaces fe-1/2/0 unit 0 family inet address 10.21.21.2/30
set interfaces fe-1/2/0 unit 0 family mpls
set interfaces fe-1/2/2 unit 0 description to-P3
set interfaces fe-1/2/2 unit 0 family inet address 10.22.22.1/30
set interfaces fe-1/2/2 unit 0 family mpls
set interfaces fe-1/2/1 unit 0 description to-ASBR2
set interfaces fe-1/2/1 unit 0 family inet address 10.26.26.2/30
set interfaces fe-1/2/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.5.5.5/32
set protocols rsvp interface fe-1/2/2.0
set protocols rsvp interface lo0.0
set protocols rsvp interface fe-1/2/0.0
set protocols rsvp interface fe-1/2/1.0
set protocols mpls traffic-engineering bgp-igp-both-ribs
set protocols mpls label-switched-path To_PE2 to 10.7.7.7
set protocols mpls interface lo0.0
set protocols mpls interface fe-1/2/0.0
set protocols mpls interface fe-1/2/2.0
set protocols mpls interface fe-1/2/1.0
set protocols bgp group To-PE2 type internal
set protocols bgp group To-PE2 local-address 10.5.5.5
set protocols bgp group To-PE2 family inet unicast
set protocols bgp group To-PE2 export next-hop-self
set protocols bgp group To-PE2 neighbor 10.7.7.7 family inet labeled-unicast
set protocols bgp group To-ASBR1 type external
set protocols bgp group To-ASBR1 family inet labeled-unicast protection
set protocols bgp group To-ASBR1 family inet labeled-unicast per-prefix-label
set protocols bgp group To-ASBR1 export To-ASBR1
set protocols bgp group To-ASBR1 neighbor 10.21.21.1 peer-as 64510
set protocols bgp group To-ASBR2 type external
set protocols bgp group To-ASBR2 family inet labeled-unicast protection
set protocols bgp group To-ASBR2 family inet labeled-unicast per-prefix-label
set protocols bgp group To-ASBR2 export To-ASBR2
set protocols bgp group To-ASBR2 neighbor 10.26.26.1 peer-as 64510
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface fe-1/2/2.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/1.0
set policy-options policy-statement To-ASBR1 term 1 from route-filter 10.7.7.7/32 exact
```

```
set policy-options policy-statement To-ASBR1 term 1 then accept
set policy-options policy-statement To-ASBR1 term 2 then reject
set policy-options policy-statement To-ASBR2 term 1 from route-filter 10.7.7.7/32 exact
set policy-options policy-statement To-ASBR2 term 1 then accept
set policy-options policy-statement To-ASBR2 term 2 then reject
set policy-options policy-statement next-hop-self then next-hop self
set routing-options autonomous-system 64511
```

Device CE1

```
set interfaces fe-1/2/0 unit 0 family inet address 10.18.18.1/30
set interfaces lo0 unit 0 family inet address 10.1.1.1/32
set protocols ospf area 0.0.0.2 interface fe-1/2/0.0
set protocols ospf area 0.0.0.2 interface lo0.0 passive
```

Device CE2

```
set interfaces fe-1/2/1 unit 0 family inet address 10.24.24.2/30
set interfaces lo0 unit 0 family inet address 10.8.8.8/32
set protocols bgp group To_PE2 neighbor 10.24.24.1 export myroutes
set protocols bgp group To_PE2 neighbor 10.24.24.1 peer-as 64511
set policy-options policy-statement myroutes from protocol direct
set policy-options policy-statement myroutes then accept
set routing-options autonomous-system 64509
```

Device P1

```
set interfaces fe-1/2/1 unit 0 family inet address 10.19.19.2/30
set interfaces fe-1/2/1 unit 0 family mpls
set interfaces fe-1/2/2 unit 0 family inet address 10.20.20.1/30
set interfaces fe-1/2/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.3.3.3/32
set protocols rsvp interface fe-1/2/1.0
set protocols rsvp interface fe-1/2/2.0
set protocols rsvp interface lo0.0
set protocols mpls interface fe-1/2/1.0
set protocols mpls interface fe-1/2/2.0
set protocols mpls interface lo0.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface fe-1/2/1.0
```

```
set protocols ospf area 0.0.0.0 interface fe-1/2/2.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

Device P2

```
set interfaces fe-1/2/0 unit 0 description to-ASBR2
set interfaces fe-1/2/0 unit 0 family inet address 10.25.25.2/30
set interfaces fe-1/2/0 unit 0 family mpls
set interfaces fe-1/2/2 unit 0 description to-PE1
set interfaces fe-1/2/2 unit 0 family inet address 10.28.28.1/30
set interfaces fe-1/2/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.10.10.10/32
set protocols rsvp interface fe-1/2/0.0
set protocols rsvp interface fe-1/2/2.0
set protocols rsvp interface lo0.0
set protocols mpls interface fe-1/2/0.0
set protocols mpls interface fe-1/2/2.0
set protocols mpls interface lo0.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface fe-1/2/0.0
set protocols ospf area 0.0.0.0 interface fe-1/2/2.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

Device P3

```
set interfaces fe-1/2/2 unit 0 family inet address 10.22.22.2/30
set interfaces fe-1/2/2 unit 0 family mpls
set interfaces fe-1/2/0 unit 0 family inet address 10.23.23.1/30
set interfaces fe-1/2/0 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.6.6.6/32
set protocols rsvp interface fe-1/2/2.0
set protocols rsvp interface fe-1/2/0.0
set protocols rsvp interface lo0.0
set protocols mpls interface fe-1/2/2.0
set protocols mpls interface fe-1/2/0.0
set protocols mpls interface lo0.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface fe-1/2/2.0
set protocols ospf area 0.0.0.0 interface fe-1/2/0.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

Device PE1

```
set interfaces fe-1/2/0 unit 0 family inet address 10.18.18.2/30
set interfaces fe-1/2/1 unit 0 family inet address 10.19.19.1/30
set interfaces fe-1/2/1 unit 0 family mpls
set interfaces fe-1/2/2 unit 0 description to-P2
set interfaces fe-1/2/2 unit 0 family inet address 10.28.28.2/30
set interfaces lo0 unit 0 family inet address 10.2.2.2/32
set protocols rsvp interface fe-1/2/0.0
set protocols rsvp interface lo0.0
set protocols rsvp interface fe-1/2/2.0
set protocols mpls label-switched-path To-ASBR1 to 10.4.4.4
set protocols mpls label-switched-path To-ASBR2 to 10.9.9.9
set protocols mpls interface fe-1/2/0.0
set protocols mpls interface lo0.0
set protocols mpls interface fe-1/2/2.0
set protocols bgp group To_ASBR1 type internal
set protocols bgp group To_ASBR1 local-address 10.2.2.2
set protocols bgp group To_ASBR1 family inet labeled-unicast
set protocols bgp group To_ASBR1 neighbor 10.4.4.4 family inet labeled-unicast resolve-vpn
set protocols bgp group To_PE2 type external
set protocols bgp group To_PE2 multihop ttl 20
set protocols bgp group To_PE2 local-address 10.2.2.2
set protocols bgp group To_PE2 family inet-vpn unicast
set protocols bgp group To_PE2 neighbor 10.7.7.7 peer-as 64511
set protocols bgp group To_ASBR2 type internal
set protocols bgp group To_ASBR2 local-address 10.2.2.2
set protocols bgp group To_ASBR2 family inet labeled-unicast
set protocols bgp group To_ASBR2 neighbor 10.9.9.9 family inet labeled-unicast resolve-vpn
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface fe-1/2/0.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/2.0
set policy-options policy-statement bgp-to-ospf term 1 from protocol bgp
set policy-options policy-statement bgp-to-ospf term 1 then accept
set policy-options policy-statement bgp-to-ospf term 2 then reject
set policy-options policy-statement vpnexport term 1 from protocol ospf
set policy-options policy-statement vpnexport term 1 then community add test_comm
set policy-options policy-statement vpnexport term 1 then accept
set policy-options policy-statement vpnexport term 2 then reject
set policy-options policy-statement vpnimport term 1 from protocol bgp
set policy-options policy-statement vpnimport term 1 from community test_comm
```

```
set policy-options policy-statement vpnimport term 1 then accept
set policy-options policy-statement vpnimport term 2 then reject
set policy-options community test_comm members target:1:64510
set routing-instances vpn2CE1 instance-type vrf
set routing-instances vpn2CE1 interface fe-1/2/0.0
set routing-instances vpn2CE1 route-distinguisher 1:64510
set routing-instances vpn2CE1 vrf-import vpnimport
set routing-instances vpn2CE1 vrf-export vpnexport
set routing-instances vpn2CE1 protocols ospf export bgp-to-ospf
set routing-instances vpn2CE1 protocols ospf area 0.0.0.2 interface fe-1/2/0.0
set routing-options autonomous-system 64510
```

Device PE2

```
set interfaces fe-1/2/0 unit 0 family inet address 10.23.23.2/30
set interfaces fe-1/2/0 unit 0 family mpls
set interfaces fe-1/2/1 unit 0 family inet address 10.24.24.1/30
set interfaces lo0 unit 0 family inet address 10.7.7.7/32
set protocols rsvp interface fe-1/2/0.0
set protocols rsvp interface lo0.0
set protocols mpls label-switched-path To-ASBR3 to 10.5.5.5
set protocols mpls interface fe-1/2/0.0
set protocols mpls interface lo0.0
set protocols bgp group To_ASBR3 type internal
set protocols bgp group To_ASBR3 local-address 10.7.7.7
set protocols bgp group To_ASBR3 family inet labeled-unicast
set protocols bgp group To_ASBR3 neighbor 10.5.5.5 family inet labeled-unicast resolve-vpn
set protocols bgp group To_PE1 type external
set protocols bgp group To_PE1 multihop ttl 20
set protocols bgp group To_PE1 local-address 10.7.7.7
set protocols bgp group To_PE1 family inet-vpn unicast
set protocols bgp group To_PE1 neighbor 10.2.2.2 peer-as 64510
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface fe-1/2/0.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set policy-options policy-statement vpnexport term 1 from protocol bgp
set policy-options policy-statement vpnexport term 1 then community add test_comm
set policy-options policy-statement vpnexport term 1 then accept
set policy-options policy-statement vpnexport term 2 then reject
set policy-options policy-statement vpnimport term 1 from protocol bgp
set policy-options policy-statement vpnimport term 1 from community test_comm
set policy-options policy-statement vpnimport term 1 then accept
```

```
set policy-options policy-statement vpnimport term 2 then reject
set policy-options community test_comm members target:1:64510
set routing-instances vpn2CE2 instance-type vrf
set routing-instances vpn2CE2 interface fe-1/2/1.0
set routing-instances vpn2CE2 route-distinguisher 1:64510
set routing-instances vpn2CE2 vrf-import vpnimport
set routing-instances vpn2CE2 vrf-export vpnexport
set routing-instances vpn2CE2 protocols bgp group To_CE2 peer-as 64509
set routing-instances vpn2CE2 protocols bgp group To_CE2 neighbor 10.24.24.2
set routing-options autonomous-system 64511
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the Junos OS CLI User Guide.

To configure the EBGP scenario:

1. Configure the router interfaces.

```
[edit interfaces]
user@ASBR3# set fe-1/2/0 unit 0 description to-ASBR1
user@ASBR3# set fe-1/2/0 unit 0 family inet address 10.21.21.2/30
user@ASBR3# set fe-1/2/0 unit 0 family mpls
user@ASBR3# set fe-1/2/2 unit 0 description to-P3
user@ASBR3# set fe-1/2/2 unit 0 family inet address 10.22.22.1/30
user@ASBR3# set fe-1/2/2 unit 0 family mpls
user@ASBR3# set fe-1/2/1 unit 0 description to-ASBR2
user@ASBR3# set fe-1/2/1 unit 0 family inet address 10.26.26.2/30
user@ASBR3# set fe-1/2/1 unit 0 family mpls
user@ASBR3# set lo0 unit 0 family inet address 10.5.5.5/32
```

2. Configure an interior gateway protocol (IGP), such as OSPF or IS-IS.

```
[edit protocols ospf]
user@ASBR3# set traffic-engineering
[edit protocols ospf area 0.0.0.0]
user@ASBR3# set interface fe-1/2/2.0
```

```
user@ASBR3# set interface lo0.0 passive
user@ASBR3# set interface fe-1/2/1.0
```

3. Configure the autonomous system (AS) number.

```
[edit routing-options]
user@ASBR3# set autonomous-system 64511
```

4. Configure the routing policy.

```
[edit policy-options policy-statement To-ASBR1]
user@ASBR3# set term 1 from route-filter 10.7.7.7/32 exact
user@ASBR3# set term 1 then accept
user@ASBR3# set term 2 then reject
[edit policy-options policy-statement To-ASBR2]
user@ASBR3# set term 1 from route-filter 10.7.7.7/32 exact
user@ASBR3# set term 1 then accept
user@ASBR3# set term 2 then reject
[edit policy-options policy-statement next-hop-self]
user@ASBR3# set then next-hop self
```

5. Configure the EBGP sessions.

```
[edit protocols bgp group To-ASBR1]
user@ASBR3# set type external
user@ASBR3# set family inet labeled-unicast protection
user@ASBR3# set family inet labeled-unicast per-prefix-label
user@ASBR3# set export To-ASBR1
user@ASBR3# set neighbor 10.21.21.1 peer-as 64510
[edit protocols bgp group To-ASBR2]
user@ASBR3# set type external
user@ASBR3# set family inet labeled-unicast protection
user@ASBR3# set family inet labeled-unicast per-prefix-label
user@ASBR3# set export To-ASBR2
user@ASBR3# set neighbor 10.26.26.1 peer-as 64510
```

6. Configure the IBGP sessions.

```
[edit protocols bgp group To-PE2]
user@ASBR3# set type internal
user@ASBR3# set local-address 10.5.5.5
user@ASBR3# set family inet unicast
user@ASBR3# set export next-hop-self
user@ASBR3# set neighbor 10.7.7.7 family inet labeled-unicast
```

7. Configure MPLS.

```
[edit protocols mpls]
user@ASBR3# set traffic-engineering bgp-igp-both-ribs
user@ASBR3# set label-switched-path To_PE2 to 10.7.7.7
user@ASBR3# set interface lo0.0
user@ASBR3# set interface fe-1/2/0.0
user@ASBR3# set interface fe-1/2/2.0
user@ASBR3# set interface fe-1/2/1.0
```

8. Configure a signaling protocol.

```
[edit protocols rsvp]
user@ASBR3# set interface fe-1/2/2.0
user@ASBR3# set interface lo0.0
user@ASBR3# set interface fe-1/2/0.0
user@ASBR3# set interface fe-1/2/1.0
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols`, `show policy-options`, and `show routing-options`, commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@ASBR3# show interfaces
fe-1/2/0 {
    unit 0 {
        description to-ASBR1;
        family inet {
```

```
            address 10.21.21.2/30;
        }
        family mpls;
    }
}
fe-1/2/1 {
    unit 0 {
        description to-ASBR2;
        family inet {
            address 10.26.26.2/30;
        }
        family mpls;
    }
}
fe-1/2/2 {
    unit 0 {
        description to-P3;
        family inet {
            address 10.22.22.1/30;
        }
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.5.5.5/32;
        }
    }
}
```

```
user@ASBR3# show protocols
rsvp {
    interface fe-1/2/2.0;
    interface lo0.0;
    interface fe-1/2/0.0;
    interface fe-1/2/1.0;
}
mpls {
    traffic-engineering bgp-igp-both-ribs;
    label-switched-path To_PE2 {
```

```
        to 10.7.7.7;
    }
    interface lo0.0;
    interface fe-1/2/0.0;
    interface fe-1/2/2.0;
    interface fe-1/2/1.0;
}
bgp {
    group To-PE2 {
        type internal;
        local-address 10.5.5.5;
        family inet {
            unicast;
        }
        export next-hop-self;
        neighbor 10.7.7.7 {
            family inet {
                labeled-unicast;
            }
        }
    }
    group To-ASBR1 {
        type external;
        family inet {
            labeled-unicast {
                protection;
            }
        }
        export To-ASBR1;
        neighbor 10.21.21.1 {
            peer-as 64510;
        }
    }
    group To-ASBR2 {
        type external;
        family inet {
            labeled-unicast {
                protection;
            }
        }
        export To-ASBR2;
        neighbor 10.26.26.1 {
            peer-as 64510;
```

```
        }
    }
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface fe-1/2/2.0;
        interface lo0.0 {
            passive;
        }
        interface fe-1/2/1.0;
    }
}
```

```
user@ASBR3# show policy-options
policy-statement To-ASBR1 {
    term 1 {
        from {
            route-filter 10.7.7.7/32 exact;
        }
        then accept;
    }
    term 2 {
        then reject;
    }
}
policy-statement To-ASBR2 {
    term 1 {
        from {
            route-filter 10.7.7.7/32 exact;
        }
        then accept;
    }
    term 2 {
        then reject;
    }
}
policy-statement next-hop-self {
    then {
        next-hop self;
```

```
    }
}
```

```
user@ASBR3# show routing-options
autonomous-system 64511;
```

If you are done configuring the devices, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Checking the BGP Neighbor Sessions | 386
- Checking the Routes | 388

Confirm that the configuration is working properly.

Checking the BGP Neighbor Sessions

Purpose

Verify that BGP protection is enabled.

Action

```
user@ASBR3# show bgp neighbor 10.21.21.1
Peer:10.21.21.1+58259 AS 64510 Local: 10.21.21.2+179 AS 64511
  Type: External    State: Established    Flags: <ImportEval Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ To-ASBR1 ]
  Options: <Preference AddressFamily PeerAS Refresh>
  Options: <Protection>
  Address families configured: inet-labeled-unicast
  Holdtime: 90 Preference: 170
  NLRI configured with protection: inet-labeled-unicast
  Number of flaps: 0
  Peer ID: 10.4.4.4         Local ID: 10.5.5.5          Active Holdtime: 90
```

```
  Keepalive Interval: 30         Group index: 4    Peer index: 0
  BFD: disabled, down
  Local Interface: fe-1/2/0.0
  NLRI for restart configured on peer: inet-labeled-unicast
  NLRI advertised by peer: inet-labeled-unicast
  NLRI for this session: inet-labeled-unicast
  Peer supports Refresh capability (2)
  Stale routes from peer are kept for: 300
  Peer does not support Restarter functionality
  NLRI that restart is negotiated for: inet-labeled-unicast
  NLRI of received end-of-rib markers: inet-labeled-unicast
  NLRI of all end-of-rib markers sent: inet-labeled-unicast
  Peer supports 4 byte AS extension (peer-as 64510)
  Peer does not support Addpath
  Table inet.0 Bit: 10001
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              2
    Received prefixes:            1
    Accepted prefixes:            1
    Suppressed due to damping:    0
    Advertised prefixes:          1
  Last traffic (seconds): Received 7    Sent 20   Checked 32
  Input messages:  Total 170    Updates 2       Refreshes 0     Octets 3326
  Output messages: Total 167    Updates 1       Refreshes 0     Octets 3288
  Output Queue[0]: 0
```

```
user@ASBR3# show bgp neighbor 10.26.26.1
Peer: 10.26.26.1+61072 AS 64510 Local: 10.26.26.2+179 AS 64511
  Type: External    State: Established    Flags: <ImportEval Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ To-ASBR2 ]
  Options: <Preference AddressFamily PeerAS Refresh>
  Options: <Protection>
  Address families configured: inet-labeled-unicast
  Holdtime: 90 Preference: 170
  NLRI configured with protection: inet-labeled-unicast
  Number of flaps: 0
  Peer ID: 10.9.9.9         Local ID: 10.5.5.5           Active Holdtime: 90
  Keepalive Interval: 30         Group index: 5    Peer index: 0
```

```
  BFD: disabled, down
  Local Interface: fe-1/2/1.0
  NLRI for restart configured on peer: inet-labeled-unicast
  NLRI advertised by peer: inet-labeled-unicast
  NLRI for this session: inet-labeled-unicast
  Peer supports Refresh capability (2)
  Stale routes from peer are kept for: 300
  Peer does not support Restarter functionality
  NLRI that restart is negotiated for: inet-labeled-unicast
  NLRI of received end-of-rib markers: inet-labeled-unicast
  NLRI of all end-of-rib markers sent: inet-labeled-unicast
  Peer supports 4 byte AS extension (peer-as 64510)
  Peer does not support Addpath
  Table inet.0 Bit: 10002
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              1
    Received prefixes:            1
    Accepted prefixes:            1
    Suppressed due to damping:    0
    Advertised prefixes:          1
  Last traffic (seconds): Received 21   Sent 9    Checked 42
  Input messages:  Total 170    Updates 2       Refreshes 0     Octets 3326
  Output messages: Total 168    Updates 1       Refreshes 0     Octets 3307
  Output Queue[0]: 0
```

Meaning

The output shows that the `Protection` option is enabled for the EBGP peers, Device ASBR1 and Device ASBR2.

This is also shown with the `NLRI configured with protection: inet-labeled-unicast` screen output.

Checking the Routes

Purpose

Make sure that the backup path is installed in the routing table.

Action

```
user@ASBR3> show route 10.2.2.2
inet.0: 12 destinations, 14 routes (12 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.2.2.2/32         *[BGP/170] 01:36:25, MED 2, localpref 100
                      AS path: 64510 I, validation-state: unverified
                    > to 10.21.21.1 via fe-1/2/0.0, Push 299824
                      to 10.26.26.1 via fe-1/2/1.0, Push 299808
                    [BGP/170] 01:36:25, MED 2, localpref 100
                      AS path: 64510 I, validation-state: unverified
                    > to 10.26.26.1 via fe-1/2/1.0, Push 299808
```

Meaning

The `show route` command displays active as well as backup paths to Device PE1.

SEE ALSO

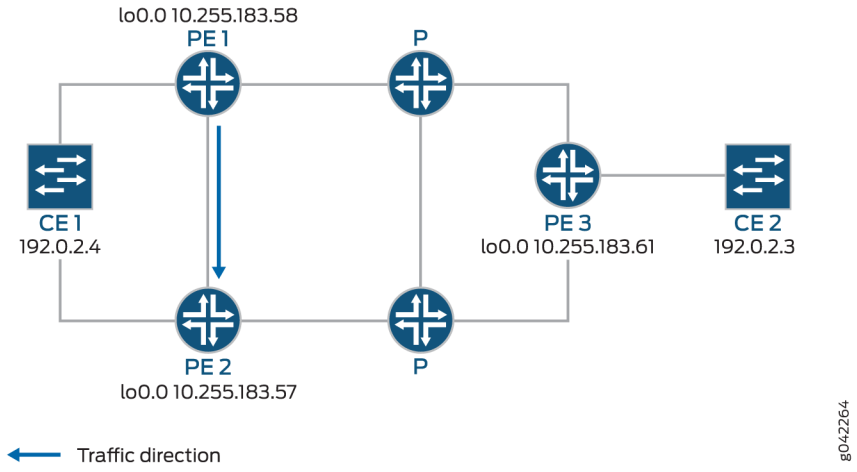
- Example: Preventing BGP Session Resets
- Examples: Configuring BGP Flap Damping

Configuring Egress Protection Service Mirroring for BGP Signaled Layer 2 Services

Starting in Junos OS Release 14.2, Junos OS supports the restoration of egress traffic when there is a link or node failure in the egress PE node. If there is a link or node failure in the core network, a protection mechanism such as MPLS fast reroute can be triggered on the transport LSPs between the PE routers to repair the connection within tens of milliseconds. An egress protection LSP addresses the problem of a node-link failure at the edge of the network (for example, a failure of a PE router).

Figure 1 shows a simplified topology of the use case that explains this feature.

Figure 16: Egress Protection LSP Configured from Router PE1 to Router PE2



CE1 is multihomed to PE1 and PE2. There are two paths connecting CE1 and CE2. The working path is CE2-PE3-P-PE1-CE1, via pseudowire PW21. The protecting path is CE2-PE3-P-PE2-CE1, via pseudowire PW22 Traffic is flowing through the working path under normal circumstances. When the end-to-end OAM between CE1 and CE2 detects failure on the working path, traffic will be switched from the working path to the protecting path. The end-to-end failure detection and recovery relies on control plane hence should be relatively slow. To achieve faster protection, local repair mechanisms similar to those used by MPLS fast reroute should be used. In Figure 1 above, if link or node failed in the core network (like link failure on P-PE1, P-PE3, or node failure on P), the MPLS fast reroute will happen on the transport LSPs between PE1 and PE3. The failure could be locally repaired within tens of milliseconds. However, if link or node failure happens at the edge (like link failure on PE3-CE2 or node failure on PE3), there is no local repair currently so we have to rely on the CE1-CE2 end-to-end protection to repair the failure.

- Device CE2—Traffic origin
- Router PE3—Ingress PE router
- Router PE1— (Primary) Egress PE router
- Router PE2—Protector PE router
- Device CE1—Traffic destination

When the link between CE1– PE1 goes downs, PE1 will briefly redirect that traffic towards CE1, to PE2. PE2 forwards it to CE1 until ingress router PE3 recalculates to forward the traffic to PE2.

Initially the traffic direction was; CE2 – PE3 – P – PE1 – CE1.

When the link between CE1– PE1 goes down, the traffic will be; CE2 – PE3 – P – PE1 – PE2 –CE1. PE3 then recalculates the path; CE2 – PE3 – P – PE2 – CE1.

1. Configure RSVP on PE1, PE2, and PE3.

```
[edit protocols]
user@PE1# set interface all
user@PE2# set interface all
user@PE3# set interface all
```

2. Configure MPLS.

```
[edit protocols mpls]
user@PE1# set interface all
user@PE2# set interface all
user@PE3# set interface all
```

3. Set PE1 as `primary` and PE2 as `protector` nodes.

```
[edit protocols mpls]
user@PE1# set egress-protection context-identifier address primary
user@PE2# set egress-protection context-identifier address protector
```

4. Enable `egress-protection` on PE1 and PE2.

```
[edit protocols bgp]
user@PE1# set group ibgp family l2vpn egress-protection
user@PE2# set group ibgp family l2vpn egress-protection
```

5. Configure LDP and ISIS on PE1, PE2, and PE3.

```
[edit protocols ldp]
user@PE1# set interface all
user@PE2# set interface all
user@PE3# set interface all
```

```
[edit protocols isis]
user@PE1# set interface all point-to-point
user@PE2# set interface all point-to-point
user@PE3# set interface all point-to-point
```

6. Configure a load balancing policy at PE1, PE2, and PE3.

```
[edit]
user@PE1# set policy-options policy-statement lb then load-balance per-packet
user@PE2# set policy-options policy-statement lb then load-balance per-packet
user@PE3# set policy-options policy-statement lb then load-balance per-packet
```

7. Configure the routing options at PE1, PE2, and PE3, to export routes based on the load balancing policy.

```
[edit]
user@PE1# set routing-options traceoptions file ro.log
user@PE1# set routing-options traceoptions flag normal
user@PE1# set routing-options traceoptions flag route
user@PE1# set routing-options autonomous-system 100
user@PE1# set routing-options forwarding-table export lb
```

```
[edit]
user@PE2# set routing-options traceoptions file ro.log
user@PE2# set routing-options traceoptions flag normal
user@PE2# set routing-options traceoptions flag route
user@PE2# set routing-options autonomous-system 100
user@PE2# set routing-options forwarding-table export lb
```

```
[edit]
user@PE3# set routing-options traceoptions file ro.log
user@PE3# set routing-options traceoptions flag normal
user@PE3# set routing-options traceoptions flag route
user@PE3# set routing-options autonomous-system 100
user@PE3# set routing-options forwarding-table export lb
```

8. Configure BGP at PE1 to advertise nrli from the routing instance with context-ID as next-hop.

```
[edit]
user@PE1# set routing-instances foo egress-protection context-identifier context-identifier
```

9. Configure l2vpn at PE1, PE2, and PE3

At PE1:

```
[edit routing-instances]
foo {
    instance-type l2vpn;
    egress-protection {
        context-identifier {
            198.51.100.0;
        }
    }
    interface ge-2/0/2.0;
    route-distinguisher 10.255.183.58:1;
    vrf-target target:9000:1;
    protocols {
        l2vpn {
            encapsulation-type ethernet-vlan;
            site foo {
                site-identifier 1;
                multi-homing;
                site-preference primary;
                interface ge-2/0/2.0 {
                    remote-site-id 2;
                }
            }
        }
    }
}
```

At PE2:

```
[edit routing-instances]
foo {
    instance-type l2vpn;
    egress-protection {
        protector;
    }
    interface ge-2/0/2.0;
    route-distinguisher 10.255.183.57:1;
    vrf-target target:9000:1;
    protocols {
        l2vpn {
```

```
            encapsulation-type ethernet-vlan;
            site foo{
                site-identifier 1;
                multi-homing;
                site-preference backup;
                interface ge-2/0/2.0 {
                    remote-site-id 2;
                }
            }
        }
    }
}
```

At PE3:

```
[edit routing-instances]
foo {
    instance-type l2vpn;
    interface ge-2/1/2.0;
    route-distinguisher 10.255.183.61:1;
    vrf-target target:9000:1;
    protocols {
        l2vpn {
            encapsulation-type ethernet-vlan;
            site foo {
                site-identifier 2;
                interface ge-2/1/2.0;
            }
        }
    }
}
```

Example: Configuring MPLS Egress Protection Service Mirroring for BGP Signaled Layer 2 Services

IN THIS SECTION

- Requirements | 395
- Overview | 395

Configuration | 397
Verification | 414

Starting in Junos OS Release 14.2, Junos OS supports the restoration of egress traffic when there is a link or node failure in the egress PE node. If there is a link or node failure in the core network, a protection mechanism such as MPLS fast reroute can be triggered on the transport LSPs between the PE routers to repair the connection within tens of milliseconds. An egress protection LSP addresses the problem of a node-link failure at the edge of the network (for example, a failure of a PE router).

This example shows how to configure link protection for BGP signaled Layer 2 services.

Requirements

MX Series Routers running Junos OS Release 14.2 or later.

Overview

IN THIS SECTION

Topology | 396

If there is a link or node failure in the core network, a protection mechanism such as MPLS fast reroute can be triggered on the transport LSPs between the PE routers to repair the connection within tens of milliseconds. An egress protection LSP addresses the problem of a node-link failure at the edge of the network (for example, a failure of a PE router).

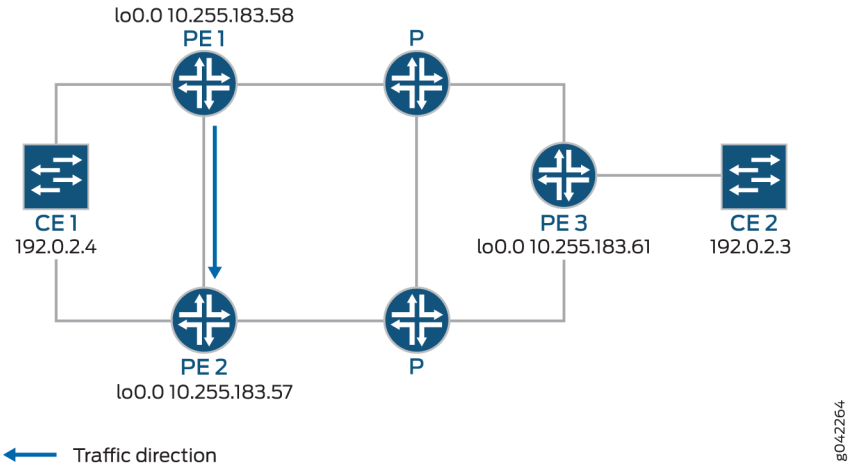
This example includes the following configuration concepts and statements that are unique to the configuration of an egress protection LSP:

- `context-identifier`—Specifies an IPv4 or IPv6 address used to define the pair of PE routers participating in the egress protection LSP. It is assigned to each ordered pair of primary PE and the protector to facilitate protection establishment. This address is globally unique, or unique in the address space of the network where the primary PE and the protector reside.
- `egress-protection`—Configures the protector information for the protected Layer 2 circuit and configures the protector Layer 2 circuit at the `[edit protocols mpls]` hierarchy level. Configures an LSP as an egress protection LSP at the `[edit protocols mpls]` hierarchy level.

- `protector`—Configures the creation of standby pseudowires on the backup PE for link or node protection for the instance.

Topology

Figure 17: Egress Protection LSP Configured from Router PE1 to Router PE2



In the event of a failure of the egress PE Router PE1, traffic is switched to the egress protection LSP configured between Router PE1 and Router PE2 (the protector PE router):

- Device CE2—Traffic origin
- Router PE3—Ingress PE router
- Router PE1— (Primary) Egress PE router
- Router PE2—Protector PE router
- Device CE1—Traffic destination

When the link between CE1– PE1 goes downs, PE1 will briefly redirect that traffic toward CE1, to PE2. PE2 forwards it to CE1 until ingress router PE3 recalculates to forward the traffic to PE2.

Initially the traffic direction was: CE2 – PE3 – P – PE1 – CE1.

When the link between CE1– PE1 goes down, the traffic will be: CE2 – PE3 – P – PE1 – PE2 –CE1. PE3 then recalculates the path: CE2 – PE3 – P – PE2 – CE1.

This example shows how to configure routers PE1, PE2, and PE3.

Configuration

IN THIS SECTION

- CLI Quick Configuration | 397
- Step-by-Step Procedure | 400
- Results | 407

CLI Quick Configuration

To quickly configure an egress protection LSP, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configurations, copy and then paste the commands into the CLI and enter `commit` from configuration mode.

PE1

```
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols mpls egress-protection context-identifier 198.51.100.3 primary
set protocols mpls egress-protection context-identifier 198.51.100.3 advertise-mode stub-alias
set protocols mpls egress-protection traceoptions file ep size 100m
set protocols mpls egress-protection traceoptions flag all
set protocols bgp traceoptions file bgp.log world-readable
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.255.183.58
set protocols bgp group ibgp family inet unicast
set protocols bgp group ibgp family l2vpn signaling egress-protection
set protocols bgp group ibgp neighbor 192.0.2.3
set protocols bgp group ibgp neighbor 192.0.2.4
set protocols isis traceoptions file isis-edge size 10m world-readable
set protocols isis traceoptions flag error
set protocols isis level 1 disable
set protocols isis level 2 wide-metrics-only
set protocols isis interface all point-to-point
set protocols isis interface all level 2 metric 10
set protocols isis interface fxp0.0 disable
set protocols ldp interface all
```

```
set protocols ldp interface fxp0.0 disable
set policy-options policy-statement lb then load-balance per-packet
set routing-options traceoptions file ro.log
set routing-options traceoptions flag all
set routing-options traceoptions flag route
set routing-options autonomous-system 100
set routing-options forwarding-table export lb
set routing-instances foo instance-type l2vpn
set routing-instances foo egress-protection context-identifier 198.51.100.3
set routing-instances foo interface ge-2/0/2.0
set routing-instances foo route-distinguisher 10.255.183.58:1
set routing-instances foo vrf-target target:9000:1
set routing-instances foo protocols l2vpn encapsulation-type ethernet-vlan
set routing-instances foo protocols l2vpn site foo site-identifier 1
set routing-instances foo protocols l2vpn site foo site-preference primary
set routing-instances foo protocols l2vpn site foo interface ge-2/0/2.0 remote-site-id 2
```

PE2

```
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols mpls egress-protection context-identifier 198.51.100.3 protector
set protocols mpls egress-protection context-identifier 198.51.100.3 advertise-mode stub-alias
set protocols mpls egress-protection traceoptions file ep size 100m
set protocols mpls egress-protection traceoptions flag all
set protocols bgp traceoptions file bgp.log world-readable
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.255.183.57
set protocols bgp group ibgp family inet unicast
set protocols bgp group ibgp family l2vpn signaling egress-protection
set protocols bgp group ibgp neighbor 192.0.2.3
set protocols bgp group ibgp neighbor 192.0.2.4
set protocols isis traceoptions file isis-edge size 10m world-readable
set protocols isis traceoptions flag error
set protocols isis level 1 disable
set protocols isis level 2 wide-metrics-only
set protocols isis interface all point-to-point
set protocols isis interface all level 2 metric 10
set protocols isis interface fxp0.0 disable
set protocols ldp interface all
```

```
set protocols ldp interface fxp0.0 disable
set policy-options policy-statement lb then load-balance per-packet
set routing-options traceoptions file ro.log
set routing-options traceoptions flag normal
set routing-options traceoptions flag route
set routing-options autonomous-system 100
set routing-options forwarding-table export lb
set routing-instances foo instance-type l2vpn
set routing-instances foo egress-protection protector
set routing-instances foo interface ge-2/0/2.0
set routing-instances foo route-distinguisher 10.255.183.57:1
set routing-instances foo vrf-target target:9000:1
set routing-instances foo protocols l2vpn encapsulation-type ethernet-vlan
set routing-instances foo protocols l2vpn site foo hot-standby
set routing-instances foo protocols l2vpn site foo site-identifier 1
set routing-instances foo protocols l2vpn site foo site-preference backup
set routing-instances foo protocols l2vpn site foo interface ge-2/0/2.0 remote-site-id 2
```

PE3

```
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp traceoptions file bgp.log world-readable
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.255.183.61
set protocols bgp group ibgp family inet unicast
set protocols bgp group ibgp family l2vpn signaling
set protocols bgp group ibgp neighbor 192.0.2.3
set protocols bgp group ibgp neighbor 192.0.2.4
set protocols isis traceoptions file isis-edge size 10m world-readable
set protocols isis traceoptions flag error
set protocols isis level 1 disable
set protocols isis level 2 wide-metrics-only
set protocols isis interface all point-to-point
set protocols isis interface all level 2 metric 10
set protocols isis interface fxp0.0 disable
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set policy-options policy-statement lb then load-balance per-packet
set routing-options traceoptions file ro.log
```

```
set routing-options traceoptions flag normal
set routing-options traceoptions flag route
set routing-options autonomous-system 100
set routing-options forwarding-table export lb
set routing-instances foo instance-type l2vpn
set routing-instances foo interface ge-2/1/2.0
set routing-instances foo route-distinguisher 10.255.183.61:1
set routing-instances foo vrf-target target:9000:1
set routing-instances foo protocols l2vpn encapsulation-type ethernet-vlan
set routing-instances foo protocols l2vpn site foo site-identifier 2
set routing-instances foo protocols l2vpn site foo interface ge-2/1/2.0 remote-site-id 1
```

Step-by-Step Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode*.

To configure an egress protection LSP for router PE1:

1. Configure RSVP.

```
[edit protocols rsvp]
user@PE1# set interface all
user@PE1# set interface fxp0.0 disable
```

2. Configure MPLS to use the egress protection LSP to protect against a link failure to Device CE1.

```
[edit protocols mpls]
user@PE1# set interface all
user@PE1# set interface fxp0.0 disable
user@PE1# set egress-protection context-identifier 198.51.100.3 primary
user@PE1# set egress-protection context-identifier 198.51.100.3 advertise-mode stub-alias
user@PE1# set egress-protection traceoptions file ep size 100m
user@PE1# set egress-protection traceoptions flag all
```

3. Configure BGP.

```
[edit protocols bgp]
user@PE1# set traceoptions file bgp.log world-readable
user@PE1# set group ibgp type internal
user@PE1# set group ibgp local-address 10.255.183.58
user@PE1# set group ibgp family inet unicast
user@PE1# set group ibgp family l2vpn signaling egress-protection
user@PE1# set group ibgp neighbor 192.0.2.3
user@PE1# set group ibgp neighbor 192.0.2.4
```

4. Configure IS-IS.

```
[edit protocols isis]
user@PE1# set traceoptions file isis-edge size 10m world-readable
user@PE1# set traceoptions flag error
user@PE1# set level 1 disable
user@PE1# set level 2 wide-metrics-only
user@PE1# set interface all point-to-point
user@PE1# set interface all level 2 metric 10
user@PE1# set interface fxp0.0 disable
```

5. Configure LDP.

```
[edit protocols ldp]
user@PE1# set interface all
user@PE1# set interface fxp0.0 disable
```

6. Configure a load-balancing policy.

```
[edit]
user@PE1# set policy-options policy-statement lb then load-balance per-packet
```

7. Configure the routing options to export routes based on the load-balancing policy.

```
[edit routing-options]
user@PE1# set traceoptions file ro.log
user@PE1# set traceoptions flag all
```

```
user@PE1# set autonomous-system 100
user@PE1# set forwarding-table export lb
```

8. Configure BGP to advertise nrli from the routing instance with context-ID as next-hop.

```
[edit routing-instances]
user@PE1# set foo instance-type l2vpn
user@PE1# set foo egress-protection context-identifier 198.51.100.3
user@PE1# set foo interface ge-2/0/2.0
user@PE1# set foo route-distinguisher 10.255.183.58:1
user@PE1# set foo vrf-target target:9000:1
```

9. Configure l2vpn instance to use the egress LSP configured.

```
[edit routing-instances]
user@PE1# set foo protocols l2vpn encapsulation-type ethernet-vlan
user@PE1# set foo protocols l2vpn site foo site-identifier 1
user@PE1# set foo protocols l2vpn site foo site-preference primary
user@PE1# set foo protocols l2vpn site foo interface ge-2/0/2.0 remote-site-id 2
```

10. If you are done configuring the device, enter `commit` from configuration mode.

Step-by-Step Procedure

To configure an egress protection LSP for Router PE2:

1. Configure RSVP.

```
[edit protocols rsvp]
user@PE2# set interface all
user@PE2# set interface fxp0.0 disable
```

2. Configure MPLS and the LSP that acts as the egress protection LSP.

```
[edit protocols mpls]
user@PE2# set interface all
user@PE2# set interface fxp0.0 disable
user@PE2# set egress-protection context-identifier 198.51.100.3 protector
user@PE2# set egress-protection context-identifier 198.51.100.3 advertise-mode stub-alias
```

```
user@PE2# set egress-protection traceoptions file ep size 100m
user@PE2# set egress-protection traceoptions flag all
```

3. Configure BGP.

```
[edit protocols bgp]
user@PE2# set traceoptions file bgp.log world-readable
user@PE2# set group ibgp type internal
user@PE2# set group ibgp local-address 10.255.183.57
user@PE2# set group ibgp family inet unicast
user@PE2# set group ibgp family l2vpn signaling
user@PE2# set group ibgp family l2vpn egress-protection
user@PE2# set group ibgp neighbor 192.0.2.3
user@PE2# set group ibgp neighbor 192.0.2.4
```

4. Configure IS-IS.

```
[edit protocols isis]
user@PE2# set traceoptions file isis-edge size 10m world-readable
user@PE2# set traceoptions flag error
user@PE2# set level 1 disable
user@PE2# set level 2 wide-metrics-only
user@PE2# set interface all point-to-point
user@PE2# set interface all level 2 metric 10
user@PE2# set interface fxp0.0 disable
```

5. Configure LDP.

```
[edit protocols ldp]
user@PE2# set interface all
user@PE2# set interface fxp0.0 disable
```

6. Configure a load-balancing policy.

```
[edit]
user@PE2# set policy-options policy-statement lb then load-balance per-packet
```

7. Configure the routing options to export routes based on the load-balancing policy.

```
[edit routing-options]
user@PE2# set traceoptions file ro.log
user@PE2# set traceoptions flag all
user@PE2# set autonomous-system 100
user@PE2# set forwarding-table export lb
```

8. Configure BGP to advertise nrli from the routing instance with context-ID as next-hop.

```
[edit routing-instances]
user@PE2# set foo instance-type l2vpn
user@PE2# set foo egress-protection protector
user@PE2# set foo interface ge-2/0/2.0
user@PE2# set foo route-distinguisher 10.255.183.57:1
user@PE2# set foo vrf-target target:9000:1
```

9. Configure l2vpn instance to use the egress LSP configured.

```
[edit routing-instances]
user@PE2# set foo protocols l2vpn encapsulation-type ethernet-vlan
user@PE2# set foo protocols l2vpn site foo hot-standby
user@PE2# set foo protocols l2vpn site foo site-identifier 1
user@PE2# set foo protocols l2vpn site foo site-preference backup
user@PE2# set foo protocols l2vpn site foo interface ge-2/0/2.0 remote-site-id 2
```

10. If you are done configuring the device, enter `commit` from configuration mode.

Step-by-Step Procedure

To configure an egress protection LSP for Router PE3:

1. Configure RSVP.

```
[edit protocols rsvp]
user@PE3# set interface all
user@PE3# set interface fxp0.0 disable
```

2. Configure MPLS.

```
[edit protocols mpls]
user@PE3# set interface all
user@PE3# set interface fxp0.0 disable
```

3. Configure BGP.

```
[edit protocols bgp]
user@PE3# set traceoptions file bgp.log world-readable
user@PE3# set group ibgp type internal
user@PE3# set group ibgp local-address 10.255.183.61
user@PE3# set group ibgp family inet unicast
user@PE3# set group ibgp family l2vpn signaling
user@PE3# set group ibgp neighbor 192.0.2.3
user@PE3# set group ibgp neighbor 192.0.2.4
```

4. Configure IS-IS.

```
[edit protocols isis]
user@PE3# set traceoptions file isis-edge size 10m world-readable
user@PE3# set traceoptions flag error
user@PE3# set level 1 disable
user@PE3# set level 2 wide-metrics-only
user@PE3# set protocols isis interface all point-to-point
[edit protocols isis]
user@PE3# set protocols isis interface all level 2 metric 10
[edit protocols isis]
user@PE3# set protocols isis interface fxp0.0 disable
```

5. Configure LDP.

```
[edit protocols ldp]
user@PE3# set interface all
user@PE3# set interface fxp0.0 disable
```

6. Configure a load-balancing policy.

```
[edit]
user@PE3# set policy-options policy-statement lb then load-balance per-packet
```

7. Configure the routing options to export routes based on the load-balancing policy.

```
[edit routing-options]
user@PE3# set traceoptions file ro.log
user@PE3# set traceoptions flag normal
user@PE3# set traceoptions flag route
user@PE3# set autonomous-system 100
user@PE3# set forwarding-table export lb
```

8. Configure BGP to advertise nlri from the routing instance with context-ID as next-hop.

```
[edit]
user@PE3# set routing-instances foo instance-type l2vpn
user@PE3# set routing-instances foo interface ge-2/1/2.0
user@PE3# set routing-instances foo route-distinguisher 10.255.183.61:1
user@PE3# set routing-instances foo vrf-target target:9000:1
```

9. Configure l2vpn to specify the interface that connects to the site and the remote interface to which you want the specified interface to connect.

```
[edit routing-instances]
user@PE3# set foo protocols l2vpn encapsulation-type ethernet-vlan
user@PE3# set foo protocols l2vpn site foo site-identifier 2
user@PE3# set foo protocols l2vpn site foo interface ge-2/1/2.0 remote-site-id 1
```

10. If you are done configuring the device, enter `commit` from configuration.

Results

From configuration mode, confirm your configuration on Router PE1 by entering the **show protocols**, **show policy-options**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@PE1# show protocols
rsvp {
    interface all;
    interface fxp0.0 {
        disable;
    }
}
mpls {
    interface all;
    interface fxp0.0 {
        disable;
    }
    egress-protection {
        context-identifier 198.51.100.3 {
            primary;
            advertise-mode stub-alias;
        }
        traceoptions {
            file ep size 100m;
            flag all;
        }
    }
}
bgp {
    traceoptions {
        file bgp.log world-readable;
    }
    group ibgp {
        type internal;
        local-address 10.255.183.58;
        family inet {
            unicast;
        }
        family l2vpn {
            signaling {
```

```
                egress-protection;
            }
        }
        neighbor 192.0.2.3;
        neighbor 192.0.2.4;
    }
}
isis {
    traceoptions {
        file isis-edge size 10m world-readable;
        flag error;
    }
    level 1 disable;
    level 2 wide-metrics-only;
    interface all {
        point-to-point;
        level 2 metric 10;
    }
    interface fxp0.0 {
        disable;
    }
}
ldp {
    interface all;
    interface fxp0.0 {
        disable;
    }
}

[edit]
user@PE1# show policy-options
policy-statement lb {
    then {
        load-balance per-packet;
    }
}
[edit]
user@PE1# show routing-options
traceoptions {
    file ro.log;
    flag all;
}
autonomous-system 100;
```

```
forwarding-table {
    export lb;
}

[edit]
user@PE1# show routing-instances
foo {
    instance-type l2vpn;
    egress-protection {
        context-identifier {
            198.51.100.3;
        }
    }
    interface ge-2/0/2.0;
    route-distinguisher 10.255.183.58:1;
    vrf-target target:9000:1;
    protocols {
        l2vpn {
            encapsulation-type ethernet-vlan;
            site foo {
                site-identifier 1;
                site-preference primary;
                interface ge-2/0/2.0 {
                    remote-site-id 2;
                }
            }
        }
    }
}
```

From configuration mode, confirm your configuration on Router PE2 by entering the **show protocols**, **show policy-options**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@PE2# show protocols
rsvp {
    interface all;
    interface fxp0.0 {
        disable;
    }
}
```

```
mpls {
    interface all;
    interface fxp0.0 {
        disable;
    }

    egress-protection {
        context-identifier 198.51.100.3 {
            protector;
            advertise-mode stub-alias;
        }
        traceoptions {
            file ep size 100m;
            flag all;
        }
    }
}
bgp {
    traceoptions {
        file bgp.log world-readable;
    }
    group ibgp {
        type internal;
        local-address 10.255.183.57;
        family inet {
            unicast;
        }
        family l2vpn {
            signaling {
                egress-protection;
            }
        }
        neighbor 192.0.2.3;
        neighbor 192.0.2.4;
    }
}
isis {
    traceoptions {
        file isis-edge size 10m world-readable;
        flag error;
    }
    level 1 disable;
    level 2 wide-metrics-only;
```

```
    interface all {
        point-to-point;
        level 2 metric 10;
    }
    interface fxp0.0 {
        disable;
    }
}
ldp {
    interface all;
    interface fxp0.0 {
        disable;
    }
}

[edit]
user@PE2# show policy-options
policy-statement lb {
    then {
        load-balance per-packet;
    }
}

[edit]
user@PE2# show routing-options
traceoptions {
    file ro.log;
    flag normal;
    flag route;
}
autonomous-system 100;
forwarding-table {
    export lb;
}

[edit]
user@PE2# show routing-instances
foo {
    instance-type l2vpn;
    egress-protection {
        protector;
    }
    interface ge-2/0/2.0;
```

```
    route-distinguisher 10.255.183.57:1;
    vrf-target target:9000:1;
    protocols {
        l2vpn {
            encapsulation-type ethernet-vlan;
            site foo {
                hot-standby;
                site-identifier 1;
                site-preference backup;
                interface ge-2/0/2.0 {
                    remote-site-id 2;
                }
            }
        }
    }
}
```

From configuration mode, confirm your configuration on Router PE3 by entering the **show protocols**, **show policy-options**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@PE3# show protocols
rsvp {
    interface all;
    interface fxp0.0 {
        disable;
    }
}
mpls {
    interface all;
    interface fxp0.0 {
        disable;
    }
}
bgp {
    traceoptions {
        file bgp.log world-readable;
    }
    group ibgp {
        type internal;
        local-address 10.255.183.61;
```

```
        family inet {
            unicast;
        }
        family l2vpn {
            signaling;
        }
        neighbor 192.0.2.3;
        neighbor 192.0.2.4;
    }
}
isis {
    traceoptions {
        file isis-edge size 10m world-readable;
        flag error;
    }
    level 1 disable;
    level 2 wide-metrics-only;
    interface all {
        point-to-point;
        level 2 metric 10;
    }
    interface fxp0.0 {
        disable;
    }
}
ldp {
    interface all;
    interface fxp0.0 {
        disable;
    }
}

[edit]
user@PE3# show policy-options
policy-statement lb {
    then {
        load-balance per-packet;
    }
}

[edit]
user@PE3# show routing-options
traceoptions {
```

```
    file ro.log;
    flag normal;
    flag route;
}
autonomous-system 100;
forwarding-table {
    export lb;
}

[edit]
user@PE3# show routing-instances
foo {
    instance-type l2vpn;
    interface ge-2/1/2.0;
    route-distinguisher 10.255.183.61:1;
    vrf-target target:9000:1;
    protocols {
        l2vpn {
            encapsulation-type ethernet-vlan;
            site foo {
                site-identifier 2;
                interface ge-2/1/2.0 {
                    remote-site-id 1;
                }
            }
        }
    }
}
```

Verification

IN THIS SECTION

- Verifying the L2VPN Configuration | **415**
- Verifying the Routing Instance Details | **416**
- Verifying the IS-IS Configuration | **417**
- Verifying the MPLS Configuration | **417**

Confirm that the configuration is working properly.

Verifying the L2VPN Configuration

Purpose

Verify that LSP is protected by the connection protection logic.

Action

From operational mode, run the `show l2vpn connections extensive` command.

```
user@PE2> show l2vpn connections extensive
```

```
Layer-2 VPN connections:
Legend for connection status (St)
EI -- encapsulation invalid      NC -- interface encapsulation not CCC/TCC/VPLS
EM -- encapsulation mismatch     WE -- interface and instance encaps not same
VC-Dn -- Virtual circuit down    NP -- interface hardware not present
CM -- control-word mismatch      -> -- only outbound connection is up
CN -- circuit not provisioned    <- -- only inbound connection is up
OR -- out of range               Up -- operational
OL -- no outgoing label          Dn -- down
LD -- local site signaled down   CF -- call admission control failure
RD -- remote site signaled down  SC -- local and remote site ID collision
LN -- local site not designated  LM -- local site ID not minimum designated
RN -- remote site not designated RM -- remote site ID not minimum designated
XX -- unknown connection status  IL -- no incoming label
MM -- MTU mismatch               MI -- Mesh-Group ID not available
BK -- Backup connection          ST -- Standby connection
PF -- Profile parse failure      PB -- Profile busy
RS -- remote site standby        SN -- Static Neighbor
LB -- Local site not best-site   RB -- Remote site not best-site
VM -- VLAN ID mismatch
Legend for interface status
Up -- operational
Dn -- down
Instance: foo
Local site: foo (1)
```

```
  connection-site           Type  St  Time last up          # Up trans
   2                         rmt   Up  Aug  3 00:08:14 2001           1
     Local circuit: ge-2/0/2.0, Status: Up
     Remote PE: 192.0.2.3
     Incoming label: 32769, Outgoing label: 32768
     Egress Protection: Yes
       Time                  Event            Interface/Lbl/PE
       Aug  3 00:08:14 2001  PE route up
       Aug  3 00:08:14 2001  Out lbl Update               32768
       Aug  3 00:08:14 2001  In lbl Update                32769
       Aug  3 00:08:14 2001  ckt0 up                 fe-0/0/0.0
```

Meaning

The `Egress Protection: Yes` output shows that the given PVC is protected by connection protection logic.

Verifying the Routing Instance Details

Purpose

Verify the routing instance information and the context identifier configured on the primary, which is used as the next-hop address in case of node-link failure.

Action

From operational mode, run the `show route foo detail` command.

```
user@PE2> show route foo detail
```

```
foo:
  Router ID: 0.0.0.0
  Type: l2vpn non-forwarding State: Active
  Interfaces:
    lt-1/2/0.56
  Route-distinguisher: 10.255.255.11:1
  Vrf-import: [ __vrf-import-foo-internal__ ]
  Vrf-export: [ __vrf-export-foo-internal__ ]
  Vrf-import-target: [ target:100:200 ]
  Vrf-export-target: [ target:100:200 ]
```

```
  Fast-reroute-priority: low
  Vrf-edge-protection-id:  198.51.100.3
  Tables:
    foo.l2vpn.0           : 5 routes (3 active, 0 holddown, 0 hidden)
    foo.l2id.0            : 6 routes (2 active, 0 holddown, 0 hidden)
```

Meaning

The context-id is set to `198.51.100.3` and the `Vrf-import: [ __vrf-import-foo-internal__]` in the output mentions the policy used for rewriting the next-hop address.

Verifying the IS-IS Configuration

Purpose

Verify the IS-IS context identifier information.

Action

From operational mode, run the `show isis context-identifier detail` command.

```
user@PE2> show isis context-identifier detail
```

```
IS-IS context database:
Context               L  Owner     Role       Primary      Metric
198.51.100.3             2  MPLS      Protector  pro17-b-lr-R1 0
  Advertiser pro17-b, Router ID 10.255.107.49, Level 2, tlv protector
  Advertiser pro17-b-lr-R1, Router ID 10.255.255.11, Metric 1, Level 2, tlv prefix
```

Meaning

Router PE2 is the protector and the configured context identifier is in use for the MPLS protocol.

Verifying the MPLS Configuration

Purpose

Verify the context identifier details on the primary and protector PEs.

Action

From operational mode, run the `show mpls context-identifier detail` command.

```
user@PE1> show mpls context-identifier detail
```

```
ID: 198.51.100.3
  Type: primary, Metric: 1, Mode: alias

Total 1, Primary 1, Protector 0
```

```
user@PE2> show mpls context-identifier detail
```

```
ID: 198.51.100.3
  Type: protector, Metric: 16777215, Mode: alias
  Context table: __198.51.100.3__.mpls.0, Label out: 299968
```

```
user@PE2> show mpls egress-protection detail
```

```
Instance                Type      Protection-Type
foo                   local-l2vpn Protector
  Route Target 100:200
```

Meaning

Context-id is `198.51.100.3`, advertise-mode is `alias`, the MPLS table created for egress protection is `__198.51.100.3__.mpls.0`, and the egress instance name is `foo`, which is of type `local-l2vpn`.

Example: Configuring Layer 3 VPN Egress Protection with PLR as Protector

IN THIS SECTION

- Requirements | 419
- Overview | 419
- Configuration | 420
- Verification | 443

This example shows how to configure fast service restoration at the egress of a Layer 3 VPN when the customer is multihomed to the service provider.

Starting in Junos OS Release 15.1, the enhanced point of local repair (PLR) functionality addresses a special scenario of egress node protection, where the PLR and the protector are co-located as one router. In this case, there is no need to have a bypass LSP reroute traffic during local repair. Instead, the PLR or the protector can send the traffic directly to the target CE (in Co-located protector model where the PLR or the protector is also the backup PE that is directly connected to the CE) or to the backup PE (in Centralized protector model where the backup PE is a separate router).

Requirements

No special configuration beyond device initialization is required before configuring this example.

This example requires Junos OS Release 15.1 or later.

Overview

IN THIS SECTION

- Topology | 420

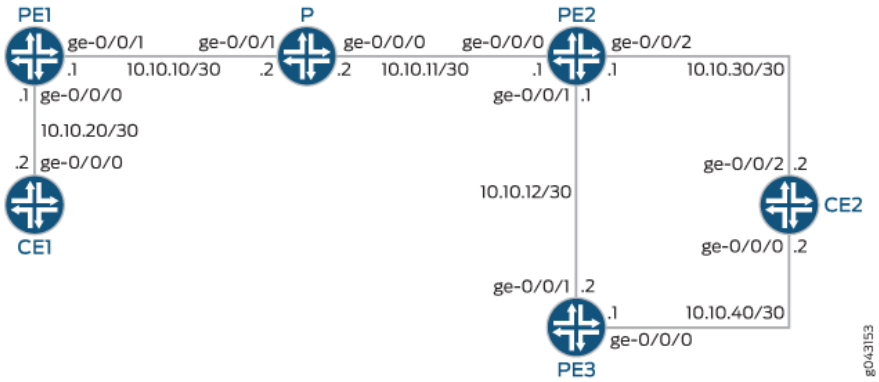
As a special scenario of egress node protection, if a router is both a Protector and a PLR, it installs backup next hops to protect the transport LSP. In particular, it does not need a bypass LSP for local repair.

In the Co-located protector model, the PLR or the Protector is directly connected to the CE via a backup AC, while in the Centralized protector model, the PLR or the protector has an MPLS tunnel to the backup PE. In either case, the PLR or the Protector will install a backup next hop with a label followed by a lookup in a `context label` table, i.e. `__context__.mpls.0`. When the egress node fails, the PLR or the Protector will switch traffic to this backup next hop in PFE. The outer label (the transport LSP label) of packets is popped, and the inner label (the layer 3 VPN label allocated by the egress node) is looked up in `__context__.mpls.0`, which results in forwarding the packets directly to the CE (in Collocated protector model) or the backup PE (in Centralized protector model).

Topology

Figure 18 on page 420 shows the sample network.

Figure 18: Co-located PLR and protector in collocated protector model



Configuration

IN THIS SECTION

- CLI Quick Configuration | **421**
- Configuring Device CE1 | **425**
- Configuring Device PE1 | **425**
- Configuring Device P | **428**
- Configuring Device PE2 | **429**
- Configuring Device PE3 | **431**

Configuring Device CE2 | 434
Results | 434

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the `[edit]` hierarchy level.

Device CE1

```
set interfaces ge-0/0/0 unit 0 family inet address 10.10.20.2/30
set interfaces lo0 unit 0 family inet address 10.255.162.87/32
```

Device PE1

```
set interfaces ge-0/0/0 unit 0 family inet address 10.10.20.1/30
set interfaces ge-0/0/1 unit 0 family inet address 10.10.10.1/30
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 127.0.0.1/32
set interfaces lo0 unit 0 family inet address 10.255.162.84/32 primary
set interfaces lo0 unit 0 family iso address 47.0005.80ff.f800.0000.0108.0001.0102.5516.2084.00
set policy-options policy-statement vpn-exp term 1 from protocol direct
set policy-options policy-statement vpn-exp term 1 from route filter 10.10.20.0/24 exact
set policy-options policy-statement vpn-exp term 1 then community add vpn
set policy-options policy-statement vpn-exp term 1 then accept
set policy-options policy-statement vpn-imp term 1 from community vpn
set policy-options policy-statement vpn-imp term 1 then accept
set policy-options policy-statement vpn-imp term 2 then reject
set policy-options community vpn members traget:1:1
set routing-options autonomous-system 65000
set protocols rsvp interface all link-protection
set protocols rsvp interface fxp0.0 disable
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp vpn-apply-export
set protocols bgp group vpn type internal
```

```
set protocols bgp group vpn local-address 10.255.162.84
set protocols bgp group vpn family inet-vpn unicast
set protocols bgp group vpn neighbor 10.255.162.91
set protocols bgp group vpn neighbor 10.255.162.89
set protocols isis interface all
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0 passive
set routing-instances vpn instance-type vrf
set routing-instances vpn interface ge-1/0/0.0
set routing-instances vpn route-distinguisher 100:100
set routing-instances vpn vrf-import vpn-imp
set routing-instances vpn vrf-export vpn-exp
set routing-instances vpn vrf-table-label
set routing-instances vpn protocols bgp group vpn type external
set routing-instances vpn protocols bgp group vpn family inet unicast
set routing-instances vpn protocols bgp group vpn peer-as 65001
set routing-instances vpn protocols bgp group vpn as-override
set routing-instances vpn protocols bgp group vpn neighbor 10.10.20.2
```

Device P

```
set interfaces ge-0/0/0 unit 0 family inet address 10.10.11.2/30
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 10.10.10.2/30
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 127.0.0.1/32
set interfaces lo0 unit 0 family inet address 10.255.162.86/32 primary
set interfaces lo0 unit 0 family iso address 47.0005.80ff.f800.0000.0108.0001.0102.5516.2086.00
set protocols rsvp interface all link-protection
set protocols rsvp interface fxp0.0 disable
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols isis interface all
set protocols isis interface fxp0.0 disable
```

Device PE2

```
set interfaces ge-0/0/0 unit 0 family inet address 10.10.11.1/30
set interfaces ge-0/0/0 unit 0 family iso
```

```
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 10.10.12.1/30
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 10.10.30.1/30
set interfaces lo0 unit 0 family inet address 127.0.0.1/32
set interfaces lo0 unit 0 family inet address 10.255.162.91/32 primary
set interfaces lo0 unit 0 family iso address 47.0005.80ff.f800.0000.0108.0001.0102.5516.2091.00
set routing-options graceful-restart
set routing-options autonomous-system 65000
set routing-options forwarding-table export pplb
set protocols rsvp interface all link-protection
set protocols rsvp interface fxp0.0 disable
set protocols mpls label-switched-path to_PE1 to 10.255.162.84
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols mpls egress-protection context-identifier 10.1.1.1 protector
set protocols mpls egress-protection context-identifier 10.1.1.1 advertise-mode stub-alias
set protocols bgp vpn-apply-export
set protocols bgp group vpn type internal
set protocols bgp group vpn local-address 10.255.162.91
set protocols bgp group vpn family inet-vpn unicast egress-protection
set protocols bgp group vpn neighbor 10.255.162.84
set protocols bgp group vpn neighbor 10.255.162.89
set protocols isis traceoptions file isis.log
set protocols isis traceoptions flag all detail
set protocols isis level 2 disable
set protocols isis interface all
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0 passive
set policy-options policy-statement pplb term 1 then load-balance per-packet
set policy-options policy-statement vpn-exp term 1 from protocol bgp
set policy-options policy-statement vpn-exp term 1 then community add vpn
set policy-options policy-statement vpn-exp term 1 then accept
set policy-options policy-statement vpn-imp term 1 from community vpn
set policy-options policy-statement vpn-imp term 1 then accept
set policy-options policy-statement vpn-imp term 2 then reject
set policy-options community vpn members target:1:1
set routing-instances vpn instance-type vrf
set routing-instances vpn interface ge-3/2/4.0
set routing-instances vpn route-distinguisher 100:100
set routing-instances vpn vrf-import vpn-imp
set routing-instances vpn vrf-export vpn-exp
```

```
set routing-instances vpn vrf-table-label
set routing-instances vpn protocols bgp group vpn type external
set routing-instances vpn protocols bgp group vpn family inet unicast
set routing-instances vpn protocols bgp group vpn peer-as 65001
set routing-instances vpn protocols bgp group vpn as-override
set routing-instances vpn protocols bgp group vpn neighbor 10.10.30.2
```

Device PE3

```
set interfaces ge-0/0/0 unit 0 family inet address 10.10.40.1/30
set interfaces ge-0/0/1 unit 0 family inet address 10.10.12.2/30
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 127.0.0.1/32
set interfaces lo0 unit 0 family inet address 10.255.162.89/32 primary
set interfaces lo0 unit 0 family iso address 47.0005.80ff.f800.0000.0108.0001.0102.5516.2089.00
set routing-options graceful-restart
set routing-options autonomous-system 65000
set routing-options forwarding-table export pplb
set protocols rsvp interface all link-protection
set protocols rsvp interface fxp0.0 disable
set protocols mpls label-switched-path to_PE2 to 10.255.162.91
set protocols mpls label-switched-path to_PE1 to 10.255.162.84
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols mpls egress-protection context-identifier 10.1.1.1 primary
set protocols mpls egress-protection context-identifier 10.1.1.1 advertise-mode stub-alias
set protocols bgp vpn-apply-export
set protocols bgp group vpn type internal
set protocols bgp group vpn local-address 10.255.162.89
set protocols bgp group vpn family inet-vpn unicast
set protocols bgp group vpn neighbor 10.255.162.84 local-preference 300
set protocols bgp group vpn neighbor 10.255.162.91
set protocols isis level 2 disable
set protocols isis interface all
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0 passive
set routing-instances vpn instance-type vrf
set routing-instances vpn egress-protection context-identifier 10.1.1.1
set routing-instances vpn interface ge-1/1/0.0
set routing-instances vpn route-distinguisher 100:100
set routing-instances vpn vrf-import vpn-imp
```

```
set routing-instances vpn vrf-export vpn-exp
set routing-instances vpn vrf-table-label
set routing-instances vpn protocols bgp group vpn type external
set routing-instances vpn protocols bgp group vpn family inet unicast
set routing-instances vpn protocols bgp group vpn peer-as 65001
set routing-instances vpn protocols bgp group vpn as-override
set routing-instances vpn protocols bgp group vpn neighbor 10.10.40.2
```

Device CE2

```
set interfaces ge-0/0/0 unit 0 family inet address 10.10.40.2/30
set interfaces ge-0/0/2 unit 0 family inet address 10.10.30.2/30
set interfaces lo0 unit 0 family inet address 127.0.0.1/32
set interfaces lo0 unit 0 family inet address 10.255.162.88/32 primary
set interfaces lo0 unit 0 family iso address 47.0005.80ff.f800.0000.0108.0001.0102.5516.2088.00
```

Configuring Device CE1

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the Junos OS CLI User Guide.

1. Configure interfaces.

```
[edit interfaces]
user@CE1# set ge-0/0/0 unit 0 family inet address 10.10.20.2/30
user@CE1# set lo0 unit 0 family inet address 10.255.162.87/32
```

Configuring Device PE1

Step-by-Step Procedure

1. Configure the interfaces.

```
[edit interfaces]
user@PE1# set ge-0/0/0 unit 0 family inet address 10.10.20.1/30
user@PE1# set ge-0/0/1 unit 0 family inet address 10.10.10.1/30
```

```
user@PE1# set ge-0/0/1 unit 0 family iso
user@PE1# set ge-0/0/1 unit 0 family mpls
user@PE1# set lo0 unit 0 family inet address 127.0.0.1/32
user@PE1# set lo0 unit 0 family inet address 10.255.162.84/32 primary
user@PE1# set lo0 unit 0 family iso address 47.0005.80ff.f800.0000.0108.0001.0102.5516.2084.00
```

2. Configure the autonomous system (AS) number.

```
[edit routing-options]
user@PE1# set autonomous-system 65000
user@PE1# set forwarding-table export pplb
```

3. Configure RSVP.

```
[edit protocols rsvp]
user@PE1# set interface all link-protection
user@PE1# set interface fxp0.0 disable
```

4. Enable MPLS.

```
[edit protocols mpls]
user@PE1# set interface all
user@PE1# set interface fxp0.0 disable
```

5. Configure BGP.

```
[edit protocols bgp]
user@PE1# set group vpn type internal
user@PE1# set group vpn local-address 10.255.162.84
user@PE1# set group vpn family inet-vpn unicast
user@PE1# set group vpn neighbor 10.255.162.91
user@PE1# set group vpn neighbor 10.255.162.89
user@PE1# set vpn-apply-export
```

6. Enable IS-IS.

```
[edit protocols isis]
user@PE1# set interface all
```

```
user@PE1# set interface fxp0.0 disable
user@PE1# set interface lo0.0 passive
```

7. (Optional) Configure OSPF

```
[edit protocols ospf]
user@PE1# set area 0.0.0.0 interface all
user@PE1# set area 0.0.0.0 interface fxp0.0 disable
user@PE1# set area 0.0.0.0 interface lo0.0 passive
user@PE1# set traffic-engineering
```

8. Configure the routing instance.

```
[edit routing-instances]
user@PE1# set vpn instance-type vrf
user@PE1# set vpn interface ge-1/0/0.0
user@PE1# set vpn route-distinguisher 100:100
user@PE1# set vpn vrf-import vpn-imp
user@PE1# set vpn vrf-export vpn-exp
user@PE1# set vpn vrf-table-label
user@PE1# set vpn protocols bgp group vpn type external
user@PE1# set vpn protocols bgp group vpn family inet unicast
user@PE1# set vpn protocols bgp group vpn peer-as 65001
user@PE1# set vpn protocols bgp group vpn as-override
user@PE1# set vpn protocols bgp group vpn neighbor 10.10.20.2
```

9. Configure the routing policy.

```
[edit]
user@PE1# set policy-options policy-statement vpn-exp term 1 from protocol direct
user@PE1# set policy-options policy-statement vpn-exp term 1 from route filter 10.10.20.0/24
exact
user@PE1# set policy-options policy-statement vpn-exp term 1 then community add vpn
user@PE1# set policy-options policy-statement vpn-exp term 1 then accept
user@PE1# set policy-options policy-statement vpn-imp term 1 from community vpn
user@PE1# set policy-options policy-statement vpn-imp term 1 then accept
user@PE1# set policy-options policy-statement vpn-imp term 2 then reject
user@PE1# set policy-options community vpn members traget:1:1
```

Configuring Device P

Step-by-Step Procedure

1. Configure the device interfaces.

```
[edit interfaces]
user@P# set ge-0/0/0 unit 0 family inet address 10.10.11.2/30
user@P# set ge-0/0/0 unit 0 family iso
user@P# set ge-0/0/0 unit 0 family mpls
user@P# set ge-0/0/1 unit 0 family inet address 10.10.10.2/30
user@P# set ge-0/0/1 unit 0 family iso
user@P# set ge-0/0/1 unit 0 family mpls
user@P# set lo0 unit 0 family inet address 127.0.0.1/32
user@P# set lo0 unit 0 family inet address 10.255.162.86/32 primary
user@P# set lo0 unit 0 family iso address 47.0005.80ff.f800.0000.0108.0001.0102.5516.2086.00
```

2. Enable IS-IS.

```
[edit protocols isis]
user@P# set interface all
user@P# set interface fxp0.0 disable
```

3. Enable MPLS.

```
[edit protocols mpls ]
user@P# set interface all
user@P# set interface fxp0.0 disable
```

4. Configure RSVP.

```
[edit protocols rsvp]
user@P# set interface all link-protection
user@P# set interface fxp0.0 disable
```

5. (Optional) Configure OSPF.

```
[edit protocols ospf]
user@P# set area 0.0.0.0 interface all
user@P# set area 0.0.0.0 interface fxp0.0 disable
user@P# set area 0.0.0.0 interface lo0.0 passive
user@P# set traffic-engineering
```

Configuring Device PE2

Step-by-Step Procedure

1. Configure the interfaces.

```
[edit interfaces]
user@PE2# set ge-0/0/0 unit 0 family inet address 10.10.11.1/30
user@PE2# set ge-0/0/0 unit 0 family iso
user@PE2# set ge-0/0/0 unit 0 family mpls
user@PE2# set ge-0/0/1 unit 0 family inet address 10.10.12.1/30
user@PE2# set ge-0/0/1 unit 0 family iso
user@PE2# set ge-0/0/1 unit 0 family mpls
user@PE2# set ge-0/0/2 unit 0 family inet address 10.10.30.1/30
user@PE2# set lo0 unit 0 family inet address 127.0.0.1/32
user@PE2# set lo0 unit 0 family inet address 10.255.162.91/32 primary
user@PE2# set lo0 unit 0 family iso address 47.0005.80ff.f800.0000.0108.0001.0102.5516.2091.00
```

2. Configure autonomous number(AS).

```
[edit routing-options]
user@PE2# set autonomous-system 65000
user@PE2# set forwarding-table export pplb
```

3. Configure RSVP.

```
[edit protocols rsvp]
user@PE2# set interface all link-protection
user@PE2# set interface fxp0.0 disable
```

4. Configure MPLS.

```
[edit protocols mpls]
user@PE2# set label-switched-path to_PE1 to 10.255.162.84
user@PE2# set interface all
user@PE2# set interface fxp0.0 disable
user@PE2# set egress-protection context-identifier 10.1.1.1 protector
user@PE2# set egress-protection context-identifier 10.1.1.1 advertise-mode stub-alias
```

5. Configure BGP.

```
[edit protocols bgp]
user@PE2# set group vpn family inet-vpn unicast egress-protection
user@PE2# set group vpn local-address 10.255.162.91
user@PE2# set group vpn neighbor 10.255.162.84
user@PE2# set group vpn neighbor 10.255.162.89
user@PE2# set group vpn type internal
user@PE2# set vpn-apply-export
```

6. Configure IS-IS.

```
[edit protocols isis]
user@PE2# set interface all
user@PE2# set interface fxp0.0 disable
user@PE2# set interface lo0.0 passive
user@PE2# set level 2 disable
user@PE2# set traceoptions file isis.log
user@PE2# set traceoptions flag all detail
```

7. (Optional) Configure OSPF.

```
[edit protocols ospf]
user@PE2# set area 0.0.0.0 interface all
user@PE2# set area 0.0.0.0 interface fxp0.0 disable
user@PE2# set area 0.0.0.0 interface lo0.0 passive
user@PE2# set traffic-engineering
```

8. Configure the routing policy.

```
[edit policy-options]
user@PE2# set community vpn members target:1:1
user@PE2# set policy-statement pplb term 1 then load-balance per-packet
user@PE2# set policy-statement vpn-exp term 1 from protocol bgp
user@PE2# set policy-statement vpn-exp term 1 then community add vpn
user@PE2# set policy-statement vpn-exp term 1 then accept
user@PE2# set policy-statement vpn-imp term 1 from community vpn
user@PE2# set policy-statement vpn-imp term 1 then accept
user@PE2# set policy-statement vpn-imp term 2 then reject
```

9. Configure the routing instance.

```
[edit routing-instances]
user@PE2# set vpn instance-type vrf
user@PE2# set vpn interface ge-3/2/4.0
user@PE2# set vpn route-distinguisher 100:100
user@PE2# set vpn vrf-import vpn-imp
user@PE2# set vpn vrf-export vpn-exp
user@PE2# set vpn vrf-table-label
user@PE2# set vpn protocols bgp group vpn type external
user@PE2# set vpn protocols bgp group vpn family inet unicast
user@PE2# set vpn protocols bgp group vpn peer-as 65001
user@PE2# set vpn protocols bgp group vpn as-override
user@PE2# set vpn protocols bgp group vpn neighbor 10.10.30.2
```

Configuring Device PE3

Step-by-Step Procedure

1. Configure the interfaces.

```
[edit interfaces]
user@PE3# set ge-0/0/0 unit 0 family inet address 10.10.40.1/30
user@PE3# set ge-0/0/1 unit 0 family inet address 10.10.12.2/30
user@PE3# set ge-0/0/1 unit 0 family iso
user@PE3# set ge-0/0/1 unit 0 family mpls
user@PE3# set lo0 unit 0 family inet address 127.0.0.1/32
```

```
user@PE3# set lo0 unit 0 family inet address 10.255.162.89/32 primary
user@PE3# set lo0 unit 0 family iso address 47.0005.80ff.f800.0000.0108.0001.0102.5516.2089.00
```

2. Configure the autonomous number (AS).

```
[edit routing-options]
user@PE3# set autonomous-system 65000
user@PE3# set forwarding-table export pplb
```

3. Configure RSVP.

```
[edit protocols rsvp]
user@PE3# set interface all link-protection
user@PE3# set interface fxp0.0 disable
```

4. Configure MPLS.

```
[edit protocols mpls]
user@PE3# set interface all
user@PE3# set interface fxp0.0 disable
user@PE3# set egress-protection context-identifier 10.1.1.1 primary
user@PE3# set egress-protection context-identifier 10.1.1.1 advertise-mode stub-alias
user@PE3# set label-switched-path to_PE2 to 10.255.162.91
user@PE3# set label-switched-path to_PE1 to 10.255.162.84
```

5. Configure BGP.

```
[edit protocols bgp]
user@PE3# set group vpn type internal
user@PE3# set group vpn local-address 10.255.162.89
user@PE3# set group vpn family inet-vpn unicast
user@PE3# set group vpn neighbor 10.255.162.84 local-preference 300
user@PE3# set group vpn neighbor 10.255.162.91
user@PE3# set vpn-apply-export
```

6. Configure IS-IS.

```
[edit protocols isis]
user@PE3# set interface all
user@PE3# set interface fxp0.0 disable
user@PE3# set interface lo0.0 passive
user@PE3# set level 2 disable
```

7. (Optional) Configure OSPF.

```
[edit protocols ospf]
user@PE3# set area 0.0.0.0 interface all
user@PE3# set area 0.0.0.0 interface fxp0.0 disable
user@PE3# set area 0.0.0.0 interface lo0.0 passive
user@PE3# set traffic-engineering
```

8. Configure the routing instance.

```
[edit routing-instances]
user@PE3# set vpn egress-protection context-identifier 10.1.1.1
user@PE3# set vpn instance-type vrf
user@PE3# set vpn interface ge-1/1/0.0
user@PE3# set vpn protocols bgp group vpn type external
user@PE3# set vpn protocols bgp group vpn family inet unicast
user@PE3# set vpn protocols bgp group vpn peer-as 65001
user@PE3# set vpn protocols bgp group vpn as-override
user@PE3# set vpn protocols bgp group vpn neighbor 10.10.40.2
user@PE3# set vpn route-distinguisher 100:100
user@PE3# set vpn vrf-export vpn-exp
user@PE3# set vpn vrf-import vpn-imp
user@PE3# set vpn vrf-table-label
```

Configuring Device CE2

Step-by-Step Procedure

1. Configure the interfaces.

```
[edit interfaces]
user@CE2# set ge-0/0/0 unit 0 family inet address 10.10.40.2/30
user@CE2# set ge-0/0/2 unit 0 family inet address 10.10.30.2/30
user@CE2# set lo0 unit 0 family inet address 127.0.0.1/32
user@CE2# set lo0 unit 0 family inet address 10.255.162.88/32 primary
user@CE2# set lo0 unit 0 family iso address 47.0005.80ff.f800.0000.0108.0001.0102.5516.2088.00
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

Device CE1

```
user@CE1# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 10.10.20.2/30;
        }
    }
}
```

Device PE1

```
user@PE1# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 10.10.20.1/30;
        }
    }
}
```

```
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.10.10.1/30;
        }
        family iso;
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 127.0.0.1/32;
            address 10.255.162.84/32 {
                primary;
            }
        }
        family iso {
            address 47.0005.80ff.f800.0000.0108.0001.0102.5516.2084.00;
        }
    }
}
```

```
user@PE1# show protocols
rsvp {
    interface all {
        link-protection;
    }
    interface fxp0.0 {
        disable;
    }
}
mpls {
    interface all;
    interface fxp0.0 {
        disable;
    }
}
bgp {
    vpn-apply-export;
    group vpn {
```

```
        type internal;
        local-address 10.255.162.84;
        family inet-vpn {
            unicast;
        }
        neighbor 10.255.162.91;
        neighbor 10.255.162.89;
    }
}
isis {
    interface all;
    interface fxp0.0 {
        disable;
    }
    interface lo0.0 {
        passive;
    }
}
```

Device P

```
user@P# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 10.10.11.2/30;
        }
        family iso;
        family mpls;
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.10.10.2/30;
        }
        family iso;
        family mpls;
    }
}
lo0 {
    unit 0 {
```

```
        family inet {
            address 127.0.0.1/32;
            address 10.255.162.86/32 {
                primary;
            }
        }
        family iso {
            address 47.0005.80ff.f800.0000.0108.0001.0102.5516.2086.00;
        }
    }
}
```

```
user@P# show protocols
rsvp {
    interface all {
        link-protection;
    }
    interface fxp0.0 {
        disable;
    }
}
mpls {
    interface all;
    interface fxp0.0 {
        disable;
    }
}
isis {
    interface all;
    interface fxp0.0 {
        disable;
    }
}
```

Device PE2

```
user@PE2# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 10.10.11.1/30;
```

```
        }
        family iso;
        family mpls;
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.10.12.1/30;
        }
        family iso;
        family mpls;
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 10.10.30.1/30;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 127.0.0.1/32;
            address 10.255.162.91/32 {
                primary;
            }
        }
        family iso {
            address 47.0005.80ff.f800.0000.0108.0001.0102.5516.2091.00;
        }
    }
}
```

```
user@PE2# show protocols
rsvp {
    interface all {
        link-protection;
    }
    interface fxp0.0 {
```

```
        disable;
    }
}
mpls {
    label-switched-path to_PE1 {
        to 10.255.162.84;
    }
    interface all;
    interface fxp0.0 {
        disable;
    }
    egress-protection {
        context-identifier 10.1.1.1 {
            protector;
            advertise-mode stub-alias;
        }
    }
}
bgp {
    vpn-apply-export;
    group vpn {
        type internal;
        local-address 10.255.162.91;
        family inet-vpn {
            unicast {
                egress-protection;
            }
        }
        neighbor 10.255.162.84;
        neighbor 10.255.162.89;
    }
}
isis {
    traceoptions {
        file isis.log;
        flag all detail;
    }
    level 2 disable;
    interface all;
    interface fxp0.0 {
        disable;
    }
    interface lo0.0 {
```

```
        passive;
    }
}
```

Device PE3

```
user@PE3# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 10.10.40.1/30;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.10.12.2/30;
        }
        family iso;
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 127.0.0.1/32;
            address 10.255.162.89/32 {
                primary;
            }
        }
        family iso {
            address 47.0005.80ff.f800.0000.0108.0001.0102.5516.2089.00;
        }
    }
}
```

```
user@PE3# show protocols
rsvp {
    interface all {
        link-protection;
```

```
    }
    interface fxp0.0 {
        disable;
    }
}
mpls {
    label-switched-path to_PE2 {
        to 10.255.162.91;
    }
    label-switched-path to_PE1 {
        to 10.255.162.84;
    }
    interface all;
    interface fxp0.0 {
        disable;
    }
    egress-protection {
        context-identifier 10.1.1.1 {
            primary;
            advertise-mode stub-alias;
        }
    }
}
bgp {
    vpn-apply-export;
    group vpn {
        type internal;
        local-address 10.255.162.89;
        family inet-vpn {
            unicast;
        }
        neighbor 10.255.162.84 {
            local-preference 300;
        }
        neighbor 10.255.162.91;
    }
}
isis {
    level 2 disable;
    interface all;
    interface fxp0.0 {
        disable;
    }
```

```
    interface lo0.0 {
        passive;
    }
}
```

Device CE2

```
user@CE2# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 10.10.40.2/30;
        }
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 10.10.30.2/30;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 127.0.0.1/32;
            address 10.255.162.88/32 {
                primary;
            }
        }
        family iso {
            address 47.0005.80ff.f800.0000.0108.0001.0102.5516.2088.00;
        }
    }
}
```

Verification

IN THIS SECTION

- Verifying the Routing Instance | 443
- Checking the Context Identifier Route | 451

Verifying the Routing Instance

Purpose

Check the routes in the routing table.

Action

```
user@PE1> show route 10.10.50  table vpn.inet.0
vpn.inet.0: 6 destinations, 7 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.50.0/24      *[BGP/170] 00:01:26, localpref 100, from 10.255.162.96
                      AS path: 65001 I, validation-state: unverified
                    > to 10.10.10.2 via ge-2/0/2.0, Push 16, Push 300064(top)
                    [BGP/170] 00:06:22, localpref 50, from 10.255.162.91
                      AS path: 65001 I, validation-state: unverified
                    > to 10.10.10.2 via ge-2/0/2.0, Push 17, Push 299920(top)
```

```
user@PE1>show route 10.10.50 extensive table vpn.inet.0

vpn.inet.0: 6 destinations, 7 routes (6 active, 0 holddown, 0 hidden)
10.10.50.0/24 (2 entries, 1 announced)
TSI:
KRT in-kernel 10.10.50.0/24 -> {indirect(1048575)}
Page 0 idx 1, (group vpn type External) Type 1 val 0x9e33490 (adv_entry)
   Advertised metrics:
     Nexthop: Self
     AS path: [65000] 65000 I
     Communities: target:1:1
```

```
Path 10.10.50.0 from 10.255.162.96 Vector len 4.  Val: 1
       *BGP    Preference: 170/-101
                Route Distinguisher: 200:100
                Next hop type: Indirect, Next hop index: 0
                Address: 0x9db63f0
                Next-hop reference count: 6
                Source: 10.255.162.96
                Next hop type: Router, Next hop index: 635
                Next hop: 10.10.10.2 via ge-2/0/2.0, selected
                Label operation: Push 16, Push 300064(top)
                Label TTL action: prop-ttl, prop-ttl(top)
                Load balance label: Label 16: None; Label 300064: None;
                Label element ptr: 0x9db60e0
                Label parent element ptr: 0x9db5e40
                Label element references: 1
                Label element child references: 0
                Label element lsp id: 0
                Session Id: 0x146
                Protocol next hop: 10.1.1.1
                Label operation: Push 16
                Label TTL action: prop-ttl
                Load balance label: Label 16: None;
                Indirect next hop: 0x9e55440 1048575 INH Session ID: 0x14d
                State: < Secondary Active Int Ext ProtectionCand >
                Local AS: 65000 Peer AS: 65000
                Age: 1:28     Metric2: 1
                Validation State: unverified
                Task: BGP_65000.10.255.162.96
                Announcement bits (2): 0-KRT 1-BGP_RT_Background
                AS path: 65001 I
                Communities: target:1:1
                Import Accepted
                VPN Label: 16
                Localpref: 100
                Router ID: 10.255.162.96
                Primary Routing Table bgp.l3vpn.0
                Indirect next hops: 1
                        Protocol next hop: 10.1.1.1 Metric: 1
                        Label operation: Push 16
                        Label TTL action: prop-ttl
                        Load balance label: Label 16: None;
                        Indirect next hop: 0x9e55440 1048575 INH Session ID: 0x14d
                        Indirect path forwarding next hops: 1
```

```
                                Next hop type: Router
                                Next hop: 10.10.10.2 via ge-2/0/2.0
                                Session Id: 0x146
            10.1.1.1/32 Originating RIB: inet.3
              Metric: 1             Node path count: 1
              Forwarding nexthops: 1
                Nexthop: 10.10.10.2 via ge-2/0/2.0
BGP    Preference: 170/-51
                Route Distinguisher: 100:100
                Next hop type: Indirect, Next hop index: 0
                Address: 0x9db6390
                Next-hop reference count: 5
                Source: 10.255.162.91
                Next hop type: Router, Next hop index: 636
                Next hop: 10.10.10.2 via ge-2/0/2.0, selected
                Label operation: Push 17, Push 299920(top)
                Label TTL action: prop-ttl, prop-ttl(top)
                Load balance label: Label 17: None; Label 299920: None;
                Label element ptr: 0x9db62c0
                Label parent element ptr: 0x9dc0d00
                Label element references: 1
                Label element child references: 0
                Label element lsp id: 0
                Session Id: 0x146
                Protocol next hop: 10.255.162.91
                Label operation: Push 17
                Label TTL action: prop-ttl
                Load balance label: Label 17: None;
                Indirect next hop: 0x9e55580 1048574 INH Session ID: 0x14c
                State: < Secondary Int Ext ProtectionCand >
                Inactive reason: Local Preference
                Local AS: 65000 Peer AS: 65000
                Age: 6:24     Metric2: 1
                Validation State: unverified
                Task: BGP_65000.10.255.162.91
                AS path: 65001 I
                Communities: target:1:1
                Import Accepted
                VPN Label: 17
                Localpref: 50
                Router ID: 10.255.162.91
                Primary Routing Table bgp.l3vpn.0
                Indirect next hops: 1
```

```
                        Protocol next hop: 10.255.162.91 Metric: 1
                        Label operation: Push 17
                        Label TTL action: prop-ttl
                        Load balance label: Label 17: None;
                        Indirect next hop: 0x9e55580 1048574 INH Session ID: 0x14c
                        Indirect path forwarding next hops: 1
                                Next hop type: Router
                                Next hop: 10.10.10.2 via ge-2/0/2.0
                                Session Id: 0x146
            10.255.162.91/32 Originating RIB: inet.3
              Metric: 1              Node path count: 1
              Forwarding nexthops: 1
                Nexthop: 10.10.10.2 via ge-2/0/2.0
```

```
user@PE2> show route table mpls.0
mpls.0: 15 destinations, 15 routes (15 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 00:23:33, metric 1
                      to table inet.0
0(S=0)             *[MPLS/0] 00:23:33, metric 1
                      to table mpls.0
1                  *[MPLS/0] 00:23:33, metric 1
                      Receive
2                  *[MPLS/0] 00:23:33, metric 1
                      to table inet6.0
2(S=0)             *[MPLS/0] 00:23:33, metric 1
                      to table mpls.0
13                 *[MPLS/0] 00:23:33, metric 1
                      Receive
17                 *[VPN/0] 00:23:33
                      to table vpn.inet.0, Pop
299856(S=0)        *[MPLS/0] 00:23:33
                      to table __10.1.1.1__.mpls.0
299904             *[LDP/9] 00:01:50, metric 1
                    > to 10.10.11.2 via xe-8/2/5.0, Pop
299904(S=0)        *[LDP/9] 00:01:50, metric 1
                    > to 10.10.11.2 via xe-8/2/5.0, Pop
299920             *[LDP/9] 00:01:50, metric 1
                    > to 10.10.11.2 via xe-8/2/5.0, Swap 299904
300016             *[LDP/9] 00:01:50, metric 1
```

```
                    > to 10.10.12.1 via ge-3/0/2.0, Pop
                      to table __10.1.1.1__.mpls.0
300016(S=0)        *[LDP/9] 00:01:50, metric 1
                    > to 10.10.12.1 via ge-3/0/2.0, Pop
                      to table __10.1.1.1__.mpls.0
300048             *[LDP/9] 00:01:50, metric 1
                    > to 10.10.12.1 via ge-3/0/2.0, Pop
300048(S=0)        *[LDP/9] 00:01:50, metric 1
                    > to 10.10.12.1 via ge-3/0/2.0, Pop
```

```
user@PE2> show route table __10.1.1.1__.mpls.0

__10.1.1.1__.mpls.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

16                 *[Egress-Protection/170] 00:22:57
                      to table __10.1.1.1-vpn__.inet.0
```

```
user@PE2> show route table __10.1.1.1__.mpls.0 extensive

__10.1.1.1__.mpls.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
16  (1 entry, 1 announced)
        State: < CalcForwarding >
TSI:
KRT in-kernel 16     /52 -> {Table}
        *Egress-Protection Preference: 170
                Next table: __10.1.1.1-vpn__.inet.0
                Next-hop index: 649
                Address: 0x9dc2690
                Next-hop reference count: 2
                State: < Active NoReadvrt ForwardingOnly Int Ext >
                Local AS: 65000
                Age: 22:59
                Validation State: unverified
                Task: Protection
                Announcement bits (1): 0-KRT
```

```
                AS path: I
                Protecting 2 routes
```

```
user@PE2> show route table __10.1.1.1-vpn__.inet.0
__10.1.1.1-vpn__.inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.30.0/24      *[Egress-Protection/170] 00:02:11
                      to table vpn.inet.0
10.10.50.0/24      *[Egress-Protection/170] 00:02:11
                    > to 10.10.30.2 via ge-3/2/4.0
```

```
user@PE2> show route table __10.1.1.1-vpn__.inet.0 extensive
__10.1.1.1-vpn__.inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
10.10.30.0/24 (1 entry, 1 announced)
        State: < CalcForwarding >
TSI:
KRT in-kernel 10.10.30.0/24 -> {Table}
        *Egress-Protection Preference: 170
                Next table: vpn.inet.0
                Next-hop index: 592
                Address: 0x9dc2630
                Next-hop reference count: 2
                State: < Active NoReadvrt ForwardingOnly Int Ext >
                Local AS: 65000
                Age: 2:13
                Validation State: unverified
                Task: Protection
                Announcement bits (1): 0-KRT
                AS path: I
                Backup route 10.10.30.0 table vpn.inet.0

10.10.50.0/24 (1 entry, 1 announced)
        State: < CalcForwarding >
TSI:
KRT in-kernel 10.10.50.0/24 -> {10.10.30.2}
        *Egress-Protection Preference: 170
                Next hop type: Router, Next hop index: 630
                Address: 0x9dc1d90
                Next-hop reference count: 7
```

```
                Next hop: 10.10.30.2 via ge-3/2/4.0, selected
                Session Id: 0x147
                State: < Active NoReadvrt ForwardingOnly Int Ext >
                Local AS: 65000
                Age: 2:13
                Validation State: unverified
                Task: Protection
                Announcement bits (1): 0-KRT
                AS path: I
                Backup route 10.10.50.0 table vpn.inet.0
```

```
user@PE2> show route table mpls.0 label 17
mpls.0: 15 destinations, 15 routes (15 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

17                 *[VPN/0] 00:25:06
                      to table vpn.inet.0, Pop
```

```
user@PE2> show route table mpls.0 label 17 extensive
mpls.0: 15 destinations, 15 routes (15 active, 0 holddown, 0 hidden)
17  (1 entry, 0 announced)
        *VPN    Preference: 0
                Next table: vpn.inet.0
                Next-hop index: 0
                Label operation: Pop
                Load balance label: None;
                Label element ptr: 0x9db3920
                Label parent element ptr: 0x0
                Label element references: 1
                Label element child references: 0
                Label element lsp id: 0
                Address: 0x9db3990
                Next-hop reference count: 1
                State: < Active NotInstall Int Ext >
                         Age: 25:30
                Validation State: unverified
```

```
                Task: RT
                AS path: I
```

```
user@PE3> show route table mpls.0
mpls.0: 11 destinations, 11 routes (11 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 00:24:16, metric 1
                      to table inet.0
0(S=0)             *[MPLS/0] 00:24:16, metric 1
                      to table mpls.0
1                  *[MPLS/0] 00:24:16, metric 1
                      Receive
2                  *[MPLS/0] 00:24:16, metric 1
                      to table inet6.0
2(S=0)             *[MPLS/0] 00:24:16, metric 1
                      to table mpls.0
13                 *[MPLS/0] 00:24:16, metric 1
                      Receive
16                 *[VPN/0] 00:24:15
                      to table vpn.inet.0, Pop
300096             *[LDP/9] 00:02:33, metric 1
                    > to 10.10.12.2 via ge-1/1/4.0, Swap 299920
300112             *[LDP/9] 00:02:33, metric 1
                    > to 10.10.12.2 via ge-1/1/4.0, Swap 299904
300128             *[LDP/9] 00:02:33, metric 1
                    > to 10.10.12.2 via ge-1/1/4.0, Pop
300128(S=0)        *[LDP/9] 00:02:33, metric 1
                    > to 10.10.12.2 via ge-1/1/4.0, Pop
```

```
user@PE3> show route table mpls.0 label 16
mpls.0: 11 destinations, 11 routes (11 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

16                 *[VPN/0] 00:24:22
                      to table vpn.inet.0, Pop
```

```
user@PE3> show route table mpls.0 label 16 extensive
mpls.0: 11 destinations, 11 routes (11 active, 0 holddown, 0 hidden)
```

```
16  (1 entry, 0 announced)
        *VPN    Preference: 0
                Next table: vpn.inet.0
                Next-hop index: 0
                Label operation: Pop
                Load balance label: None;
                Label element ptr: 0x31d1ec0
                Label parent element ptr: 0x0
                Label element references: 1
                Label element child references: 0
                Label element lsp id: 0
                Address: 0x31d1f30
                Next-hop reference count: 1
                State: < Active NotInstall Int Ext >
                Age: 24:24
                Validation State: unverified
                Task: RT
                AS path: I
```

Checking the Context Identifier Route

Purpose

Examine the information about the context identifier (10.1.1.1).

Action

```
user@PE1> show route 10.1.1.1
inet.0: 47 destinations, 47 routes (46 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.1.1/32         *[IS-IS/15] 00:04:08, metric 31
                    > to 10.10.10.2 via ge-2/0/2.0

inet.3: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.1.1/32         *[LDP/9] 00:04:08, metric 1
                    > to 10.10.10.2 via ge-2/0/2.0, Push 300064

inet.5: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
```

```
+ = Active Route, - = Last Active, * = Both

10.1.1.1/32         *[IS-IS/15] 00:04:08, metric 31, metric2 1
                    > to 10.10.10.2 via ge-2/0/2.0, Push 299856, Push 299920(top)
```

```
user@PE2> show route 10.1.1.1
inet.0: 48 destinations, 49 routes (47 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.1.1/32         *[MPLS/2] 00:26:00, metric 16777215
                      Receive
                    [IS-IS/15] 00:04:17, metric 11
                    > to 10.10.12.1 via ge-3/0/2.0

inet.3: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.1.1/32         *[LDP/9] 00:04:17, metric 1
                    > to 10.10.12.1 via ge-3/0/2.0
```

```
user@PE2> show mpls context-identifier
ID                Type        Metric     ContextTable
10.1.1.1           protector   16777215   __10.1.1.1__.mpls.0
Total 1, Primary 0, Protector 1
```

```
user@PE2> show mpls context-identifier detail
ID: 10.1.1.1
  Type: protector, Metric: 16777215, Mode: alias
  Context table: __10.1.1.1__.mpls.0, Label out: 299856

Total 1, Primary 0, Protector 1
```

```
user@PE3> show route 10.1.1.1
inet.0: 47 destinations, 47 routes (46 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.1.1/32         *[MPLS/1] 00:26:09, metric 1
```

```
                    Receive

inet.3: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.1.1/32         *[MPLS/1] 00:26:09, metric 1
                      Receive

inet.5: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.1.1/32         *[IS-IS/15] 00:04:27, metric 1, metric2 1
                    > to 10.10.12.2 via ge-1/1/4.0, Push 299856
```

```
user@PE3> show mpls context-identifier

ID                Type        Metric     ContextTable
10.1.1.1           primary     1
Total 1, Primary 1, Protector 0
```

```
user@PE3> show mpls context-identifier detail
ID: 10.1.1.1
  Type: primary, Metric: 1, Mode: alias

Total 1, Primary 1, Protector 0
```

Understanding MPLS and Path Protection on EX Series Switches

Junos OS MPLS for Juniper Networks EX Series Ethernet Switches provides path protection to protect your MPLS network from label switched path (LSP) failures.

By default, an LSP routes itself hop-by-hop from the ingress provider edge switch through the provider switches toward the egress provider edge switch. The LSP generally follows the shortest path as dictated by the local routing table, usually taking the same path as destination-based, best-effort traffic. These paths are “soft” in nature because they automatically reroute themselves whenever a change occurs in a routing table or in the status of a node or link.

Typically, when an LSP fails, the switch immediately upstream from the failure signals the outage to the ingress provider edge switch. The ingress provider edge switch calculates a new path to the egress provider edge switch, establishes the new LSP, and then directs traffic from the failed path to the new path. This rerouting process can be time-consuming and prone to failure. For example, the outage

signals to the ingress switch might get lost or the new path might take too long to come up, resulting in significant packet drops.

You can configure path protection by configuring primary and secondary paths on the ingress switch. If the primary path fails, the ingress switch immediately reroutes traffic from the failed path to the standby path, eliminating the need for the ingress switch to calculate a new route and signal a new path. For information about configuring standby LSPs, see "Configuring Path Protection in an MPLS Network (CLI Procedure)" on page 364.

Verifying Path Protection in an MPLS Network

IN THIS SECTION

- Verifying the Primary Path | **454**
- Verifying the RSVP-Enabled Interfaces | **456**
- Verifying a Secondary Path | **456**

To verify that path protection is working correctly on EX Series switches, perform the following tasks:

Verifying the Primary Path

IN THIS SECTION

- Purpose | **454**
- Action | **454**
- Meaning | **455**

Purpose

Verify that the primary path is operational.

Action

```
user@switch> show mpls lsp extensive ingress
```

```
Ingress LSP: 2 sessions

127.1.8.8
  From: 127.1.9.9, State: Up, ActiveRoute: 0, LSPname: lsp_to_240
  ActivePath: primary_path_lsp_to_240 (primary)
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary   primary_path_lsp_to_240 State: Up
    Priorities: 7 0
    SmartOptimizeTimer: 180
          Exclude: red
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 2)
 10.3.3.2 S 10.3.4.2 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
          10.3.3.2 10.3.4.2
    6 Mar 11 23:58:01.684 Selected as active path: due to 'primary'
    5 Mar 11 23:57:00.750 Record Route:  10.3.3.2 10.3.4.2
    4 Mar 11 23:57:00.750 Up
    3 Mar 11 23:57:00.595 Originate Call
    2 Mar 11 23:57:00.595 CSPF: computation result accepted  10.3.3.2 10.3.4.2
    1 Mar 11 23:56:31.135 CSPF failed: no route toward 10.3.2.2[25 times]
Standby   secondary_path_lsp_to_240 State: Up
Standby   secondary_path_lsp_to_240 State: Up
    Priorities: 7 0
    SmartOptimizeTimer: 180
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 1)
 10.3.5.2 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
         10.3.5.2
    7 Mar 11 23:58:01.684 Deselected as active: due to 'primary'
    6 Mar 11 23:46:17.298 Selected as active path
    5 Mar 11 23:46:17.295 Record Route:  5.5.5.2
    4 Mar 11 23:46:17.287 Up
    3 Mar 11 23:46:16.760 Originate Call
    2 Mar 11 23:46:16.760 CSPF: computation result accepted  10.3.5.2
    1 Mar 11 23:45:48.095 CSPF failed: no route toward 10.5.5.5[2 times]
  Created: Wed Mar 11 23:44:37 2009
 [Output truncated]
```

Meaning

As indicated by the **ActivePath** in the output, the LSP **primary_path_lsp_to_240** is active.

Verifying the RSVP-Enabled Interfaces

IN THIS SECTION

- Purpose | 456
- Action | 456
- Meaning | 456

Purpose

Verify the status of Resource Reservation Protocol (RSVP)-enabled interfaces and packet statistics.

Action

```
user@switch> show rsvp interfaces

RSVP interface: 1 active
                  Active Subscr- Static      Available   Reserved    Highwater
Interface   State resv   iption  BW          BW          BW          mark
ge-0/0/20.0 Up         2   100%  1000Mbps    1000Mbps    0bps        0bps
```

Meaning

This output verifies that RSVP is enabled and operational on interface **ge-0/0/20.0**.

Verifying a Secondary Path

IN THIS SECTION

- Purpose | 457
- Action | 457
- Meaning | 458

Purpose

Verify that a secondary path is established.

Action

Deactivate a switch that is critical to the primary path and then issue the following command:

```
user@switch> show mpls lsp extensive

Ingress LSP: 1 sessions

127.0.0.8
  From: 127.0.0.1, State: Up, ActiveRoute: 0, LSPname: lsp_to_240
  ActivePath: secondary_path_lsp_to_240 (secondary)
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
  Primary   primary_path_lsp_to_240 State: Dn
    Priorities: 7 0
    SmartOptimizeTimer: 180
          Exclude: red
    Will be enqueued for recomputation in 8 second(s).
   51 Mar  8 12:23:31.268 CSPF failed: no route toward 127.0.0.11[11420 times]
   50 Mar  4 15:35:25.610 Clear Call: CSPF computation failed
   49 Mar  4 15:35:25.610 CSPF: link down/deleted: 127.0.0.2(127.0.0.1:0)(127.0.0.1)->
0.0.0.0(127.0.0.20:0)(127.0.0.20)
   48 Mar  4 15:35:25.576 Deselected as active
   47 Mar  4 15:35:25.550 No Route toward dest
   46 Mar  4 15:35:25.550 ?????
   45 Mar  4 15:35:25.549 127.0.0.12: Down
   44 Mar  4 15:33:29.839 Selected as active path
   43 Mar  4 15:33:29.837 Record Route:  127.0.0.20 127.0.0.40
   42 Mar  4 15:33:29.835 Up
   41 Mar  4 15:33:29.756 Originate Call
   40 Mar  4 15:33:29.756 CSPF: computation result accepted  127.0.0.20 127.0.0.40
   39 Mar  4 15:33:00.395 CSPF failed: no route toward 127.0.0.11[7 times]
   38 Mar  4 15:30:31.412 Clear Call: CSPF computation failed
   37 Mar  4 15:30:31.412 CSPF: link down/deleted: 127.0.0.2(127.0.0.1:0)(127.0.0.1)->
0.0.0.0(127.0.0.20:0)(127.0.0.20)
   36 Mar  4 15:30:31.379 Deselected as active
   35 Mar  4 15:30:31.350 No Route toward dest
   34 Mar  4 15:30:31.350 ?????
```

```
    33 Mar  4 15:30:31.349 127.0.0.12: Down
    32 Mar  4 15:29:05.802 Selected as active path
    31 Mar  4 15:29:05.801 Record Route:  127.0.0.20 127.0.0.40
    30 Mar  4 15:29:05.801 Up
    29 Mar  4 15:29:05.686 Originate Call
    28 Mar  4 15:29:05.686 CSPF: computation result accepted  127.0.0.20 127.0.0.40
    27 Mar  4 15:28:35.852 CSPF failed: no route toward 127.0.0.11[132 times]
    26 Mar  4 14:25:12.113 Clear Call: CSPF computation failed
    25 Mar  4 14:25:12.113 CSPF: link down/deleted: 0.0.0.0(127.0.0.20:0)(127.0.0.20)->
0.0.0.0(10.10.10.10:0)(10.10.10.10)
 *Standby   secondary_path_lsp_to_240 State: Up
    Priorities: 7 0
    SmartOptimizeTimer: 180
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 1)
 [Output truncated]
```

Meaning

As indicated by the **ActivePath** in the output, the LSP **secondary_path_lsp_to_240** is active.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
15.1	Starting in Junos OS Release 15.1, the enhanced point of local repair (PLR) functionality addresses a special scenario of egress node protection, where the PLR and the protector are co-located as one router. In this case, there is no need to have a bypass LSP reroute traffic during local repair.
14.2	Starting in Junos OS Release 14.2, Junos OS supports the restoration of egress traffic when there is a link or node failure in the egress PE node.
14.2	Starting in Junos OS Release 14.2, Junos OS supports the restoration of egress traffic when there is a link or node failure in the egress PE node.

RELATED DOCUMENTATION

Basic MPLS Configuration | 46

Link Protection for MPLS LSPs

IN THIS SECTION

- Link Protection | **459**
- Multiple Bypass LSPs for Link Protection | **460**
- Node Protection | **461**
- Bandwidth Protection for Links and Nodes (Dynamic and Static) | **462**
- Fast Reroute, Node Protection, and Link Protection | **463**
- Configuring Link Protection on Interfaces Used by LSPs | **468**
- Configuring Node Protection or Link Protection for LSPs | **478**
- Configuring Inter-AS Node and Link Protection | **478**
- Configuring Constraint Aware Bypass LSPs | **479**

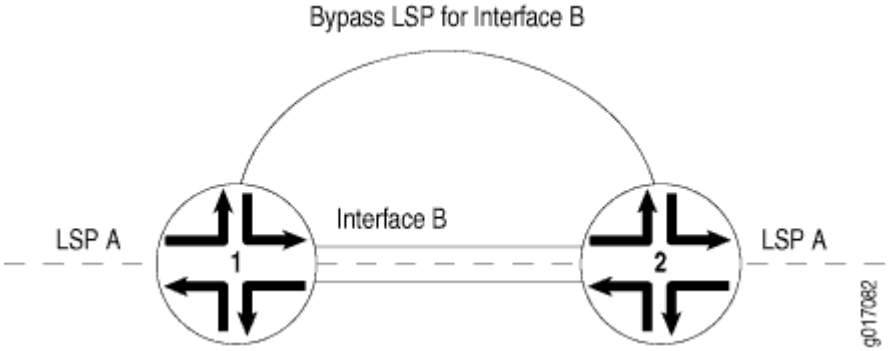
Link Protection

Link protection helps to ensure that traffic going over a specific interface to a neighboring router or switch can continue to reach this router (switch) if that interface fails. When link protection is configured for an interface and an LSP that traverses this interface, a bypass LSP is created that will handle this traffic if the interface fails. The bypass LSP uses a different interface and path to reach the same destination. The path used can be configured explicitly, or you can rely on CSPF. The RSVP metric for the bypass LSP is set in the range of 20,000 through 29,999 (this value is not user configurable).

If a link-protected interface fails, traffic is quickly switched to the bypass LSP. Note that a bypass LSP cannot share the same egress interface with the LSPs it monitors.

In Figure 19 on page 460, link protection is enabled on Interface B between Router 1 and Router 2. It is also enabled on LSP A, an LSP that traverses the link between Router 1 and Router 2. If the link between Router 1 and Router 2 fails, traffic from LSP A is quickly switched to the bypass LSP generated by link protection.

Figure 19: Link Protection Creating a Bypass LSP for the Protected Interface



Although LSPs traversing an interface can be configured to take advantage of link protection, it is important to note that it is specifically the interface that benefits from link protection. If link protection is enabled on an interface but not on a particular LSP traversing that interface, then if the interface fails, that LSP will also fail.

NOTE: Link protection does not work on unnumbered interfaces.

To protect traffic over the entire route taken by an LSP, you should configure fast reroute. For more information, see "Configuring Fast Reroute" on page 580.

Multiple Bypass LSPs for Link Protection

By default, link protection relies on a single bypass LSP to provide path protection for an interface. However, you can also specify multiple bypass LSPs to provide link protection for an interface. You can individually configure each of these bypass LSPs or create a single configuration for all of the bypass LSPs. If you do not configure the bypass LSPs individually, they all share the same path and bandwidth constraints.

The following algorithm describes how and when an additional bypass LSP is activated for an LSP:

1. If any currently active bypass can satisfy the requirements of the LSP (bandwidth, link protection, or node-link protection), the traffic is directed to that bypass.

2. If no active bypass LSP is available, scan through the manual bypass LSPs in first-in, first-out (FIFO) order, skipping those that are already active (each manual bypass can only be activated once). The first inactive manual bypass that can satisfy the requirements is activated and traffic is directed to that bypass.

3. If no manual bypass LSPs are available and if the `max-bypasses` statement activates multiple bypass LSPs for link protection, determine whether an automatically configured bypass LSP can satisfy the requirements. If an automatically configured bypass LSP is available and if the total number of active

automatically configured bypass LSPs does not exceed the maximum bypass LSP limit (configured with the `max-bypasses` statement), activate another bypass LSP.

For information about how to configure multiple bypass LSPs for link protection, see "Configuring Bypass LSPs" on page 468.

Node Protection

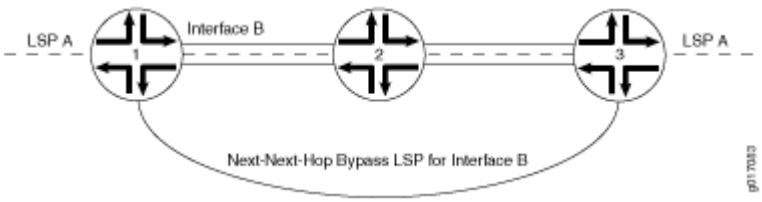
Node protection extends the capabilities of link protection. Link protection helps to ensure that traffic going over a specific interface to a neighboring router can continue to reach this router if that interface fails. Node protection ensures that traffic from an LSP traversing a neighboring router can continue to reach its destination even if the neighboring router fails.

When you enable node protection for an LSP, you must also enable link protection. Once enabled, node protection and link protection establish the following types of bypass LSPs:

- Next-hop bypass LSP—Provides an alternate route for an LSP to reach a neighboring router. This type of bypass LSP is established when you enable either node protection or link protection.
- Next-next-hop bypass LSP—Provides an alternate route for an LSP to get around a neighboring router en route to the destination router. This type of bypass LSP is established exclusively when node protection is configured. If a next-next-hop bypass LSP cannot be created, an attempt is made to signal a next-hop bypass LSP.

In Figure 20 on page 461, node protection is enabled on Interface B on Router 1. Node protection is also enabled on LSP A, an LSP that traverses the link transiting Router 1, Router 2, and Router 3. If Router 2 suffers a hardware or software failure, traffic from LSP A is switched to the next-next-hop bypass LSP generated by node protection.

Figure 20: Node Protection Creating a Next-Next-Hop Bypass LSP



The time needed by node protection to switch traffic to a next-next-hop bypass LSP can be significantly longer than the time needed by link protection to switch traffic to a next-hop bypass LSP. Link protection relies on a hardware mechanism to detect a link failure, allowing it to quickly switch traffic to a next-hop bypass LSP.

Node failures are often due to software problems on the node router. Node protection relies on the receipt of hello messages from a neighboring router to determine whether it is still functioning. The time it takes node protection to divert traffic partly depends on how often the node router sends hello

messages and how long it takes the node-protected router to react to having not received a hello message. However, once the failure is detected, traffic can be quickly diverted to the next-next-hop bypass LSP.

NOTE: Node protection provides traffic protection in the event of an error or interruption of the physical link between two routers. It does not provide protection in the event of control plane errors. The following provides an example of a control plane error:

- A transit router changes the label of a packet due to a control plane error.
- When the ingress router receives the packet, it considers the label change to be a catastrophic event and deletes both the primary LSP and the associated bypass LSP.

Bandwidth Protection for Links and Nodes (Dynamic and Static)

IN THIS SECTION

Overview | 462

Overview

When configuring an LSP, you can specify a bandwidth to reserve for protection for that path. This guarantees that the LSP has the necessary capacity on its links it traverses during local repair. Junos OS supports configuring bypass (backup) LSPs that protect the primary LSP against link or node failures. These backup LSPs can have their bandwidth reserved, ensuring that sufficient bandwidth is available for rerouted traffic during failures.

For example, if your network has an LSP that is going through some hops (nodes) and if 100% bandwidth protection is desired, then each hop will ideally reserve a bypass for 100% capacity of the protected LSP. In this case, the overall network capacity required will be significantly high. You need to decide how many bypass LSPs you require, the capacity for each bypass LSP, and the ratio of traffic (or bandwidth percentage) that needs protection. You can either choose a specific path or hop(s) that the bypass should traverse by configuring static bypass LSPs or use one or more auto-created dynamic bypasses by configuring dynamic bypass LSPs.

Use the following formula to configure bypass LSPs for bandwidth protection:

- *interface-subscription[i] * interface-link-bandwidth[i] <= bypass-n-bandwidth[i] * bypass-n-subscription [i]—*

When you configure static bypass LSPs, sum of **interface subscription** and **interface link bandwidth** must not exceed the sum of configured ***bypass-n-bandwidth*** and ***bypass-n-subscription***. For example, (*bypass-1-bandwidth*, *bypass-1-subscription*, *bypass-2-bandwidth, bypass-2-subscription*, and so on).

- *interface-subscription[i] * interface-link-bandwidth[i] <= max-bypasses[i] * bypass-bandwidth[i] * bypass-subscription[i]*—

  When you configure dynamic bypass LSPs, the sum of **interface subscription** and **interface link bandwidth** must not exceed the sum of configured **maximum bypass**, **bypass bandwidth**, and the **bypass subscription**.

Table 10: Description

Value	Description
interface-subscription	Total interface bandwidth
interface-link-bandwidth	Link bandwidth of interface
max-bypasses	The maximum number of bypasses that will be created for dynamic bypass LSPs. The bandwidth value configured will be equally distributed over LSPs, per the max-bypass value.
bypass-bandwidth	The bandwidth (in Mbps) you can configure for a given bypass LSP. If you are configuring manual bypass LSPs, you can configure different bypass-bandwidth for each LSP.
bypass-subscription	Set the total ratio of traffic you wish to enable protection for. Set the bypass-bandwidth value to fit the planned bypass-subscription.

Fast Reroute, Node Protection, and Link Protection

IN THIS SECTION

- LSP Protection Overview | 464
- LSP Protection Types Comparison | 464

One-to-One Backup Implementation | 465
Facility Backup Implementation | 466

This document discusses the following sections:

LSP Protection Overview

RSVP-TE extensions establish backup label-switched path (LSP) tunnels for local repair of LSP tunnels. These mechanisms enable immediate re-direction of traffic onto backup LSP tunnels, in the event of a failure.

RFC 4090, *Fast Reroute Extensions to RSVP-TE for LSP Tunnels*, describes two different types of traffic protection for RSVP-signaled LSPs:

- One-to-one backup—In this method, detour LSPs for each protected LSP is created at each potential point of local repair.
- Facility backup—In this method, a bypass tunnel is created to protect a set of LSPs that have similar backup constraints at a potential failure point, by taking advantage of the MPLS label stacking.

The one-to-one backup and the facility backup methods protect links and nodes during network failure, and can co-exist in a mixed network.

LSP Protection Types Comparison

In the Junos OS, the one-to-one backup of traffic protection is provided by fast reroute. Each LSP requires a protecting LSP to be signaled at each hop except the egress router. This method of LSP protection cannot be shared.

In the facillity backup method, the LSP traffic protection is provided on the node and link. Unlike fast reroute, this protecting LSP can be shared by other LSPs.

Table 11 on page 464 summarizes the traffic protection types.

Table 11: One-to-One Backup Compared with Facility Backup

Comparison	One-to-One Backup	Facility Backup
Name of the protecting LSP	Detour LSP	Bypass LSP

Table 11: One-to-One Backup Compared with Facility Backup *(Continued)*

Comparison	One-to-One Backup	Facility Backup
Sharing of the protecting LSP	Cannot be shared	Can be shared by multiple LSPs
Junos configuration statements	`fast-reroute`	`node-link-protection` and `link-protection`

One-to-One Backup Implementation

In the one-to-one backup method, the points of local repair maintain separate backup paths for each LSP passing through a facility. The backup path terminates by merging back with the primary path at a node called the merge point. In this approach, the merge point can be any node downstream from the protected facility.

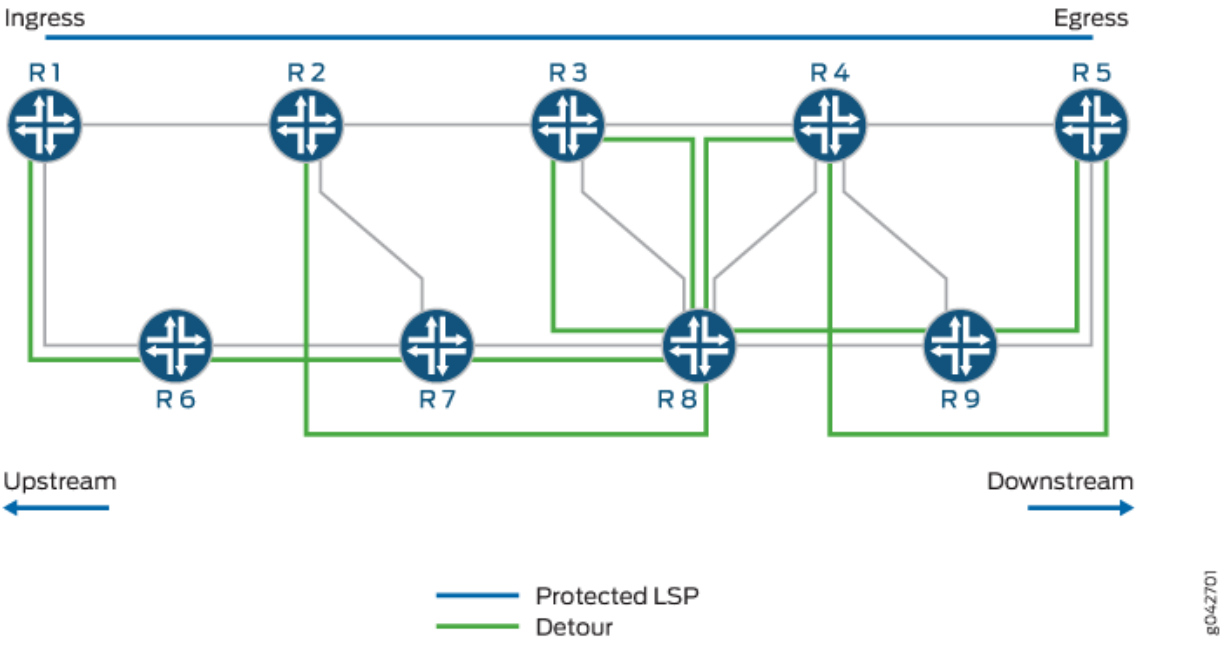
In the one-to-one backup method, an LSP is established that intersects the original LSP downstream of the point of link or node failure. A separate backup LSP is established for each LSP that is backed up.

One-to-one backup is appropriate under the following circumstances:

- Protection of a small number of LSPs relative to the total number of LSPs.
- Path selection criteria, such as bandwidth, priority, and link coloring for detour paths is critical.
- Control of individual LSPs is important.

In Figure 21 on page 466, Routers R1 and R5 are the ingress and egress routers, respectively. A protected LSP is established between the two routers transiting Routers R2, R3, and R4. Router R2 provides user traffic protection by creating a partial backup LSP that merges with the protected LSP at Router R4. This partial one-to-one backup LSP is called a detour. Detours are always calculated to avoid the immediate downstream link and node, providing against both link and node failure.

Figure 21: One-to-One Backup



In the example, the protected LSP is `R1-R2-R3-R4-R5`, and the following detours are established:

- Router R1—`R1-R6-R7-R8-R3`
- Router R2—`R2-R7-R8-R4`
- Router R3—`R3-R8-R9-R5`
- Router R4—`R4-R9-R5`

To protect an LSP that traverses `N` nodes fully, there can be as many as (`N - 1`) detours. The point of local repair sends periodic refresh messages to maintain each backup path, as a result maintaining state information for backup paths protecting individual LSPs is a significant resource burden for the point of local repair. To minimize the number of LSPs in the network, it is desirable to merge a detour back to its protected LSP, when feasible. When a detour LSP intersects its protected LSP at an LSR with the same outgoing interface, it is merged.

Facility Backup Implementation

In the facility backup approach, a point of local repair maintains a single backup path to protect a set of primary LSPs traversing the point of local repair, the facility, and the merge point. The facility backup is based on interface rather than on LSP. While fast reroute protects interfaces or nodes along the entire path of a LSP, the facility backup protection can be applied on interfaces as needed. As a result, fewer

states need to be maintained and refreshed which results in a scalable solution. The facility backup method is also called many-to-one backup.

The facility backup method takes advantage of the MPLS label stack. Instead of creating a separate LSP for every backed-up LSP, a single LSP is created that serves to back up a set of LSPs. Such an LSP tunnel is called a bypass tunnel. In this method, a router immediately upstream from a link failure uses an alternate interface to forward traffic to its downstream neighbor, and the merge point should be the node immediately downstream to the facility. This is accomplished by preestablishing a bypass path that is shared by all protected LSPs traversing the failed link. A single bypass path can safeguard a set of protected LSPs. When an outage occurs, the router immediately upstream from the link outage switches protected traffic to the bypass link, then signals the link failure to the ingress router.

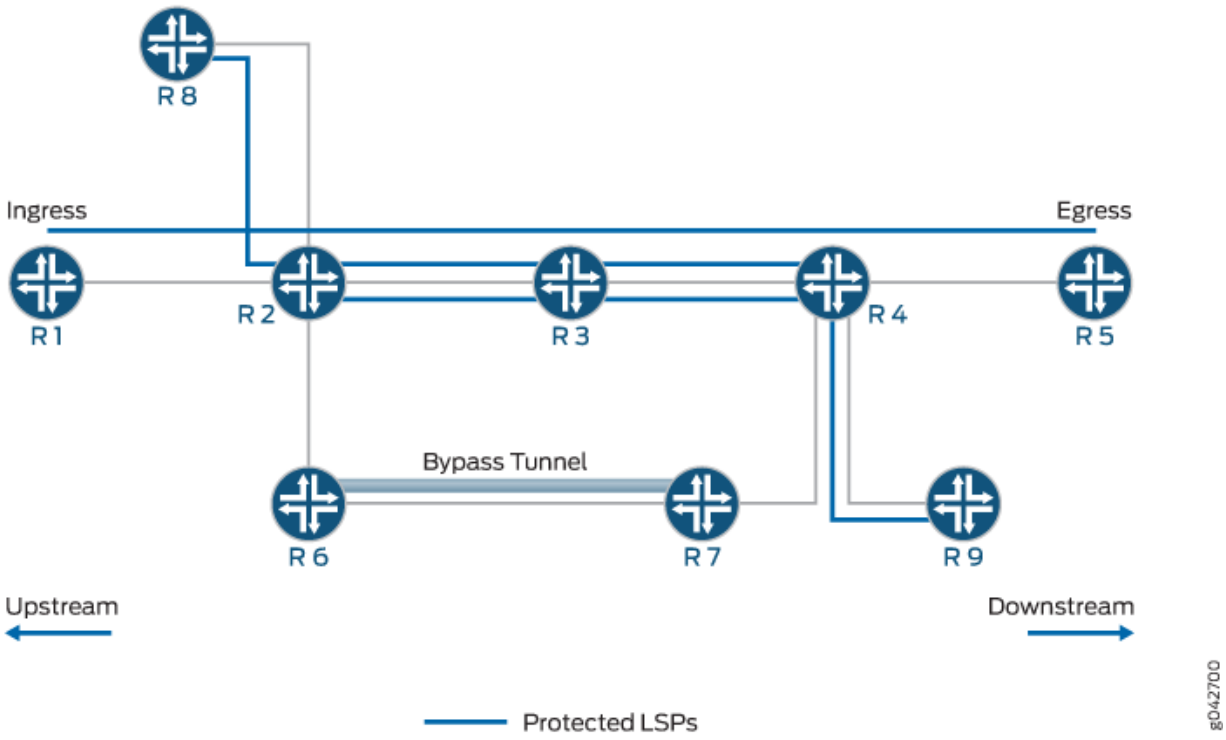
The bypass tunnel must intersect the path of the original LSP(s) somewhere downstream of the point of local repair. This constrains the set of LSPs being backed up through that bypass tunnel to those that pass through some common downstream nodes. All LSPs that pass through the point of local repair and through this common node, and that do not also use the facilities involved in the bypass tunnel are candidates for this set of LSPs.

The facility backup method is appropriate in the following situations:

- The number of LSPs to be protected is large.
- Satisfying path selection criteria (priority, bandwidth, and link coloring) for bypass paths is less critical.
- Control at the granularity of individual LSPs is not required.

In Figure 22 on page 468, Routers R1 and R5 are the ingress and egress routers, respectively. Router R2 has established a bypass tunnel that protects against the failure of Router R2-R3 link and Router R3 node. A bypass tunnel is established between Routers R6 and R7. There are three different protected LSPs that are using the same bypass tunnel for protection.

Figure 22: Facility Backup



The facility backup method provides a scalability improvement, wherein the same bypass tunnel is also used to protect LSPs from any of RoutersR1, R2, or R8 to any of Routers R4, R5, or R9.

Configuring Link Protection on Interfaces Used by LSPs

IN THIS SECTION

- Configuring Bypass LSPs | **470**
- Configuring Administrative Groups for Bypass LSPs | **471**
- Configuring the Bandwidth for Bypass LSPs | **472**
- Configuring Class of Service for Bypass LSPs | **472**
- Configuring the Hop Limit for Bypass LSPs | **473**
- Configuring the Maximum Number of Bypass LSPs | **473**
- Disabling CSPF for Bypass LSPs | **474**
- Disabling Node Protection for Bypass LSPs | **474**
- Configuring the Optimization Interval for Bypass LSPs | **474**

- Configuring Unreserved bandwidth Optimization for Bypass LSPs | **475**
- Configuring an Explicit Path for Bypass LSPs | **476**
- Configuring the Amount of Bandwidth Subscribed for Bypass LSPs | **477**
- Configuring Priority and Preemption for Bypass LSPs | **477**

When you configure node protection or link protection on a router for LSPs as described in "Configuring Node Protection or Link Protection for LSPs" on page 478, you also must configure the `link-protection` statement on the RSVP interfaces used by the LSPs.

To configure link protection on the interfaces used by the LSPs, include the *link-protection* statement:

```
link-protection {
    disable;
    admin-group
        exclude group-names;
        include-all group-names;
        include-any group-names;
    }
    bandwidth bps;
    bypass bypass-name {
        bandwidth bps;
        description text;
        hop-limit number;
        no-cspf;
        path address <strict | loose>;
        priority setup-priority reservation-priority;
        to address;
    }
    class-of-service cos-value;
    hop-limit number;
    max-bypasses number;
    no-cspf;
    no-node-protection;
    optimize-timer seconds;
    path address <strict | loose>;
    priority setup-priority reservation-priority;
    subscription percent {
        ct0 percent;
```

```
        ct1 percent;
        ct2 percent;
        ct3 percent;
    }
}
```

You can include this statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name*]
- [edit logical-systems *logical-system-name* protocols rsvp interface *interface-name*]

All the statements under link-protection are optional.

The following sections describe how to configure link protection:

Configuring Bypass LSPs

You can configure specific bandwidth and path constraints for a bypass LSP. Each manual bypass LSP on a router should have a unique “to” IP address. You can also individually configure each bypass LSP generated when you enable multiple bypass LSPs. If you do not configure the bypass LSPs individually, they all share the same path and bandwidth constraints (if any).

If you specify the bandwidth, hop-limit, and path statements for the bypass LSP, these values take precedence over the values configured at the [edit protocols rsvp interface *interface-name* link-protection] hierarchy level. The other attributes (subscription, no-node-protection, and optimize-timer) are inherited from the general constraints.

To configure a bypass LSP, specify a name for the bypass LSP using the bypass statement. The name can be up to 64 characters in length.

```
bypass bypass-name {
    bandwidth bps;
    description text;
    class-of-service cos-value;
    hop-limit number;
    no-cspf;
    path address <strict | loose>;
    priority setup-priority reservation-priority;
    to address;
}
```

You can include this statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name* link-protection]
- [edit logical-systems *logical-system-name* protocols rsvp interface *interface-name* link-protection]

Configuring the Next-Hop or Next-Next-Hop Node Address for Bypass LSPs

If you configure a bypass LSP, you must also configure the to statement. The to statement specifies the address for the interface of the immediate next-hop node (for link protection) or the next-next-hop node (for node-link protection). The address specified determines whether this is a link protection bypass or a node-link protection bypass. On multiaccess networks (for example, a LAN), this address is also used to specify which next-hop node is being protected.

Configuring Administrative Groups for Bypass LSPs

Administrative groups, also known as link coloring or resource class, are manually assigned attributes that describe the “color” of links, such that links with the same color conceptually belong to the same class. You can use administrative groups to implement a variety of policy-based LSP setups. You can configure administrative groups for bypass LSPs. For more information about configuring administrative groups, see "Configuring Administrative Groups for LSPs" on page 613.

To configure administrative groups for bypass LSPs, include the admin-group statement:

```
admin-group {
    exclude group-names;
    include-all group-names;
    include-any group-names;
}
```

To configure an administrative group for all of the bypass LSPs, include the admin-group statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name* link-protection]
- [edit logical-systems *logical-system-name* protocols rsvp interface *interface-name* link-protection]

To configure an administrative groups for a specific bypass LSP, include the admin-group statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name* link-protection bypass *bypass-name*]
- [edit logical-systems *logical-system-name* protocols rsvp interface *interface-name* link-protection bypass *bypass-name*]

Configuring the Bandwidth for Bypass LSPs

You can specify the amount of bandwidth allocated for automatically generated bypass LSPs or you can individually specify the amount of bandwidth allocated for each LSP.

If you have enabled multiple bypass LSPs, this statement is required.

To specify the bandwidth allocation, include the `bandwidth` statement:

```
bandwidth bps;
```

For automatically generated bypass LSPs, include the `bandwidth` statement at the following hierarchy levels:

- [edit protocols rsvp `interface` *interface-name* link-protection]
- [edit logical-systems *logical-system-name* protocols rsvp `interface` *interface-name* link-protection]

For individually configured bypass LSPs, include the `bandwidth` statement at the following hierarchy levels:

- [edit protocols rsvp `interface` *interface-name* link-protection `bypass` *bypass-name*]
- [edit logical-systems *logical-system-name* protocols rsvp `interface` *interface-name* link-protection `bypass` *bypass-name*]

Configuring Class of Service for Bypass LSPs

You can specify the class-of-service value for bypass LSPs by including the `class-of-service` statement:

```
class-of-service cos-value;
```

To apply a class-of-service value to all the automatically generated bypass LSPs, include the `class-of-service` statement at the following hierarchy levels:

- [edit protocols rsvp `interface` *interface-name* link-protection]
- [edit logical-systems *logical-system-name* protocols rsvp `interface` *interface-name* link-protection]

To configure a class-of-service value for a specific bypass LSPs, include the `class-of-service` statement at the following hierarchy levels:

- [edit protocols rsvp `interface` *interface-name* link-protection `bypass` *bypass-name*]
- [edit logical-systems *logical-system-name* protocols rsvp `interface` *interface-name* link-protection `bypass` *bypass-name*]

Configuring the Hop Limit for Bypass LSPs

You can specify the maximum number of hops a bypass can traverse. By default, each bypass can traverse a maximum of 255 hops (the ingress and egress routers count as one hop each, so the minimum hop limit is two).

To configure the hop limit for bypass LSPs, include the `hop-limit` statement:

```
hop-limit number;
```

For automatically generated bypass LSPs, include the `hop-limit` statement at the following hierarchy levels:

- `[edit protocols rsvp interface interface-name link-protection]`
- `[edit logical-systems logical-system-name protocols rsvp interface interface-name link-protection]`

For individually configured bypass LSPs, include the `hop-limit` statement at the following hierarchy levels:

- `[edit protocols rsvp interface interface-name link-protection bypass bypass-name]`
- `[edit logical-systems logical-system-name protocols rsvp interface interface-name link-protection bypass bypass-name]`

Configuring the Maximum Number of Bypass LSPs

You can specify the maximum number of dynamic bypass LSPs permitted for protecting an interface using the `max-bypasses` statement at the `[edit protocols rsvp interface interface-name link-protection]` hierarchy level. When this statement is configured, multiple bypasses for link protection are enabled. Call admission control (CAC) is also enabled.

By default, this option is disabled and only one bypass is enabled for each interface. You can configure a value of between `0` through `99` for the `max-bypasses` statement. Configuring a value of `0` prevents the creation of any dynamic bypass LSPs for the interface. If you configure a value of `0` for the `max-bypasses` statement, you need to configure one or more static bypass LSPs to enable link protection on the interface.

If you configure the `max-bypasses` statement, you must also configure the `bandwidth` statement (discussed in "Configuring the Bandwidth for Bypass LSPs" on page 472).

To configure the maximum number of bypass LSPs for a protected interface, include the `max-bypasses` statement:

```
max-bypasses number;
```

You can include this statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name* link-protection]
- [edit logical-systems *logical-system-name* protocols rsvp interface *interface-name* link-protection]

Disabling CSPF for Bypass LSPs

Under certain circumstances, you might need to disable CSPF computation for bypass LSPs and use the configured Explicit Route Object (ERO) if available. For example, a bypass LSP might need to traverse multiple OSPF areas or IS-IS levels, preventing the CSPF computation from working. To ensure that link and node protection function properly in this case, you have to disable CSPF computation for the bypass LSP.

You can disable CSPF computation for all bypass LSPs or for specific bypass LSPs.

To disable CSPF computation for bypass LSPs, include the no-cspf statement:

```
no-cspf;
```

For a list of hierarchy levels where you can include this statement, see the statement summary for this statement.

Disabling Node Protection for Bypass LSPs

You can disable node protection on the RSVP interface. Link protection remains active. When this option is configured, the router can only initiate a next-hop bypass, not a next-next-hop bypass.

To disable node protection for bypass LSPs, include the no-node-protection statement:

```
no-node-protection;
```

You can include this statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name* link-protection]
- [edit logical-systems *logical-system-name* protocols rsvp interface *interface-name* link-protection]

Configuring the Optimization Interval for Bypass LSPs

You can configure an optimization interval for bypass LSPs using the optimize-timer statement. At the end of this interval, an optimization process is initiated that attempts to either minimize the number of bypasses currently in use, minimize the total amount of bandwidth reserved for all of the bypasses, or

both. You can configure an optimization interval from 1 through 65,535 seconds. A default value of 0 disables bypass LSP optimization.

When you configure the `optimize-timer` statement, bypass LSPs are reoptimized automatically when you configure or change the configuration of any of the following:

- Administrative group for a bypass LSP—The configuration for an administrative group has been changed on a link along the path used by the bypass LSP. Configure an administrative group using the `admin-group` statement at the `[edit protocols rsvp interface` *`interface-name`* `link-protection]` hierarchy level.
- Fate sharing group—The configuration for a fate sharing group has been changed. Configure a fate sharing group using the `group` statement at the `[edit routing-options fate-sharing]` hierarchy level.
- IS-IS overload—The configuration for IS-IS overload has been changed on a router along the path used by the bypass LSP. Configure IS-IS overload using the `overload` statement at the `[edit protocols isis]` hierarchy level.
- IGP metric—The IGP metric has been changed on a link along the path used by the bypass LSP.

To configure the optimization interval for bypass LSPs, include the `optimize-timer` statement:

```
optimize-timer seconds;
```

You can include this statement at the following hierarchy levels:

- `[edit protocols rsvp interface` *`interface-name`* `link-protection]`
- `[edit logical-systems` *`logical-system-name`* `protocols rsvp interface` *`interface-name`* `link-protection]`

Configuring Unreserved bandwidth Optimization for Bypass LSPs

The default approach of RSVP bypass produces a bypass method that optimizes traffic engineering (TE) metric. The Constrained Shortest Path First (CSPF) can optionally use a different approach to protect a link or a node by leveraging the computation based on unreserved bandwidths on (TE) links.

To enable this feature, use the `optimize bandwidth` configuration statement at the `edit protocols rsvp interface` *`interface`* `link-protection` hierarchy level. Enabling the new configuration statement maximizes the end-to-end unreserved bandwidth.

NOTE: To apply optimize bandwidth configuration statement, enable the **set protocols isis l3-unicast-topology** configuration.

```
link-protection {
                                optimize {
                                   bandwidth;
                                          }
                                              }
```

For configuring bandwidth optimization algorithm for bypass LSPs, include the `optimize bandwidth` statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name* link-protection]
- [edit logical-systems *logical-system-name* protocols rsvp interface *interface-name* link-protection]

Configuring an Explicit Path for Bypass LSPs

By default, when you establish a bypass LSP to an adjacent neighbor, CSPF is used to discover the least-cost path. The `path` statement allows you to configure an explicit path (a sequence of strict or loose routes), giving you control over where and how the bypass LSP is established. To configure an explicit path, include the `path` statement:

```
path address <strict | loose>;
```

For automatically generated bypass LSPs, include the `path` statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name* link-protection]
- [edit logical-systems *logical-system-name* protocols rsvp interface *interface-name* link-protection]

For individually configured bypass LSPs, include the `path` statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name* link-protection bypass *bypass-name*]
- [edit logical-systems *logical-system-name* protocols rsvp interface *interface-name* link-protection bypass *bypass-name*]

Configuring the Amount of Bandwidth Subscribed for Bypass LSPs

You can configure the amount of bandwidth subscribed to bypass LSPs. You can configure the bandwidth subscription for the whole bypass LSP or for each class type that might traverse the bypass LSP. You can configure any value between 1 percent and 65,535 percent. By configuring a value less than 100 percent, you are undersubscribing the bypass LSPs. By configuring a value greater than 100 percent, you are oversubscribing the bypass LSPs.

The ability to oversubscribe the bandwidth for the bypass LSPs makes it possible to more efficiently use network resources. You can configure the bandwidth for the bypass LSPs based on the average network load as opposed to the peak load.

To configure the amount of bandwidth subscribed for bypass LSPs, include the `subscription` statement:

```
subscription percentage {
    ct0 percentage;
    ct1 percentage;
    ct2 percentage;
    ct3 percentage;
}
```

You can include this statement at the following hierarchy levels:

- `[edit protocols rsvp interface interface-name link-protection]`
- `[edit logical-systems logical-system-name protocols rsvp interface interface-name link-protection]`

Configuring Priority and Preemption for Bypass LSPs

When there is insufficient bandwidth to establish a more important LSP, you might want to tear down a less important existing LSP to release the bandwidth. You do this by preempting the existing LSP.

For more detailed information on configuring setup priority and reservation priority for LSPs, see "Configuring Priority and Preemption for LSPs" on page 612.

To configure the bypass LSP's priority and preemption properties, include the `priority` statement:

```
priority setup-priority reservation-priority;
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

Configuring Node Protection or Link Protection for LSPs

When you configure node protection or link protection on a router or switch, bypass LSPs are created to the next-hop or next-next-hop routers (switches) for the LSPs traversing the router (switch). You must configure node protection or link protection for each LSP that you want protected. To extend protection along the entire path used by an LSP, you must configure protection on each router that the LSP traverses.

You can configure node protection or link protection for both static and dynamic LSPs.

To configure node protection on a router for a specified LSP, include the `node-link-protection` statement:

```
node-link-protection;
```

You can include this statement at the following hierarchy levels:

- `[edit protocols mpls label-switched-path` *`lsp-name`*`]`
- `[edit logical-systems` *`logical-system-name`* `protocols mpls label-switched-path` *`lsp-name`*`]`

To configure link protection on a router for a specified LSP, include the *link-protection* statement:

```
link-protection;
```

You can include this statement at the following hierarchy levels:

- `[edit protocols mpls label-switched-path` *`lsp-name`*`]`
- `[edit logical-systems` *`logical-system-name`* `protocols mpls label-switched-path` *`lsp-name`*`]`

NOTE: To complete the configuration of node or link protection, you must also configure link protection on all unidirectional RSVP interfaces that the LSPs traverse, as described in "Configuring Link Protection on Interfaces Used by LSPs" on page 468.

Configuring Inter-AS Node and Link Protection

To interoperate with other vendors' equipment, the Junos OS supports the record route object (RRO) node ID subobject for use in inter-AS link and node protection configurations. The RRO node ID subobject is defined in RFC 4561, *Definition of a Record Route Object (RRO) Node-Id Sub-Object*. This functionality is enabled by default in Junos OS Release 9.4 and later.

If you have Juniper Networks routers running Junos OS Release 9.4 and later releases in the same MPLS-TE network as routers running Junos OS Release 8.4 and earlier releases, you might need to disable the RRO node ID subobject by configuring the `no-node-id-subobject` statement:

```
no-node-id-subobject;
```

You can include this statement at the following hierarchy levels:

- `[edit protocols rsvp]`
- `[edit logical-systems` *`logical-system-name`* `protocols rsvp]`

Configuring Constraint Aware Bypass LSPs

IN THIS SECTION

- Benefits of Constraint Aware RSVP Bypass LSPs | **479**

You can configure RSVP bypass LSPs to be aware of and inherit all the path constraints from the primary LSPs. You can also explicitly configure bypass constraints for individual LSPs.

Benefits of Constraint Aware RSVP Bypass LSPs

- Control the MPLS path and prevent bypass LSPs from traversing through a specific geographical area in a global MPLS RSVP network

You can configure constraint aware RSVP bypass LSPs with link-protection, node-link-protection, and containerized for primary LSPs.

To configure constraint aware bypass LSPs, the `inherit-lsp constraints` and `bypass-constraints` statements are introduced at the `[edit protocols mpls label-swithed-path` *`lsp-name`* `link-protection]` and at the `[edit protocols mpls label-swithed-path` *`lsp-name`* `node-link-protection]` hierarchy levels.

To configure constraint aware bypass LSPs on an ingress LER, you can either inherit the bypass constraints or explicitly define separate constraints for bypass.

To inherit the bypass constraints, include the `inherit-lsp constraints` statement at the `[edit protocols mpls label-swithed-path` *`lsp-name`* `link-protection]` and at the `[edit protocols mpls label-swithed-path` *`lsp-name`* `node-link-protection]` hierarchy levels.

To define separate constraints for bypass LSPs, include the `bypass-constraints` statement at the `[edit protocols mpls label-swithed-path` *`lsp-name`* `link-protection]` and at the `[edit protocols mpls label-swithed-path` *`lsp-name`* `node-link-protection]` hierarchy levels. You can configure the options inside the `bypass-constraints` statement hierarchy to either inherit all the constraints from the primary LSP or specific constraints for that LSP's bypass.

NOTE: Configuring transport-class statement is not mandatory while configuring constraint aware bypass LSPs. The constraint aware bypass LSP feature works with or without transport-class statement configured for the LSP.

On PLRs, you need to include the `constraint-aware-bypass` statement at the `[edit protocols rsvp interface` *`interface-name`* `link-protection]` hierarchy level to support constraint aware bypass LSPs.

If you have multiple LSPs at a particular ingress LER and you decide to enable the constraint aware bypass feature only on a small subset of LSPs, then the `bypass-constraints` statement helps in enabling the feature only on those specific LSPs.

If you do not configure `bypass-constraints` statement at the ingress for a particular LSP, then that LSP continues to have the current non-constraint aware bypass association. The transit LSRs and PLR automatically establishes constraint aware bypass LSPs only for the LSPs for which the `bypass-constraints` statement is configured at the ingress under MPLS and at PLR under RSVP.

When you modify the constraint aware bypass LSPs configuration on the ingress, then such configuration changes are handled in the make before break (MBB) fashion for the LSP. That is, a new instance of the LSP is signaled with the new bypass constraint objects. When the PLR receives the path message for the new instance of the LSP, the PLR establishes the appropriate bypass LSP that satisfies those constraints.

Similarly, when you modify the constraint aware bypass configuration at PLR, all the bypass LSPs are recomputed and re-associated with the primary LSPs at that PLR.

Dynamic link-protection bypass LSPs are created along the path at the PLRs which follows the same CSPF constraints like excluding a specific admin-group.

NOTE: The constraint aware bypass LSP feature does not work when:

- LSPs are configured with no-cspf statement at the ingress.
- LSPs are configured with pop-and-forward functionality.
- Bypass LSPs are manually configured at PLR.

- `max-bypasses` statement is configured at PLR

If there are any changes to the resource affinities signaled for the LSP, then each individual LSR along the LSP's path automatically recomputes the bypasses as required.

When the `bypass-constraints` statement is configured at the ingress indicating the intent to signal constraint aware bypass LSPs, the ingress LER includes the FAST_REROUTE object in the RSVP path message for the LSP. Ingress sets the Flags field to 0x02 indicating Facility Backup as the type of protection. Ingress also populates all the constraints as configured for that bypass LSP.

On receipt of the FAST_REROUTE object in the RSVP path message, every PLR router creates and maintains a lsp-affinities-profile which stores all the received constraints. PLR performs a lookup amongst existing bypass LSPs to see if any of the existing bypass LSPs satisfy the constraint requirements. If existing bypass LSPs satisfies the constraints, then that bypass LSP is associated with the primary LSP. Otherwise, if the constraints are not satisfied by existing bypass LSP, then PLR proceeds to establish a new bypass matching all the constraints received in the incoming RSVP path message.

The outputs of `show rsvp session` and `show rsvp session bypass` commands have been enhanced to display the bypass constraints inherited from the primary LSPs and the resource affinities signaled for the LSP.

The following is a sample output of `show rsvp session extensive` command displaying the bypass constraints information:

```
user@host> show rsvp session extensive
8.8.8.8
  From: 1.1.1.1, LSPstate: Up, ActiveRoute: 0
  LSPname: lsp-R1-R8-1, LSPpath: Primary
  LSPtype: Static Configured
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 299984
  Resv style: 1 SE, Label in: -, Label out: 299984
  Time left:    -, Since: Sun Nov 19 02:46:26 2023
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 3 receiver 18914 protocol 0
  Link protection desired
  Type: Link protected LSP, using Bypass->22.1.9.9
      6 Nov 20 15:16:04 Link protection up, using Bypass->22.1.9.9
      5 Nov 20 15:16:03 Bypass in down state, Bypass->22.1.9.9[3 times, first Nov 20 15:16:01 ]
  Bypass constraints for LSP:
      Hop Limit : 9
      Include Any: green        Include All: Brown         Exclude: Red
```

```
  Enhanced FRR: Enabled (Downstream), LP-MP is 9.9.9.9
  PATH rcvfrom: localclient
  Adspec: sent MTU 1500
  Path MTU: received 1500
  PATH sentto: 22.1.9.9 (ge-0/0/2.0) 4 pkts
       outgoing message state: refreshing, Message ID: 180, Epoch: 11062187
  RESV rcvfrom: 22.1.9.9 (ge-0/0/2.0) 6 pkts, Entropy label: Yes
       incoming message handle: R-82/6, Message ID: 170, Epoch: 11062161
  Explct route: 22.1.9.9 22.9.10.10 22.8.10.8
  Record route: <self> 9.9.9.9 (node-id) 22.1.9.9 10.10.10.10 (node-id) 22.9.10.10 8.8.8.8 (node-
id) 22.8.10.8
```

The following is a sample output of `show rsvp session bypass extensive` command displaying the bypass constraints information:

```
user@host> show rsvp session bypass extensive
  9.9.9.9
  From: 1.1.1.1, LSPstate: Up, ActiveRoute: 0
  LSPname: Bypass->22.1.9.9
  LSPtype: Static Configured
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 300048
  Resv style: 1 SE, Label in: -, Label out: 300048
  Time left:    -, Since: Mon Nov 20 15:16:03 2023
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 40332 protocol 0
  Type: Bypass LSP
    Number of data route tunnel through: 1
    Number of RSVP session tunnel through: 0
    Number of protected LSP instances: 1
  Bypass constraints for LSP:
      Hop Limit : 9
      Include Any: green        Include All: Brown         Exclude: Red
  Enhanced FRR: Enabled (Downstream)
  PATH rcvfrom: localclient
  Adspec: sent MTU 1500
  Path MTU: received 1500
  PATH sentto: 22.1.5.5 (ge-0/0/1.0) 1 pkts
```

RELATED DOCUMENTATION

Basic MPLS Configuration | 46

CHAPTER 7

Measuring MPLS Traffic

IN THIS CHAPTER

- Gather Statistics on MPLS Sessions | 484

Gather Statistics on MPLS Sessions

IN THIS SECTION

- Configuring MPLS to Gather Statistics | 484
- On-Demand Packet Loss and Delay Measurement for UHP LSPs Overview | 486
- Example: Configuring On-Demand Loss and Delay Measurement | 493
- Example: Configuring Pro-active Loss and Delay Measurements for Bidirectional MPLS LSPs | 506
- Configuring On-Demand Loss and Delay Measurement | 516
- Configuring Pro-Active Loss and Delay Measurements | 517

Configuring MPLS to Gather Statistics

You can configure MPLS so that it periodically gathers traffic statistics about all MPLS sessions, including transit sessions, by configuring the `statistics` statement. You must configure the `statistics` statement if you want to collect MPLS traffic statistics using SNMP polling of MPLS Management Information Bases (MIBs).

To enable or disable MPLS statistics collection, include the `statistics` statement:

```
statistics {
    auto-bandwidth (MPLS Statistics);
    file filename <files number> <size size> <world-readable | no-world-readable>;
    interval seconds;
```

```
    no-transit-statistics;
    transit-statistics-polling;
}
```

You can configure these statements at the following hierarchy levels:

- `[edit protocols mpls]`
- `[edit logical-systems` *`logical-system-name`* `protocols mpls]`

The default interval is 300 seconds.

If you configure the `file` option, the statistics are placed in a file, with one entry per LSP. During the specified interval, the following information is recorded in this file:

- The number of packets, number of bytes, packets per second, and bytes per second transmitted by each LSP. Feature parity for the display of packet and byte statistics for sub-LSPs of a point-to-multipoint LSP on the Junos Trio chipset is supported in Junos OS Releases 11.1R2, 11.2R2, and 11.4.
- The percent of bandwidth transmitted over a given LSP in relation to the bandwidth percentage configured for that LSP. If no bandwidth is configured for an LSP, 0 percent is recorded in the percentage column.

At the end of each periodic report, a summary shows the current time, total number of sessions, number of sessions read, number of sessions ignored, and read errors, if any. Ignored sessions are typically those not in the up state or those with a reserved (0 through 15) incoming label (typically the egress point of an LSP). The reason for a read error appears on the same line as the entry for the LSP on which the error occurred. Gathering statistics is an unreliable process; occasional read errors might affect their accuracy. Sample output follows:

```
lsp6                    0 pkt              0 Byte      0 pps        0 Bps    0
lsp5                    0 pkt              0 Byte      0 pps        0 Bps    0
lsp6.1              34845 pkt        2926980 Byte   1049 pps    88179 Bps  132
lsp5.1                  0 pkt              0 Byte      0 pps        0 Bps    0
lsp4                    0 pkt              0 Byte      0 pps        0 Bps    0
Dec  7 17:28:38 Total 6 sessions: 5 success, 0 fail, 1 ignored
```

On-Demand Packet Loss and Delay Measurement for UHP LSPs Overview

IN THIS SECTION

- Importance of Measuring Packet Loss and Delay | 486
- Defining Packet Loss, Delay, and Throughput | 487
- Packet Loss and Delay Measurement Mechanisms | 487
- Packet Loss and Delay Metrics | 488
- Packet Loss and Delay Measurement Concepts | 488
- Packet Loss and Delay Measurement Functionality | 491
- Packet Loss and Delay Features | 492

This topic describes methods for measuring packet loss, delay, and throughput for point-to-point ultimate hop popping (UHP) label-switched paths (LSPs) in MPLS networks to enable monitoring of network performance.

Importance of Measuring Packet Loss and Delay

The rise of bandwidth-consuming applications, such as IPTV and mobile video, coupled with the pressure to minimize the cost per bit and maximize the value per bit, is forcing carriers to transition their transport networks from circuit-based technologies to packet-based technologies. MPLS is a widely successful, connection-oriented packet transport technology that is ideally suited for packet-based transport networks.

With the emergence of new applications on data networks, it is becoming increasingly important for service providers to accurately predict the impact of new application rollouts. Understanding and modelling network performance in the network is especially relevant for deployment of new-world applications to ensure successful implementations. In packet networks, packet loss and delay are two of the most fundamental measures of performance. Their role is even more central when it comes to end-to-end measurements.

The traffic belonging to most of the end-to-end user applications is either loss sensitive (file transfer), delay sensitive (voice or video applications), or both (interactive computing applications). The service-level agreements (SLAs) of service providers depend on the ability to measure and monitor these network performance metrics, as the SLAs are directly or indirectly dependent on the loss and delay the customer traffic experiences in the service provider network.

To ensure compliance to the SLA, service providers need tools to measure and monitor the performance metrics for packet loss, one-way delay and two-way delay, and related metrics, such as delay variation

and channel throughput. This measurement capability provides service providers with greater visibility into the performance characteristics of their networks, thereby facilitating planning, troubleshooting, and network performance evaluation.

Defining Packet Loss, Delay, and Throughput

In packet networks, packet loss and delay are two of the most fundamental measures of performance.

- **Loss**—Packet loss is the failure of one or more transmitted packets to arrive at their destination. Packet loss refers to the packets of data that are dropped by the network to manage congestion.

  Data applications are very tolerant to packet loss, as they are generally not time sensitive and can retransmit the packets that were dropped. However, in video conference environments and pure audio communications, such as VoIP, packet loss can create jitter.

- **Delay**—Packet delay (also called latency) is the amount of time it takes for a packet of data to get from one designated point to another, depending on the speed of the transmission medium, such as copper wire, optical fiber, or radio waves, and the delays in transmission by devices along the way, such as routers and modems.

  A low latency indicates a high network efficiency.

- **Throughput**—Packet delay measures the amount of time between the start of an action and its completion, whereas throughput is the total number of such actions that occur in a given amount of time.

Packet Loss and Delay Measurement Mechanisms

Packet delay and loss are two fundamental measures of network performance. Junos OS provides an on-demand mechanism to measure packet loss and delay over associated bidirectional MPLS ultimate hop popping (UHP) label-switched paths (LSPs).

The on-demand delay and packet loss measurement mechanism is initiated using the following CLI commands:

- `monitor mpls loss rsvp`—Performs an on-demand loss measurement for associated bidirectional UHP LSPs.

- `monitor mpls delay rsvp`—Performs an on-demand delay measurement for associated bidirectional UHP LSPs.

- `monitor mpls loss-delay rsvp`—Performs an on-demand combined loss and delay measurement for associated bidirectional UHP LSPs.

For initiating the delay and packet loss measuring mechanism, the desired parameters for measurement, such as the type of measurement and LSP name, need to be entered. On receiving the parameters, a summary of the performance monitoring data is displayed and the mechanism is terminated.

Packet Loss and Delay Metrics

The following performance metrics are measured using the on-demand packet loss and delay mechanisms:

- Loss measurement (packet and octet)
- Throughput measurement (packet and octet)
- Two-way channel delay
- Round-trip delay
- Inter-packet delay variation (IPDV)

The `monitor mpls loss rsvp` command performs the loss and throughput measurement, and the `monitor mpls delay rsvp` command performs the two-way channel delay, round-trip delay, and IPDV measurements. The `monitor mpls loss-delay rsvp` command performs a combined loss and delay measurement and measures all of the above-mentioned performance metrics simultaneously.

Packet Loss and Delay Measurement Concepts

The following concepts help to better understand the functionality of packet loss and delay:

- **Querier**—A querier is the ingress provider edge (PE) router, which originates the query message for loss or delay measurement.
- **Responder**—A responder is the egress PE router, which receives and responds to the query messages from a querier.
- **Associated bidirectional LSP**—An associated bidirectional LSP consists of two unidirectional LSPs that are tied together (or associated with each other) through configuration on both of the LSP end points.

  The on-demand loss and delay measurement can be carried out only on associated bidirectional UHP LSPs.
- **Generic associated channel (G-Ach)**—The performance monitoring messages for the on-demand loss and delay measurement flow over the MPLS G-Ach. This type of channel supports only in-band responses, and does not provide support for out-of-band or no-response modes.
- **Measurement point (MP)**—MP is the location at which a condition is described for the measurement.

The MP for packet loss on the transmit side is between the switching fabric and the transmit interface. The counter value is stamped in the loss measurement message in the hardware before it is queued for transmission.

The MP for packet loss on the receive side is between the receive interface and the switching fabric. The MP is distributed on the receive side. Furthermore, when the transmit interface is an aggregate interface, the MP is distributed as well.

- **Query rate**—Query rate is the interval between two queries sent for loss and delay measurement.

  Because the loss and delay measurement messages originate from the Routing Engine, a high query rate for multiple channels puts a heavy burden on the Routing Engine. The minimum query interval supported is 1 second.

  The query rate should be high for 32-bit counters, because the counters might wrap quickly when data traffic rate is very high. The query rate can be low when 64-bit counters are in use at all the four measurement point locations involved in loss measurement. Junos OS supports only 64-bit counters.

- **Traffic class**—By default, loss measurement is supported for the whole channel. Junos OS also supports traffic class scoped packet loss measurement, where counters that maintain data traffic statistics per traffic class have to be created.

  Per traffic class counters are not created by default. To configure traffic class scoped loss measurement, include the `traffic-class-statistics` statement at the `[edit protocols mpls statistics]` hierarchy level.

  When `traffic-class-statistics` is configured, control packets flowing over the G-Ach are not counted in the transmit and receive counters.

  > **NOTE**: Enabling and disabling of traffic class statistics results in the resetting of all counters (aggregate counter and per-class counters) for the LSPs.

- **Loss measurement mode**—Junos OS supports the direct-mode of on-demand loss measurement, and does not provide support for the inferred-mode.

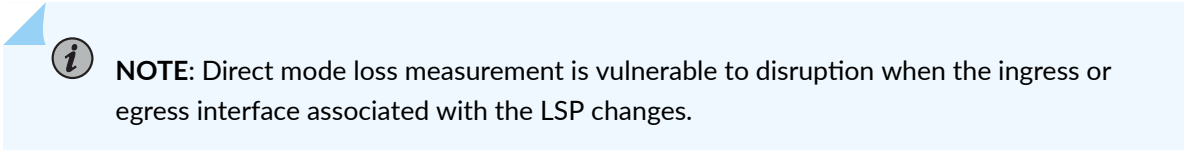
  Direct loss measurement requires data traffic statistics to be maintained at the ingress and egress of two unidirectional LSPs of the associated bidirectional LSP. When an MX Series router is using only MPCs and MICs, counters to maintain data traffic statistics are created by default at the ingress of all types of LSPs and egress of UHP LSPs.

  However, the direct-mode of loss measurement is not fully accurate due to the following reasons:

  - Parallel forwarding nature of the hardware.
  - Presence of equal cost multipath (ECMP) in the network, such as aggregated Ethernet interfaces, which can result in re-ordering of data packets relative to the loss measurement messages.

- Control packets that do not flow over G-Ach are not counted at the LSP ingress, but are counted at the LSP egress.
- Data traffic re-ordering relative to the loss measurement message when a Diffserv is implemented in the MPLS network and loss measurement scope is the complete channel and not traffic class scoped.

  To overcome this limitation, perform traffic class scoped loss measurement when a Diffserv is implemented.



NOTE: Direct mode loss measurement is vulnerable to disruption when the ingress or egress interface associated with the LSP changes.

- **Loss measurement synchronization**—The synchronization conditions specified in section 2.9.8 of RFC 6374 do not hold true in the absolute sense. However, as the loss measurement counters are stamped in hardware, the errors introduced due to not satisfying the synchronization conditions are relatively small. These errors need to be quantified.

  When the transmit or receive interface of the LSP is an aggregate interface, more errors are introduced as compared to when the interfaces are non-aggregate interfaces. In any case, the loss measurement counters are stamped in hardware, and the error needs to be quantified.

- **Delay measurement accuracy**—When the transmit and receive interfaces reside on different Packet Forwarding Engines, the clock must be synchronized on these Packet Forwarding Engines for two-way delay measurements. This condition holds true for the platform on which the on-demand delay measurement feature is implemented.

  When there are aggregate interfaces or ECMP, the delay is measured for only one of the potential paths.

  When a combined loss and delay message is used for delay calculation, the accuracy of delay is lower compared to when the delay measurement message is used in some cases, such as when the transmit or receive interface is an aggregate interface.

  Delay measurement is always performed on a per-traffic-class basis, and the accuracy of the measurement needs to be quantified after testing.

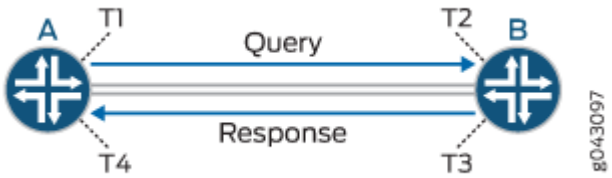
- **Timestamp format**—Junos OS supports only the IEEE 1588 Precision Time Protocol (PTP) [IEEE1588] format for recording delay measurement messages. Network Time Format (NTP) is not supported.

- **Operations, administration, and maintenance (OAM)**—To indicate that all the OAM messages for MPLS LSPs flow over the MPLS G-Ach, and to enable the MPLS performance monitoring messages to be carried over the MPLS G-Ach, the `oam mpls-tp-mode` statement must be included at the `[edit protocols mpls label-switched-path lsp-name]` hierarchy level.

Packet Loss and Delay Measurement Functionality

Figure 23 on page 491 illustrates the basic methods used for the bidirectional measurement of packet loss and delay. A bidirectional channel exists between the two routers, Router A and Router B. The temporal reference points – T1, T2, T3, and T4 – are associated with a measurement operation that takes place at Router A. The operation consists of Router A sending a query message to Router B, and Router B sending back a response. Each reference point indicates the point of time at which either the query or the response message is transmitted or received over the channel.

Figure 23: Basic Bidirectional Measurement



In Figure 23 on page 491, Router A can arrange to measure the packet loss over the channel in the forward and reverse directions by sending loss measurement query messages to Router B. Each of the forward and reverse messages contain the count of packets transmitted prior to time T1 over the channel to Router B (A_TxP).

When the message reaches Router B, two values are appended to the message and the message is reflected back to Router A. The two values are the count of packets received prior to time T2 over the channel from Router A (B_RxP) and the count of packets transmitted prior to time T3 over the channel to Router A (B_TxP).

When the response reaches Router A, a fourth value is appended to the message – the count of packets received prior to time T4 over the channel from Router B (A_RxP).

These four counter values – (A_TxP), (B_RxP), (B_TxP), and (A_RxP) – enable Router A to compute the desired loss statistics. Because the transmit count at Router A and the receive count at Router B (and vice versa) might not be synchronized at the time of the first message, and to limit the effects of counter wrap, the loss is computed in the form of a delta between the messages.

The transmit loss (A_TxLoss[n-1,n]) and receive loss (A_RxLoss[n-1,n]) within the measurement interval marked by the messages LM[n-1] and LM[n] are computed by Router A as follows:

1. A_TxLoss[n-1,n] = (A_TxP[n] - A_TxP[n-1]) - (B_RxP[n] - B_RxP[n-1])
2. A_RxLoss[n-1,n] = (B_TxP[n] - B_TxP[n-1]) - (A_RxP[n] - A_RxP[n-1])

The arithmetic is modulo the counter size.

To measure at Router A the delay over the channel to Router B, a delay measurement query message is sent from Router A to Router B containing a timestamp recording the instant at which it is transmitted. In Figure 23 on page 491, the timestamp is recorded in T1.

When the message reaches Router B, a timestamp is added, recording the instant at which it is received (T2). The message can now be reflected from Router B to Router A, with Router B adding its transmit timestamp (T3) and Router A adding its receive timestamp (T4).

These four timestamps – T1, T2, T3, and T4 – enable Router A to compute the one-way delay in each direction, as well as the two-way delay for the channel. The one-way delay computations require that the clocks of Routers A and B be synchronized.

At this point, Router A can compute the two-way channel delay and round-trip delay associated with the channel as follows:

1. Two-way channel delay = (T4 - T1) - (T3 - T2)
2. Round-trip delay = T4 - T1

Packet Loss and Delay Features

Supported Features of Packet Loss and Delay

Junos OS supports the following features with on-demand loss and delay measurement:

- Performance monitoring for associated bidirectional MPLS point-to-point UHP LSPs only
- Loss measurement
- Throughput measurement
- Two-way delay measurement (channel delay and round-trip delay)
- Inter-packet delay variation (IPDV)
- Direct-mode loss measurement
- Aggregated Ethernet and aggregated SONET interfaces
- Multichassis support
- 64-bit compatible

Unsupported Features of Packet Loss and Delay

Junos OS does not support the following on-demand loss and delay measurement functionality:

- Loss and delay measurement for pseudowires (section 2.9.1 of RFC 6374)
- Unidirectional measurement (section 2.6 of RFC 6374)

- Dyadic measurement (section 2.7 of RFC 6374)
- Loss and delay measurement in loopback mode (section 2.8 of RFC 6374)
- Loss and delay measurement to an intermediate node from an LSP endpoint (section 2.9.5 of RFC 6374)
- External post-processing (section 2.9.7 of RFC 6374)
- Inferred-mode loss measurement (section 2.9.8 of RFC 6374)
- Pro-active mode
- Logical systems
- SNMP

Example: Configuring On-Demand Loss and Delay Measurement

IN THIS SECTION

- Requirements | **493**
- Overview | **494**
- Configuration | **495**
- Verification | **500**

This example shows how to enable on-demand loss and delay measurement for point-to-point ultimate hop popping (UHP) label-switched paths (LSPs) in MPLS networks to monitor network performance.

Requirements

This example uses the following hardware and software components:

- Two MX Series 5G Universal Routing Platforms that contain MPC/MICs only
- Junos OS Release 14.2 or later running on all the routers

Before you begin:

1. Configure the device interfaces.
2. Configure the autonomous system numbers and router IDs for the devices.

3. Configure the following protocols:

   - RSVP
   - MPLS
   - OSPF

Overview

IN THIS SECTION

- Topology | **494**

Starting with Junos OS Release 14.2, an on-demand tool to monitor and measure packet loss, packet delay, or both for associated bidirectional MPLS ultimate hop popping (UHP) point-to-point label-switched paths (LSPs) is introduced. The tool can be enabled using the following CLI commands – `monitor mpls loss rsvp`, `monitor mpls delay rsvp`, and `monitor mpls loss-delay rsvp`.

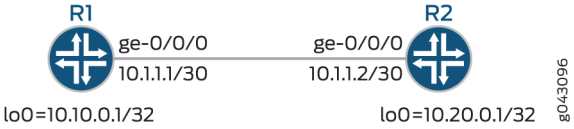
These commands provide an on-demand summary of performance metrics for direct mode packet loss, two-way packet delay, and related metrics, such as inter-packet delay variation and channel throughput measurement.

This functionality provides real-time visibility into network performance, thereby facilitating network performance planning, troubleshooting, and evaluation.

Topology

Figure 24 on page 494 illustrates the on-demand loss and delay measurement using a simple two-router topology.

Figure 24: Configuring On-Demand Loss and Delay Measurement



In this example, an associated bidirectional LSP is configured between Routers R1 and R2, for which the performance metrics is measured.

Configuration

IN THIS SECTION

- CLI Quick Configuration | 495
- Procedure | 496
- Results | 498

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

R1

```
set chassis fpc 0 pic 3 tunnel-services bandwidth 1g
set chassis network-services enhanced-ip
set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/30
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.10.0.1/32
set interfaces lo0 unit 0 family mpls
set routing-options router-id 10.10.0.1
set protocols rsvp interface ge-0/0/0.0
set protocols rsvp interface lo0.0
set protocols rsvp interface fxp0.0 disable
set protocols mpls statistics traffic-class-statistics
set protocols mpls label-switched-path R1-R2 to 10.20.0.1
set protocols mpls label-switched-path R1-R2 oam mpls-tp-mode
set protocols mpls label-switched-path R1-R2 ultimate-hop-popping
set protocols mpls label-switched-path R1-R2 associate-lsp R2-R1
set protocols mpls interface ge-0/0/0.0
set protocols mpls interface lo0.0
set protocols mpls interface fxp0.0 disable
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
```

R2

```
set chassis fpc 0 pic 3 tunnel-services bandwidth 1g
set chassis network-services enhanced-ip
set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.2/30
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.20.0.1/32
set interfaces lo0 unit 0 family mpls
set routing-options router-id 10.20.0.1
set protocols rsvp interface ge-0/0/0.0
set protocols rsvp interface lo0.0
set protocols rsvp interface fxp0.0 disable
set protocols mpls statistics traffic-class-statistics
set protocols mpls label-switched-path R2-R1 to 10.10.0.1
set protocols mpls label-switched-path R2-R1 oam mpls-tp-mode
set protocols mpls label-switched-path R2-R1 ultimate-hop-popping
set protocols mpls label-switched-path R2-R1 associate-lsp R1-R2
set protocols mpls interface ge-0/0/0.0
set protocols mpls interface lo0.0
set protocols mpls interface fxp0.0 disable
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0.interface fxp0.0 disable
```

Procedure

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure Router R1:

1. Enable the chassis with tunnel services and enhanced IP network services configuration.

```
[edit chassis]
user@R1# set fpc 0 pic 3 tunnel-services bandwidth 1g
user@R1# set network-services enhanced-ip
```

2. Configure the interfaces for Router R1.

```
[edit interfaces]
user@R1# set ge-0/0/0 unit 0 family inet address 10.1.1.1/30
user@R1# set ge-0/0/0 unit 0 family mpls
user@R1# set lo0 unit 0 family inet address 10.10.0.1/32
user@R1# set lo0 unit 0 family mpls
```

3. Configure the router ID for Router R1.

```
[edit routing-options]
user@R1# set router-id 10.10.0.1
```

4. Enable RSVP on all the interfaces of Router R1, excluding the management interface.

```
[edit protocols]
user@R1# set rsvp interface ge-0/0/0.0
user@R1# set rsvp interface lo0.0
user@R1# set rsvp interface fxp0.0 disable
```

5. Enable MPLS on all the interfaces of Router R1, excluding the management interface.

```
[edit protocols]
user@R1# set mpls interface ge-0/0/0.0
user@R1# set mpls interface lo0.0
user@R1# set mpls interface fxp0.0 disable
```

6. Configure an associated bidirectional LSP to Router R2.

```
[edit protocols]
user@R1# set mpls label-switched-path R1-R2 to 10.20.0.1
user@R1# set mpls label-switched-path R1-R2 oam mpls-tp-mode
user@R1# set mpls label-switched-path R1-R2 ultimate-hop-popping
user@R1# set mpls label-switched-path R1-R2 associate-lsp R2-R1
```

7. Create traffic classes for maintaining data traffic statistics per traffic class.

This enables traffic class scoped loss measurement.

```
[edit protocols]
user@R1# set mpls statistics traffic-class-statistics
```

8. Configure OSPF with traffic engineering capabilities, and enable OSPF on all the interfaces of Router R1, excluding the management interface.

```
[edit protocols]
user@R1# set ospf traffic-engineering
user@R1# set ospf area 0.0.0.0 interface ge-0/0/0.0
user@R1# set ospf area 0.0.0.0 interface lo0.0
user@R1# set ospf interface fxp0.0 disable
```

Results

From configuration mode, confirm your configuration by entering the `show chassis`, `show interfaces`, `show routing-options`, and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R1# show chassis
fpc 0 {
    pic 3 {
        tunnel-services {
            bandwidth 1g;
        }
    }
}
network-services enhanced-ip;
```

```
user@R1# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 1o.1.1.1/30;
        }
        family mpls;
    }
```

```
}
lo0 {
    unit 0 {
        family inet {
            address 10.10.0.1/32;
        }
        family mpls;
    }
}
```

```
user@R1# show routing-options
router-id 10.10.0.1;
```

```
user@R1# show protocols
rsvp {
    interface ge-0/0/0.0;
    interface lo0.0;
    interface fxp0.0 {
        disable;
    }
}
mpls {
    statistics {
        traffic-class-statistics;
    }
    label-switched-path R1-R2 {
        to 10.20.0.1;
        oam mpls-tp-mode;
        ultimate-hop-popping;
        associate-lsp R2-R1;
    }
    interface ge-0/0/0.0;
    interface lo0.0;
    interface fxp0.0 {
        disable;
    }
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
```

```
        interface ge-0/0/0.0;
        interface lo0.0;
        interface fxp0.0 {
            disable;
        }
    }
}
```

Verification

IN THIS SECTION

- Verifying the LSP Status | **500**
- Verifying Packet Loss Measurement | **501**
- Verifying Packet Delay Measurement | **503**
- Verifying Packet Loss-Delay Measurement | **504**

Confirm that the configuration is working properly.

Verifying the LSP Status

Purpose

Verify that the associated bidirectional LSP between Routers R1 and R2 is up.

Action

From operational mode, run the `show mpls lsp` command.

```
user@R1> show mpls lsp
Ingress LSP: 1 sessions
To              From            State Rt P     ActivePath       LSPname
10.20.0.1        10.10.0.1        Up     0 *                     R1-R2 Assoc-Bidir
Total 1 displayed, Up 1, Down 0

Egress LSP: 1 sessions
To              From            State   Rt Style Labelin Labelout LSPname
```

```
10.10.0.1        10.20.0.1        Up       0  1 FF  299776        - R2-R1 Assoc-Bidir
Total 1 displayed, Up 1, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Meaning

The associated bidirectional LSP R1-R2 is up and active.

Verifying Packet Loss Measurement

Purpose

Verify the on-demand loss measurement result.

Action

From operational mode, run the `monitor mpls loss rsvp R1-R2 count 2 detail` command.

```
user@R1> monitor mpls loss rsvp R1-R2 count 2 detail
(0)
Response code                          : Success
Origin timestamp                       : 1404129082 secs, 905571890 nsecs
Forward transmit count                 : 83040
Forward receive count                  : 83040
Reverse transmit count                 : 83100
Reverse receive count                  : 83100
(1)
Response code                          : Success
Origin timestamp                       : 1404129083 secs, 905048410 nsecs
Forward transmit count                 : 83841
Forward receive count                  : 83841
Reverse transmit count                 : 83904
Reverse receive count                  : 83904
Current forward transmit count         : 801
Current forward receive count          : 801
Current forward loss                   : 0 packets
Current forward loss ratio             : 0.000000
Current forward throughput             : 0.801 kpps
Current reverse transmit count         : 804
```

```
Current reverse receive count           : 804
Current reverse loss                    : 0 packets
Current reverse loss ratio              : 0.000000
Current reverse throughput              : 0.804 kpps
(2)
Response code                           : Success
Origin timestamp                        : 1404129084 secs, 904828715 nsecs
Forward transmit count                  : 84423
Forward receive count                   : 84423
Reverse transmit count                  : 84487
Reverse receive count                   : 84487
Current forward transmit count          : 582
Current forward receive count           : 582
Current forward loss                    : 0 packets
Current forward loss ratio              : 0.000000
Current forward throughput              : 0.582 kpps
Current reverse transmit count          : 583
Current reverse receive count           : 583
Current reverse loss                    : 0 packets
Current reverse loss ratio              : 0.000000
Current reverse throughput              : 0.583 kpps

Cumulative forward transmit count       : 1383
Cumulative forward loss                 : 0 packets
Average forward loss ratio              : 0.000000
Average forward throughput              : 0.692 kpps
Cumulative reverse transmit count       : 1387
Cumulative reverse loss                 : 0 packets
Average reverse loss ratio              : 0.000000
Average reverse throughput              : 0.694 kpps

LM queries sent                         : 3
LM responses received                   : 3
LM queries timedout                     : 0
LM responses dropped due to errors      : 0
```

Meaning

The packet loss measurement for two counts is displayed.

Verifying Packet Delay Measurement

Purpose

Verify the on-demand delay measurement result.

Action

From operational mode, run the `monitor mpls delay rsvp R1-R2 count 2 detail` command.

```
user@R1> monitor mpls delay rsvp R1-R2 count 2 detail
(1)
Response code                          : Success
Querier transmit timestamp             : 1404129122 secs, 479955401 nsecs
Responder receive timestamp            : 1404129122 secs, 468519022 nsecs
Responder transmit timestamp           : 1404129122 secs, 470255123 nsecs
Querier receive timestamp              : 1404129122 secs, 481736403 nsecs
Current two-way channel delay          : 44 usecs
Current round-trip-time                : 1781 usecs
(2)
Response code                          : Success
Querier transmit timestamp             : 1404129123 secs, 480926210 nsecs
Responder receive timestamp            : 1404129123 secs, 469488696 nsecs
Responder transmit timestamp           : 1404129123 secs, 471130706 nsecs
Querier receive timestamp              : 1404129123 secs, 482613911 nsecs
Current two-way channel delay          : 45 usecs
Current round-trip-time                : 1687 usecs

Best two-way channel delay             : 44 usecs
Worst two-way channel delay            : 45 usecs
Average two-way channel delay          : 45 usecs
Best round-trip-time                   : 1687 usecs
Worst round-trip-time                  : 1781 usecs
Average round-trip-time                : 1734 usecs
Average forward delay variation        : 1 usecs
Average reverse delay variation        : 1 usecs

DM queries sent                        : 2
DM responses received                  : 2
DM queries timedout                    : 0
DM responses dropped due to errors     : 0
```

Meaning

The packet delay measurement for two counts is displayed.

Verifying Packet Loss-Delay Measurement

Purpose

Verify the on-demand loss and delay measurement result.

Action

From operational mode, run the `monitor mpls loss-delay rsvp R1-R2 count 2 detail` command.

```
user@R1> monitor mpls loss-delay rsvp R1-R2 count 2 detail
(0)
Response code                          : Success
Forward transmit count                 : 142049
Forward receive count                  : 142049
Reverse transmit count                 : 142167
Reverse receive count                  : 142167
Querier transmit timestamp             : 1404129161 secs, 554422723 nsecs
Responder receive timestamp            : 1404129161 secs, 542877570 nsecs
Responder transmit timestamp           : 1404129161 secs, 546004545 nsecs
Querier receive timestamp              : 1404129161 secs, 557599327 nsecs
(1)
Response code                          : Success
Forward transmit count                 : 143049
Forward receive count                  : 143049
Reverse transmit count                 : 143168
Reverse receive count                  : 143168
Current forward transmit count         : 1000
Current forward receive count          : 1000
Current forward loss                   : 0 packets
Current forward loss ratio             : 0.000000
Current forward throughput             : 1.000 kpps
Current reverse transmit count         : 1001
Current reverse receive count          : 1001
Current reverse loss                   : 0 packets
Current reverse loss ratio             : 0.000000
Current reverse throughput             : 1.001 kpps
Querier transmit timestamp             : 1404129162 secs, 554465742 nsecs
```

```
Responder receive timestamp             : 1404129162 secs, 542919166 nsecs
Responder transmit timestamp            : 1404129162 secs, 545812736 nsecs
Querier receive timestamp               : 1404129162 secs, 557409175 nsecs
Current two-way channel delay           : 49 usecs
Current round-trip-time                 : 2943 usecs
(2)
Response code                           : Success
Forward transmit count                  : 143677
Forward receive count                   : 143677
Reverse transmit count                  : 143799
Reverse receive count                   : 143799
Current forward transmit count          : 628
Current forward receive count           : 628
Current forward loss                    : 0 packets
Current forward loss ratio              : 0.000000
Current forward throughput              : 0.627 kpps
Current reverse transmit count          : 631
Current reverse receive count           : 631
Current reverse loss                    : 0 packets
Current reverse loss ratio              : 0.000000
Current reverse throughput              : 0.630 kpps
Querier transmit timestamp              : 1404129163 secs, 556698575 nsecs
Responder receive timestamp             : 1404129163 secs, 545150128 nsecs
Responder transmit timestamp            : 1404129163 secs, 546918408 nsecs
Querier receive timestamp               : 1404129163 secs, 558515047 nsecs
Current two-way channel delay           : 48 usecs
Current round-trip-time                 : 1816 usecs

Cumulative forward transmit count       : 1628
Cumulative forward loss                 : 0 packets
Average forward loss ratio              : 0.000000
Average forward throughput              : 0.813 kpps
Cumulative reverse transmit count       : 1632
Cumulative reverse loss                 : 0 packets
Average reverse loss ratio              : 0.000000
Average reverse throughput              : 0.815 kpps

Best two-way channel delay              : 48 usecs
Worst two-way channel delay             : 49 usecs
Average two-way channel delay           : 49 usecs
Best round-trip-time                    : 1816 usecs
Worst round-trip-time                   : 3176 usecs
Average round-trip-time                 : 2645 usecs
```

```
Average forward delay variation          : 1 usecs
Average reverse delay variation          : 0 usecs

LDM queries sent                         : 3
LDM responses received                   : 3
LDM queries timedout                     : 0
LDM responses dropped due to errors      : 0
```

Meaning

The packet loss and delay measurement for two counts is displayed.

Example: Configuring Pro-active Loss and Delay Measurements for Bidirectional MPLS LSPs

IN THIS SECTION

- Requirements | **506**
- Overview | **507**
- Configuration | **508**
- Verification | **514**

This example shows how to configure pro-active loss and delay measurements for point-to-point ultimate-hop popping label-switched paths (LSPs) in MPLS networks to monitor network performance.

Requirements

This example uses the following hardware and software components:

- Two MX Series 5G Universal Routing Platforms that contain MPC/MICs only
- Junos OS Release 15.1 or later running on all the routers

Before you begin:

1. Configure the device interfaces.
2. Configure the autonomous system numbers and router IDs for the devices.
3. Configure the following protocols:

a. MPLS

b. OSPF

c. RSVP

Overview

IN THIS SECTION

Topology | 507

Starting with Junos OS Release 15.1, a pro-active tool to monitor and measure packet loss, packet delay, or both for associated bidirectional MPLS ultimate-hop popping point-to-point label-switched paths (LSPs) is introduced.

This feature provides the following performance metrics:

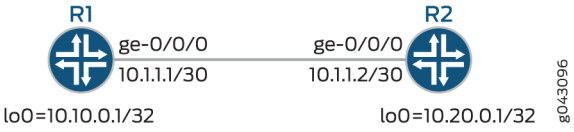
- Inter-packet delay variation (IPDV)
- Loss measurement
- Round-trip delay (RTT)
- Throughput measurement
- Two-way channel delay

This functionality provides real-time visibility into network performance, thereby facilitating network performance planning, troubleshooting, and evaluation.

Topology

Figure 25 on page 507 illustrates the pro-active loss and delay measurements using a simple two-router topology.

Figure 25: Configuring Pro-Active Loss and Delay Measurements



In this example, an associated bidirectional LSP is configured between Routers R1 and R2, for which the performance metrics are measured.

Configuration

IN THIS SECTION

- CLI Quick Configuration | 508
- Procedure | 510
- Results | 512

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

R1

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/30
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.10.0.1/32
set interfaces lo0 unit 0 family mpls
set protocols mpls interface ge-0/0/0.0
set protocols mpls interface lo0.0
set protocols mpls interface fxp0.0 disable
set protocols mpls label-switched-path R1-R2 associate-lsp R2-R1
set protocols mpls label-switched-path R1-R2 install 10.20.0.0/24 active
set protocols mpls label-switched-path R1-R2 oam mpls-tp-mode
set protocols mpls label-switched-path R1-R2 oam performance-monitoring querier delay traffic-
class tc-0 query-interval 1000
set protocols mpls label-switched-path R1-R2 oam performance-monitoring querier loss traffic-
class none query-interval 1000
set protocols mpls label-switched-path R1-R2 oam performance-monitoring querier loss-delay
traffic-class tc-0 query-interval 1000
set protocols mpls label-switched-path R1-R2 oam performance-monitoring responder delay min-
query-interval 1000
```

```
set protocols mpls label-switched-path R1-R2 oam performance-monitoring responder loss min-query-interval 1000
set protocols mpls label-switched-path R1-R2 to 10.20.0.1
set protocols mpls label-switched-path R1-R2 ultimate-hop-popping
set protocols mpls statistics traffic-class-statistics
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ospf traffic-engineering
set protocols rsvp interface ge-0/0/0.0
set protocols rsvp interface lo0.0
set protocols rsvp interface fxp0.0 disable
set routing-options router-id 10.10.0.1
```

R2

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.2/30
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.20.0.1/32
set interfaces lo0 unit 0 family mpls
set protocols mpls interface ge-0/0/0.0
set protocols mpls interface lo0.0
set protocols mpls interface fxp0.0 disable
set protocols mpls label-switched-path R2-R1 associate-lsp R1-R2
set protocols mpls label-switched-path R2-R1 install 10.10.0.0/24 active
set protocols mpls label-switched-path R2-R1 oam mpls-tp-mode
set protocols mpls label-switched-path R2-R1 oam performance-monitoring responder delay min-query-interval 1000
set protocols mpls label-switched-path R2-R1 oam performance-monitoring responder loss min-query-interval 1000
set protocols mpls label-switched-path R2-R1 oam performance-monitoring querier delay traffic-class tc-0 query-interval 1000
set protocols mpls label-switched-path R2-R1 oam performance-monitoring querier loss traffic-class none query-interval 1000
set protocols mpls label-switched-path R2-R1 oam performance-monitoring querier loss-delay traffic-class tc-0 query-interval 1000
set protocols mpls label-switched-path R2-R1 to 10.10.0.1
set protocols mpls label-switched-path R2-R1 ultimate-hop-popping
set protocols mpls statistics traffic-class-statistics
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

```
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ospf traffic-engineering
set protocols rsvp interface ge-0/0/0.0
set protocols rsvp interface lo0.0
set protocols rsvp interface fxp0.0 disable
set routing-options router-id 10.20.0.1
```

Procedure

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure Router R1:

1. Enable the enhanced IP network services configuration.

```
[edit chassis]
user@R1# set network-services enhanced-ip
```

2. Configure the interfaces for Router R1.

```
[edit interfaces]
user@R1# set ge-0/0/0 unit 0 family inet address 10.1.1.1/30
user@R1# set ge-0/0/0 unit 0 family mpls
user@R1# set lo0 unit 0 family inet address 10.10.0.1/32
user@R1# set lo0 unit 0 family mpls
```

3. Configure the router ID for Router R1.

```
[edit routing-options]
user@R1# set router-id 10.10.0.1
```

4. Enable RSVP on all the interfaces of Router R1, excluding the management interface.

```
[edit protocols]
user@R1# set rsvp interface ge-0/0/0.0
```

```
user@R1# set rsvp interface lo0.0
user@R1# set rsvp interface fxp0.0 disable
```

5. Enable MPLS on all the interfaces of Router R1, excluding the management interface.

```
[edit protocols]
user@R1# set mpls interface ge-0/0/0.0
user@R1# set mpls interface lo0.0
user@R1# set mpls interface fxp0.0 disable
```

6. Configure an associated bidirectional LSP to Router R2.

```
[edit protocols]
user@R1# set mpls label-switched-path R1-R2 to 10.20.0.1
user@R1# set mpls label-switched-path R1-R2 install 10.20.0.0/24 active
user@R1# set mpls label-switched-path R1-R2 oam mpls-tp-mode
user@R1# set mpls label-switched-path R1-R2 ultimate-hop-popping
user@R1# set mpls label-switched-path R1-R2 associate-lsp R2-R1
```

7. Create traffic classes for maintaining data traffic statistics per traffic class.

   This enables traffic class scoped loss and delay measurement.

```
[edit protocols]
user@R1# set mpls statistics traffic-class-statistics
```

8. Configure performance monitoring at the querier side.

```
[edit protocols]
user@R1# set mpls label-switched-path R1-R2 oam performance-monitoring querier delay
traffic-class tc-0 query-interval 1000
user@R1# set mpls label-switched-path R1-R2 oam performance-monitoring querier loss traffic-
class none query-interval 1000
user@R1# set mpls label-switched-path R1-R2 oam performance-monitoring querier loss-delay
traffic-class tc-0 query-interval 1000
```

9. Configure performance monitoring at the responder side.

```
[edit protocols]
user@R1# set mpls label-switched-path R1-R2 oam performance-monitoring responder delay min-query-interval 1000
user@R1# set mpls label-switched-path R1-R2 oam performance-monitoring responder loss min-query-interval 1000
```

10. Configure OSPF with traffic engineering capabilities, and enable OSPF on all the interfaces of Router R1, excluding the management interface.

```
[edit protocols]
user@R1# set ospf traffic-engineering
user@R1# set ospf area 0.0.0.0 interface ge-0/0/0.0
user@R1# set ospf area 0.0.0.0 interface lo0.0
user@R1# set ospf interface fxp0.0 disable
```

Results

From configuration mode, confirm your configuration by entering the `show chassis`, `show interfaces`, `show routing-options`, and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R1# show chassis
network-services enhanced-ip;
```

```
user@R1# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 10.1.1.1/30;
        }
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
```

```
            address 10.10.0.1/32;
        }
        family mpls;
    }
}
```

```
user@R1# show routing-options
router-id 10.10.0.1;
```

```
user@R1# show protocols
rsvp {
    interface ge-0/0/0.0;
    interface lo0.0;
    interface fxp0.0 {
        disable;
    }
}
mpls {
    label-switched-path R1-R2 {
        to 10.20.0.1;
        install 10.20.0.0/24  active;
        oam {
            mpls-tp-mode;
            performance-monitoring {
                querier {
                    loss {
                        traffic-class none {
                            query-interval 1000;
                        }
                    }
                    delay {
                        traffic-class tc-0 {
                            query-interval 1000;
                        }
                    }
                    loss-delay {
                        traffic-class none {
                            query-interval 1000;
                        }
                    }
```

```
                }
                responder {
                    loss {
                        min-query-interval 1000;
                    }
                    delay {
                        min-query-interval 1000;
                    }
                }
            }
        }
        ultimate-hop-popping;
        associate-lsp R2-R1;
    }
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface ge-0/0/0.0;
        interface lo0.0;
        interface fxp0.0 {
            disable;
        }
    }
}
```

Verification

IN THIS SECTION

- Verifying Loss and Delay Measurement | 514

Verifying Loss and Delay Measurement

Purpose

Verify the loss and delay measurement.

Action

From operational mode, run the `show performance-monitoring mpls lsp` command.

```
user@R1> show performance-monitoring mpls lsp
Session Total: 3 Up: 3 Down: 0
  LSP name:R1-R2, PM State:Up
    Loss measurement Data:
      Duration: 00:04:43
      Traffic-class: None
      Queries sent: 282
      Responses received: 282
      Responses dropped due to errors: 0
      Queries timeout: 0
      Forward loss measurement:
        Average packet loss: 0
        Average packet throughput: 554338
      Reverse loss measurement:
        Average packet loss: 0
        Average packet throughput: 1352077
  LSP name:R1-R2, PM State:Up
    Delay measurement Data:
      Duration: 00:04:43
      Traffic-class: 0
      Queries sent: 282
      Responses received: 282
      Responses dropped due to errors: 0
      Queries timeout: 0
      Best 2-way channel delay: 72 usecs
      Worst 2-way channel delay: 365 usecs
      Best round trip time: 843 usecs
      Worst round trip time: 105523 usecs
      Avg absolute fw delay variation: 1619 usecs
      Avg absolute rv delay variation: 1619 usecs
  LSP name:R1-R2, PM State:Up
    Loss measurement Data:
      Duration: 00:04:43
      Traffic-class: None
      Queries sent: 282
      Responses received: 282
      Responses dropped due to errors: 0
      Queries timeout: 0
```

```
    Forward loss measurement:
      Average packet loss: 0
      Average packet throughput: 553927
    Reverse loss measurement:
      Average packet loss: 0
      Average packet throughput: 1351531
  Delay measurement Data:
    Best 2-way channel delay: 76 usecs
    Worst 2-way channel delay: 368 usecs
    Best round trip time: 1082 usecs
    Worst round trip time: 126146 usecs
    Avg absolute fw delay variation: 1618 usecs
    Avg absolute rv delay variation: 1619 usecs
```

Meaning

The packet loss and delay measurement metrics for LSP are displayed.

Configuring On-Demand Loss and Delay Measurement

You can configure an on-demand loss and delay measurement for point-to-point ultimate hop popping (UHP) label-switched paths (LSPs) in MPLS networks to monitor network performance. The `monitor mpls loss rsvp`, `monitor mpls delay rsvp`, and `monitor mpls loss-delay rsvp` CLI commands provide an on-demand summary of performance metrics for direct mode packet loss, two-way packet delay, and related metrics, such as inter-packet delay variation and channel throughput measurement.

This functionality provides real-time visibility into network performance, thereby facilitating network performance planning, troubleshooting, and evaluation.

Before you begin:

1. Configure the device interfaces.
2. Configure the device router ID.
3. Configure the following protocols:
   - RSVP
   - OSPF

     Enable traffic engineering capabilities.
   - MPLS

To configure the PE device:

1. Enable the chassis with tunnel services and enhanced IP network services configuration.

```
[edit chassis]
user@R1# set fpc fpc-slot pic pic-slot tunnel-services bandwidth bandwidth
user@R1# set network-services enhanced-ip
```

2. Configure an associated bidirectional LSP to the remote router.

```
[edit protocols]
user@R1# set mpls label-switched-path lsp-name to remote-router-ip-address
user@R1# set mpls label-switched-path lsp-name oam mpls-tp-mode
user@R1# set mpls label-switched-path lsp-name ultimate-hop-popping
user@R1# set mpls label-switched-path lsp-name associate-lsp lsp-name
```

3. Create traffic classes for maintaining data traffic statistics per traffic class.

   This enables traffic class scoped loss measurement.

```
[edit protocols]
user@R1# set mpls statistics traffic-class-statistics
```

Configuring Pro-Active Loss and Delay Measurements

You can configure pro-active loss and delay measurements for point-to-point ultimate-hop popping label-switched paths (LSPs) in MPLS networks to monitor network performance. The `show performance-monitoring mpls lsp` CLI command provides a summary of performance metrics for direct mode packet loss, two-way packet delay, and related metrics, such as inter-packet delay variation and channel throughput measurement.

This functionality provides real-time visibility into network performance, thereby facilitating network performance planning, troubleshooting, and evaluation.

This feature provides the following performance metrics:

- Inter-packet delay variation (IPDV)
- Loss measurement
- Round-trip delay (RTT)
- Throughput measurement
- Two-way channel delay

Before you begin:

1. Configure the device interfaces.
2. Configure the autonomous system numbers and router IDs for the devices.
3. Configure the following protocols:
   - MPLS
   - OSPF
   - RSVP

To configure pro-active loss and delay measurements on the PE device:

1. Configure an associated bidirectional LSP to Router R2.

```
[edit protocols]
user@host# set mpls label-switched-path lsp-name to remote-router-ip-address
user@host# set mpls label-switched-path lsp-name install destination-prefix/prefix-length
active
user@host# set mpls label-switched-path lsp-name oam mpls-tp-mode
user@host# set mpls label-switched-path lsp-name ultimate-hop-popping
user@host# set mpls label-switched-path lsp-name associate-lsp remote-lsp-name
```

2. Create traffic classes for maintaining data traffic statistics per traffic class.

   This enables traffic class scoped loss and delay measurements.

```
[edit protocols]
user@host# set mpls statistics traffic-class-statistics
```

3. Configure performance monitoring at the querier side.

```
[edit protocols]
user@host# set mpls label-switched-path lsp-name oam performance-monitoring querier delay
traffic-class tc-value query-interval milliseconds
user@host# set mpls label-switched-path lsp-name oam performance-monitoring querier loss
traffic-class tc-value query-interval milliseconds
user@host# set mpls label-switched-path lsp-name oam performance-monitoring querier loss-
delay traffic-class tc-value query-interval milliseconds
```

4. Configure performance monitoring at the responder side.

```
[edit protocols]
user@host# set mpls label-switched-path lsp-name oam performance-monitoring responder delay
min-query-interval milliseconds
user@host# set mpls label-switched-path lsp-name oam performance-monitoring responder loss
min-query-interval milliseconds
```

RELATED DOCUMENTATION

Basic MPLS Configuration | 46

4

PART

MPLS LSPs

- Understanding MPLS LSPs | **521**
- Configuring MPLS LSPs | **603**

CHAPTER 8

Understanding MPLS LSPs

IN THIS CHAPTER

- LSP Overview | 521
- LSP Labels | 523
- LSP Routes | 575
- LSP Computation | 585
- LSP Routers | 590

LSP Overview

IN THIS SECTION

- How a Packet Travels Along an LSP | 521
- Types of LSPs | 522
- Scope of LSPs | 523

How a Packet Travels Along an LSP

When an IP packet enters an LSP, the ingress router examines the packet and assigns it a label based on its destination, placing the label in the packet's header. The label transforms the packet from one that is forwarded based on its IP routing information to one that is forwarded based on information associated with the label.

The packet is then forwarded to the next router in the LSP. This router and all subsequent routers in the LSP do not examine any of the IP routing information in the labeled packet. Rather, they use the label to look up information in their label forwarding table. They then replace the old label with a new label and forward the packet to the next router in the path.

When the packet reaches the egress router, the label is removed, and the packet again becomes a native IP packet and is again forwarded based on its IP routing information.

Types of LSPs

There are three types of LSPs:

- Static LSPs—For static paths, you must manually assign labels on all routers involved (ingress, transit, and egress). No signaling protocol is needed. This procedure is similar to configuring static routes on individual routers. Like static routes, there is no error reporting, liveliness detection, or statistics reporting.
- LDP-signaled LSPs—See "LDP Introduction" on page 1317.
- RSVP-signaled LSPs—For signaled paths, RSVP is used to set up the path and dynamically assign labels. (RSVP signaling messages are used to set up signaled paths.) You configure only the ingress router. The transit and egress routers accept signaling information from the ingress router, and they set up and maintain the LSP cooperatively. Any errors encountered while establishing an LSP are reported to the ingress router for diagnostics. For signaled LSPs to work, a version of RSVP that supports tunnel extensions must be enabled on all routers.

There are two types of RSVP-signaled LSPs:

- Explicit-path LSPs—All intermediate hops of the LSP are manually configured. The intermediate hops can be strict, loose, or any combination of the two. Explicit path LSPs provide you with complete control over how the path is set up. They are similar to static LSPs but require much less configuration.
- Constrained-path LSPs—The intermediate hops of the LSP are automatically computed by the software. The computation takes into account information provided by the topology information from the IS-IS or OSPF link-state routing protocol, the current network resource utilization determined by RSVP, and the resource requirements and constraints of the LSP. For signaled constrained-path LSPs to work, either the IS-IS or OSPF protocol and the IS-IS or OSPF traffic engineering extensions must be enabled on all routers.

There are three primary ways to signal a virtual private LAN service (VPLS) VPN in an MPLS network. You can power the VPLS using the BGP control plane, or using the LDP control plane, or using a method called FEC 129 which combines both methods of signaling.

One advantage of the BGP control plane is the ability for a provider edge (PE) router to automatically discover any other PE routers that take part in a given VPLS VPN. This is possible thanks to the exchange of route target information that identifies whether a router participates in a particular VPN. The result is that BGP can autonomously signal pseudowires between the PE routers participating in the same VPLS. This is in contrast to an LDP-signaled VPLS, where remote PEs must be defined explicitly.

As PE routers are added to and removed from the VPLS network, BGP can signal pseudowires to new PE routers. Similarly, BGP can tear down old pseudowires to old PE routers where the VPLS instance has been removed. Each PE router only needs to be configured with the relevant route target information of the VPLS routing instance. There is no need to configure neighbor PEs explicitly.

Note that all of this is completely independent of the choice of protocol to signal the transport label-switched paths (LSPs) that carry the VPN packets between PEs. For example, you are free to choose LDP, RSVP, SR-MPLS, or static LSPs. The choice of transport LSP protocol is entirely separate from the choice of VPLS signaling protocol. This is true of all methods of VPLS signaling.

Scope of LSPs

For constrained-path LSPs, the LSP computation is confined to one IGP domain, and cannot cross any AS boundary. This prevents an AS from extending its IGP into another AS.

Explicit-path LSPs, however, can cross as many AS boundaries as necessary. Because intermediate hops are manually specified, the LSP does not depend on the IGP topology or a local forwarding table.

RELATED DOCUMENTATION

MPLS Overview | 2

LSP Labels

IN THIS SECTION

- MPLS Label Overview | 524
- MPLS Label Allocation | 524
- Operations on MPLS Labels | 525
- Understanding MPLS Label Operations | 526
- Understanding MPLS Label Manager | 529
- Special MPLS Labels | 530
- Entropy Label Support in Mixed Mode Overview | 531
- Abstract Hops for MPLS LSPs Overview | 531
- Example: Configuring Abstract Hops for MPLS LSPs | 545
- Configuring the Maximum Number of MPLS Labels | 567

- Configuring MPLS to Pop the Label on the Ultimate-Hop Router | **569**
- Advertising Explicit Null Labels to BGP Peers | **570**
- Understanding MPLS Label Operations on EX Series Switches | **571**

MPLS Label Overview

Packets traveling along an LSP are identified by a label—a 20-bit, unsigned integer in the range 0 through 1,048,575. For push labels on ingress routers, no labels in this range are restricted. For incoming labels on the transit static LSP, the label value is restricted to 1,000,000 through 1,048,575.

On MX Series, PTX Series, and T Series routers, the value for entropy and flow labels is restricted to 16 through 1,048,575.

MPLS Label Allocation

In the Junos OS, label values are allocated per router or switch—the rest of this explanation uses router to cover both. The display output shows only the label (for example, `01024`). Labels for multicast packets are independent of those for unicast packets. Currently, the Junos OS does not support multicast labels.

Labels are assigned by downstream routers relative to the flow of packets. A router receiving labeled packets (the next-hop router) is responsible for assigning incoming labels. A received packet containing a label that is unrecognized (unassigned) is dropped. For unrecognized labels, the router does not attempt to unwrap the label to analyze the network layer header, nor does it generate an Internet Control Message Protocol (ICMP) destination unreachable message.

A packet can carry a number of labels, organized as a last-in, first-out stack. This is referred to as a *label stack*. At a particular router, the decision about how to forward a labeled packet is based exclusively on the label at the top of the stack.

Figure 26 on page 525 shows the encoding of a single label. The encoding appears after data link layer headers, but before any network layer header.

Figure 26: Label Encoding

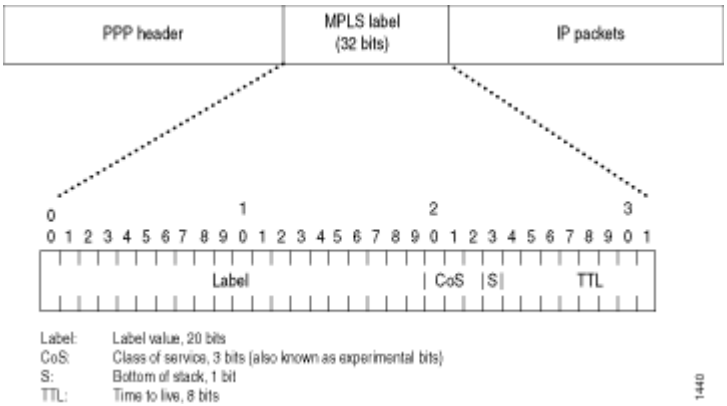
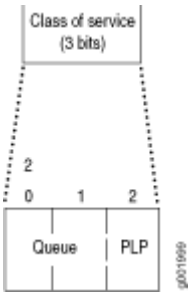


Figure 27 on page 525 illustrates the purpose of the class-of-service bits (also known as the EXP or experimental bits). Bits 20 and 21 specify the queue number. Bit 22 is the packet loss priority (PLP) bit used to specify the random early detection (RED) drop profile. For more information about *class of service* and the class-of-service bits, see "Configuring Class of Service for MPLS LSPs" on page 1951.

Figure 27: Class-of-Service Bits



Operations on MPLS Labels

The router supports the following label operations:

- Push—Add a new label to the top of the packet. For IPv4 packets, the new label is the first label. The time-to-live (TTL) and s bits are derived from the IP packet header. The MPLS *class of service* (CoS) is derived from the queue number. If the push operation is performed on an existing MPLS packet, you will have a packet with two or more labels. This is called label stacking. The top label must have its s bit set to 0, and might derive CoS and TTL from lower levels. The new top label in a label stack always initializes its TTL to 255, regardless of the TTL value of lower labels.
- Pop—Remove the label from the beginning of the packet. Once the label is removed, the TTL is copied from the label into the IP packet header, and the underlying IP packet is forwarded as a native IP packet. In the case of multiple labels in a packet (label stacking), removal of the top label yields

another MPLS packet. The new top label might derive CoS and TTL from a previous top label. The popped TTL value from the previous top label is not written back to the new top label.

- Swap—Replace the label at the top of the label stack with a new label. The S and CoS bits are copied from the previous label, and the TTL value is copied and decremented (unless the `no-decrement-ttl` or `no-propagate-ttl` statement is configured). A transit router supports a label stack of any depth.
- Multiple Push—Add multiple labels (up to three) on top of existing packets. This operation is equivalent to pushing multiple times.
- Swap and Push—Replace the existing top of the label stack with a new label, and then push another new label on top.

Understanding MPLS Label Operations

IN THIS SECTION

- MPLS Label-Switched Paths and MPLS Labels | **527**
- Reserved Labels | **527**
- MPLS Label Operations | **528**
- Penultimate-Hop Popping and Ultimate-Hop Popping | **529**

In the traditional packet-forwarding paradigm, as a packet travels from one switch to the next, an independent forwarding decision is made at each hop. The IP network header is analyzed and the next hop is chosen based on this analysis and on the information in the routing table. In an MPLS environment, the analysis of the packet header is made only once, when a packet enters the MPLS tunnel (that is, the path used for MPLS traffic).

When an IP packet enters a label-switched path (LSP), the ingress provider edge (PE) switch examines the packet and assigns it a label based on its destination, placing the label in the packet's header. The label transforms the packet from one that is forwarded based on its IP routing information to one that is forwarded based on information associated with the label. The packet is then forwarded to the next provider switch in the LSP. This switch and all subsequent switches in the LSP do not examine any of the IP routing information in the labeled packet. Rather, they use the label to look up information in their label forwarding table. They then replace the old label with a new label and forward the packet to the next switch in the path. When the packet reaches the egress PE switch, the label is removed, and the packet again becomes a native IP packet and is forwarded based on its IP routing information.

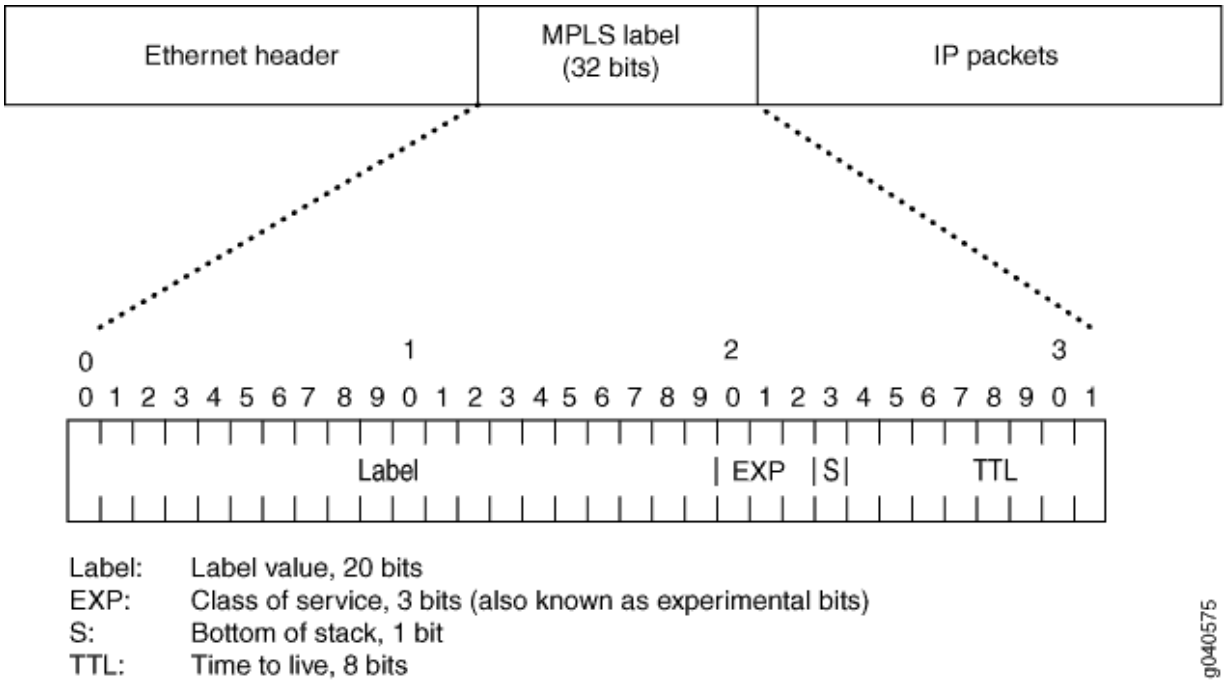
This topic describes:

MPLS Label-Switched Paths and MPLS Labels

When a packet enters the MPLS network, it is assigned to an LSP. Each LSP is identified by a label, which is a short (20-bit), fixed-length value at the front of the MPLS label (32 bits). Labels are used as lookup indexes for the label forwarding table. For each label, this table stores forwarding information. Because no additional parsing or lookup is done on the encapsulated packet, MPLS supports the transmission of any other protocols within the packet payload.

Figure 28 on page 527 shows the encoding of a single label. The encoding appears after data link layer headers, but before any network layer header.

Figure 28: Label Encoding



Reserved Labels

Labels range from 0 through 1,048,575. Labels 0 through 999,999 are for internal use.

Some of the reserved labels (in the range 0 through 15) have well-defined meanings. The following reserved labels are used by QFX Series and EX4600 devices:

- 0, IPv4 Explicit Null label—This value is valid only when it is the sole label entry (no label stacking). It indicates that the label must be popped on receipt. Forwarding continues based on the IP version 4 (IPv4) packet.

- 1, Router Alert label—When a packet is received with a top label value of 1, it is delivered to the local software module for processing.
- 3, Implicit Null label—This label is used in the signaling protocol (RSVP) only to request label popping by the downstream switch. It never actually appears in the encapsulation. Labels with a value of 3 must not be used in the data packet as real labels. No payload type (IPv4 or IPv6) is implied with this label.

MPLS Label Operations

QFX Series and EX4600 devices support the following MPLS label operations:

- Push
- Pop
- Swap

NOTE: There is a limit with regard to the number of labels that QFX and EX4600 devices can affix (push operations) to the label stack or remove (pop operations) from the label stack.

- For Push operations—As many as three labels are supported.
- For Pop operations—As many as three labels are supported.

The push operation affixes a new label to the top of the IP packet. For IPv4 packets, the new label is the first label. The time to live (TTL) field value in the packet header is derived from the IP packet header. The push operation cannot be applied to a packet that already has an MPLS label.

The pop operation removes a label from the beginning of the packet. Once the label is removed, the TTL is copied from the label into the IP packet header, and the underlying IP packet is forwarded as a native IP packet

The swap operation removes an existing MPLS label from an IP packet and replaces it with a new MPLS label, based on the following:

- Incoming interface
- Label
- Label forwarding table

Figure 29 on page 529 shows an IP packet without a label arriving on the customer edge interface (ge-0/0/1) of the ingress PE switch. The ingress PE switch examines the packet and identifies that packet's destination as the egress PE switch. The ingress PE switch applies label 100 to the packet and sends the MPLS packet to its outgoing MPLS core interface (ge-0/0/5). The MPLS packet is transmitted

on the MPLS tunnel through the provider switch, where it arrives at interface ge-0/0/5 with label 100. The provider switch swaps label 100 with label 200 and forwards the MPLS packet through its core interface (ge-0/0/7) to the next hop on the tunnel, which is the egress PE switch. The egress PE switch receives the MPLS packet through its core interface (ge-0/0/7), removes the MPLS label, and sends the IP packet out of its customer edge interface (ge-0/0/1) to a destination that is beyond the tunnel.

Figure 29: MPLS Label Swapping

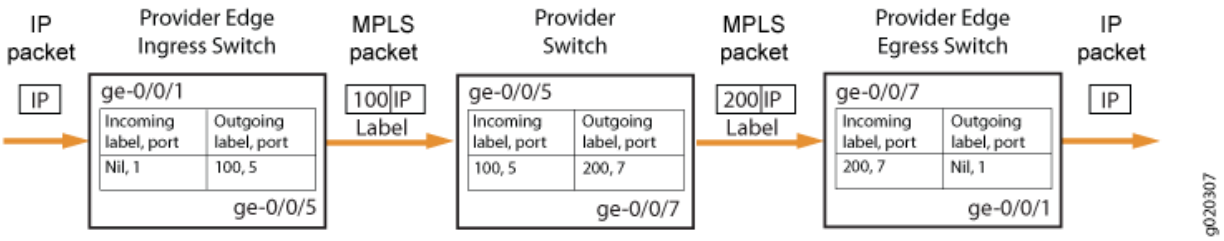


Figure 29 on page 529 shows the path of a packet as it passes in one direction from the ingress PE switch to the egress PE switch. However, the MPLS configuration also allows traffic to travel in the reverse direction. Thus, each PE switch operates as both an ingress switch and an egress switch.

Penultimate-Hop Popping and Ultimate-Hop Popping

The switches enable penultimate-hop popping (PHP) by default with IP over MPLS configurations. With PHP, the penultimate provider switch is responsible for popping the MPLS label and forwarding the traffic to the egress PE switch. The egress PE switch then performs an IP route lookup and forwards the traffic. This reduces the processing load on the egress PE switch, because it is not responsible for popping the MPLS label.

- The default advertised label is label 3 (Implicit Null label). If label 3 is advertised, the penultimate-hop switch removes the label and sends the packet to the egress PE switch.
- If ultimate-hop popping is enabled, label 0 (IPv4 Explicit Null label) is advertised and the egress PE switch of the LSP removes the label.

Understanding MPLS Label Manager

MPLS label manager is used to manage different label types such as LSI, dynamic, block, and static, which are supported on platforms using Modular Port Concentrators (MPCs) equipped with Junos Trio chipsets. These line cards provide more flexibility and scalability, when the `enhanced-ip` command is configured on the device.

The existing behavior of `label-space` command is retained, which is *not recommended*. To provide additional functionality such as multiple ranges for each type of label, `label-range` command is introduced under the `[edit protocols mpls label usage]` hierarchy, which is independent of `label-space` configuration. You can choose either style if only one range is needed for each type of label.

The following features are optimized with the `enhanced-ip` command configured on the device:

- Allows you to define the system wide global label pool to be used by segment-routing global block (SRGB) through IS-IS routing protocol.
- Increases the `vrf-table-label` space to at least 16,000, if the platform can support the scale.
- Allows you to specify the label value to be used by static VRF table label.
- Allows you to specify the label value range to be used by supported label application types.
- Allows you to change dynamically the SRGB and label type ranges.

Special MPLS Labels

Some of the reserved labels (in the 0 through 15 range) have well-defined meanings. For more complete details, see RFC 3032, *MPLS Label Stack Encoding*.

- 0, IPv4 Explicit Null label—This value is legal only when it is the sole label entry (no label stacking). It indicates that the label must be popped upon receipt. Forwarding continues based on the IP version 4 (IPv4) packet.
- 1, Router Alert label—When a packet is received with a top label value of 1, it is delivered to the local software module for processing.
- 2, IPv6 Explicit Null label—This value is legal only when it is the sole label entry (no label stacking). It indicates that the label must be popped on receipt. Forwarding continues based on the IP version 6 (IPv6) packet.
- 3, Implicit Null label—This label is used in the control protocol (LDP or RSVP) only to request label popping by the downstream router. It never actually appears in the encapsulation. Labels with a value of 3 should not be used in the data packet as real labels. No payload type (IPv4 or IPv6) is implied with this label.
- 4 through 6—Unassigned.
- 7, Entropy label indicator—This label is used when an Entropy label is in the label stack and precedes the Entropy label.
- 8 through 15—Unassigned.

Special labels are commonly used between the egress and penultimate routers of an LSP. If the LSP is configured to carry IPv4 packets only, the egress router might signal the penultimate router to use 0 as a final-hop label. If the LSP is configured to carry IPv6 packets only, the egress router might signal the penultimate router to use 2 as a final-hop label.

The egress router might simply signal the penultimate router to use 3 as the final label, which is a request to perform penultimate-hop label popping. The egress router will not process a labeled packet; rather, it receives the payload (IPv4, IPv6, or others) directly, reducing one MPLS lookup at egress.

For label-stacked packets, the egress router receives an MPLS label packet with its top label already popped by the penultimate router. The egress router cannot receive label-stacked packets that use label 0 or 2. It typically requests label 3 from the penultimate router.

Entropy Label Support in Mixed Mode Overview

Starting with Junos OS Release 14.2, entropy label is supported in mixed mode chassis where the entropy label can be configured without enhanced-ip configuration. The entropy label helps transit routers load-balance MPLS traffic across ECMP paths or link aggregation groups. The entropy label introduces a load-balancing label to be used by routers to load balance traffic rather than relying on deep packet inspection, reducing the packet processing requirements in the forwarding plane at the expense of increased label stack depth. Junos OS supports the entropy label only for MX Series routers with MPCs or MICs and can be enabled with enhanced-ip mode. But, this leads to a packet drop if the core-facing interface has an entropy label configured on the MPC or MIC and the other end of this core-facing connection has a DPC line card. In order to avoid this, the entropy label is now supported in mixed mode where the entropy label can be configured without enhanced-ip configuration. This allows MX Series router DPCs to support a pop out entropy label. However, this does not support a flow label.

Abstract Hops for MPLS LSPs Overview

IN THIS SECTION

- Understanding Abstract Hops | **531**
- Benefits of Using Abstract Hops | **532**
- Junos OS Implementation of Abstract Hops | **534**

An abstract hop is a logical combination of the existing traffic engineering constraints, such as administrative groups, extended administrative groups, and Shared Risk Link Groups (SRLGs), which results in a user-defined group or cluster of routers that can be sequenced and used as constraints for setting up an MPLS label-switched path (LSP). Abstract hops overcome the limitations of existing path constraint specifications and provide several benefits to the traffic engineering capabilities of MPLS.

Understanding Abstract Hops

The path constraint for setting up of an MPLS LSP can be specified as either individual routers in the form of real hops or as a set of routers by way of administrative group or color specification. When a

path constraint uses real hops (strict or loose), the LSP is set up along a specified sequence of routers (for example, R1, R2, … R*n*). When a path constraint uses an administrative group or color specification, a group of routers that meet the specified criteria is used to set up the LSP without picking a specific router, and unlike real-hop constraint, there is no sequence among the different groups of routers used in the constraint.

The drawback of real-hop constraint is that in a failure scenario, if any of the router hops goes down or the bandwidth utilization of the attached interface gets saturated, the path goes down (or relies on local or end-to-end protection). Although other alternative routers might be available to recover or set up the LSP, the LSP remains down until the operator configures another router hop sequence as the path constraint to bring the path up again or to disengage the protection path.

The administrative group or color specification constraint overcomes this limitation of a real-hop constraint to a certain extent. Here, when one of the routers in the group goes down or has its link capacity saturated, setting up of the LSP is not affected. This is because the next hop router to be used in the path constraint is not picked beforehand, and the LSP is set up along other routers that have the same administrative group or color without operator intervention. However, the drawback with router group constraints is that a sequence cannot be specified among the hop constraints.

Abstract hops overcome these drawbacks by creating user-defined router groups, where each member router meets a user-defined constraint. The user-defined constraint is a logical combination of the existing traffic engineering constraints, such as administrative groups, extended administrative groups, and Shared Risk Link Groups (SRLGs). Ordering is achieved among the router groups by specifying a sequence of abstract hops used in a path constraint. As a result, abstract hops combine the ordering property of real-hop constraint specification and the resilience that comes with the other traffic engineering constraints.

A path can use a combination of real and abstract hops as constraints. When using abstract hops, instead of specifying a sequence of routers (R1, R2, … R*n*) as with real hops, you specify an ordered set of router groups or abstract hops (G1, G2, … G*n*) as the path constraint. Each specified router group, G*i* for example, consists of some user-defined set of routers—R1, R2, R*j*, … R*n*. When one of the routers in the group goes down, say Router R*j* in group G*i*, another router, say Router R*k*, from the same group G*i* is picked up by path computation to replace the router that went down (that is, Router R*j*). This is because the path constraint is sequenced and has to go through a sequence of abstract hops, instead of a sequence of individual routers.

Benefits of Using Abstract Hops

Abstract hops are user-defined router groups. Similar to real-hop constraints that use a sequence of individual routers, a sequence of abstract hops can be used for setting up a label-switched path (LSP). The use of abstract hops provides resiliency to sequenced path constraints. The other benefits of using abstract hops include:

Specifying a Sequence of Constraint Combinations

Currently, it is possible to specify a path that can go through links that satisfy multiple attributes. Such a path constraint is called a compound constraint combination; for example, a constraint (Ci) that includes low latency links of green color and also excludes SRLG north.

However, there is no support for specifying a path with a sequence of compound constraint combinations. For example, a sequenced constraint (C1, C2, Ci, ...Cn) that includes low latency green links, no latency blue links, and then low latency red links.

The need for such a sequenced compound constraint combination arises when there is a requirement to establish paths through a sequence of geographical regions with a different link affinity (attributes) requirement in each region. Abstract hops meet this requirement by allowing computing nodes to map each constraint combination (Ci, for example) with the user-defined group of routers—that is, the abstract hops.

Avoiding New Network Configuration on Transit Nodes

With current path constraint specification capabilities, it is possible to include or exclude links of certain attributes along an entire path; for example, excluding SRLG west from a path. However, there is no support to either conditionally exclude or include attributes, or to apply different exclude or include attributes in different parts of the path; for example, excluding SRLG west only when traversing red links.

As a workaround, a new administrative group can be created to identify all such red links that do not have SRLG west, and configure all the relevant links appropriately with that administrative group. The drawback of this approach is that configuration changes are required throughout the network to reflect the new administrative group membership.

Instead, by using abstract hops, the configuration changes can be contained on the ingress router only. At the ingress router, the constraint combination is mapped to the abstract hop, thereby meeting the aforementioned requirement without the need for any new configuration on the transit nodes.

Combining Centralized and Distributed Path Computation Paradigms

Traffic engineering of MPLS paths can be achieved by distributed computing or with a centralized controller for computing paths. A combination of both the computation types is called the hybrid computation paradigm. The key feature of the hybrid computation approach is the ability of the centralized controller—referred to as a Path Computation Element (PCE)—to loosely specify the path computation directives, per path, to the ingress router—referred to as a Path Computation Client (PCC)—and the ability of the ingress router to use it as input for path computation.

A sequence of abstract hops serves the purpose of acting as the guideline from the centralized controller. Abstract hops provide the flexibility to the controller to weave into the path constraint and

attributes. This also enables the controller to build in the element of sequence in the constraint. The controller does not have to specify each hop the path needs to take, leaving room for the ingress router to act within the limits of the guideline or directive.

Table 12 on page 534 lists the key features of the hybrid computation paradigm and provides a comparison of this approach with the current path computation methods.

Table 12: Hybrid Computation for Abstract Hops

Features	Distributed Constrained Shortest Path First	Centralized Constrained Shortest Path First	Hybrid Constrained Shortest Path First
React to frequent changes in a large network	Yes		Yes
Sophisticated path computation with global view		Yes	Yes
Incorporation of business logic in path computation		Yes	Yes
Resilience (no single point of failure)	Yes		Yes
Predictability		Yes	Yes
React to network load in (close to) real time	Yes		Yes
Field tested (versus early adoption)	Yes		Yes

Junos OS Implementation of Abstract Hops

The order-aware abstract hops feature is introduced in Junos OS Release 17.1. The following sections describe the implementation of abstract hops in Junos OS:

Defining Abstract Hops

An abstract hop is a group of routers that users can define to be used in setting up a label-switched path (LSP). The user can control which routers to include in the group by defining a logical combination of heterogeneous link attributes or constraints called constituent attributes. The routers with links that satisfy the defined constituent attributes make it to the group of routers representing the abstract hop.

The mapping of constituent attributes with the abstract hop is local to the computing node or the ingress of the LSP being setup. As a result, abstract hops do not have associated interior gateway protocol updates or signaling protocol extensions, and implementing abstract hops in a network does not require new configuration on the transit nodes.

A constituent list enables defining of a set of constituent traffic engineering attributes, that is identified by a user-defined name. Constituent lists are used in an abstract hop definition by using any of the following configuration statements:

- **include-any-list**—Link satisfies the constituent-list if any of the specified constituent attributes are true for the link.
- **include-all-list**—Link satisfies the constituent-list if all the specified constituent attributes are true for the link.
- **exclude-all-list**—Link satisfies the constituent-list if none of the specified constituent attributes are true for the link.
- **exclude-any-list**—Link satisfies the constituent-list if at least one of the specified constituent attributes is not true for the link.

An abstract hop is defined as a logical combination of constituent-list references that can belong to any of the aforementioned categories. To achieve this, logical operators `AND` and `OR` are included in the abstract hop definition, and applied to the constituent list.

- **OR**—At least one of the constituent-list references in the abstract hop definition must be satisfied by a link for the attached node to be part of the abstract hop.
- **AND**—All of the constituent-list references in the abstract hop definition must be satisfied by a link for the attached node to be part of the abstract hop.

Sample Abstract Hop Definition

Taking as an example, the definition of abstract hops hopA is as follows:

Abstract hops hopA must include all routers whose emanating links satisfy the logical combination of the following link attributes, respectively:

- **hopA**—((administrative group red && Srlg south) || (administrative group green || Srlg north)), where:

- *administrative group red* and *Srlg south* belong to include-all constituent list (listA1, in this example).
- *administrative group green* and *Srlg north* belong to include-any constituent list (listA2, in this example).
- //is the OR operator.

The configuration for abstract hops hopA is as follows:

- **hopA configuration**

```
[edit protocols mpls]
Constituent-list listA1 {
    administrative-group red;
    Srlg south;
}
Constituent-list listA2 {
    administrative-group green;
    Srlg north;
}
Abstract-hop hopA{
    Operator OR;
    Constituent-list listA1 include-all-list;
    Constituent-list listA2 include-any-list;
}
```

Verifying Abstract Hop Configuration

The `show mpls abstract hop membership <abstract hop name>` command is used to view members of an abstract hop. The command output provides the abstract hop to traffic engineering database node mapping.

```
user@host> show mpls abstract-hop-membership
Abstract hop: hop1A
   Credibility: 0
Address: 128.102.165.105
Address: 128.102.166.237
Address: 128.102.168.0
Address: 128.102.173.123

Abstract hop: hopB
   Credibility: 0
Address: 128.102.160.211
```

```
Address: 128.102.165.5
Address: 128.102.166.237
Address: 128.102.172.157
Address: 128.102.172.196
```

Here, the output field `Credibility` indicates the credibility associated with interior gateway protocol in use.

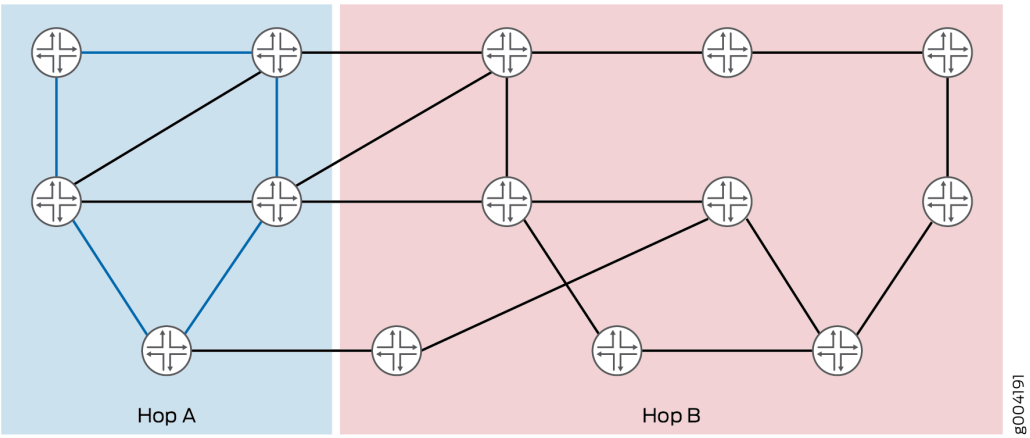
The output of the `show ted database extensive local` command provides the view captured in traffic engineering database. A keyword `local` is added to indicate that the output would include any local instrumentation. The command output shows the abstract hop as an attribute of links that satisfy the associated logical combination of link attributes.

```
user@host> show ted database extensive local
TED database: 0 ISIS nodes 8 INET nodes
NodeID: 128.102.173.123
  Type: Rtr, Age: 3098 secs, LinkIn: 4, LinkOut: 3
  Protocol: OSPF(0.0.0.0)
    To: 128.102.168.0, Local: 1.3.0.1, Remote: 1.3.0.2
      Local interface index: 332, Remote interface index: 0
      Color: 0x2 green
      Abstract hops: hopA
      Metric: 1
      Static BW: 1000Mbps
      Reservable BW: 1000Mbps
      Available BW [priority] bps:
          [0] 970Mbps      [1] 970Mbps     [2] 970Mbps     [3] 970Mbps
          [4] 970Mbps      [5] 970Mbps     [6] 970Mbps     [7] 970Mbps
      Interface Switching Capability Descriptor(1):
        Switching type: Packet
        Encoding type: Packet
        Maximum LSP BW [priority] bps:
          [0] 970Mbps      [1] 970Mbps     [2] 970Mbps     [3] 970Mbps
          [4] 970Mbps      [5] 970Mbps     [6] 970Mbps     [7] 970Mbps
    To: 128.102.165.105, Local: 1.1.0.1, Remote: 1.1.0.2
      Local interface index: 330, Remote interface index: 0
      Srlg: south
      Abstract hops: hopB
      Metric: 1
      Static BW: 1000Mbps
      Reservable BW: 1000Mbps
      Available BW [priority] bps:
```

```
      [0] 960Mbps      [1] 960Mbps      [2] 960Mbps      [3] 960Mbps
      [4] 960Mbps      [5] 960Mbps      [6] 960Mbps      [7] 960Mbps
  Interface Switching Capability Descriptor(1):
    Switching type: Packet
    Encoding type: Packet
    Maximum LSP BW [priority] bps:
      [0] 960Mbps      [1] 960Mbps      [2] 960Mbps      [3] 960Mbps
      [4] 960Mbps      [5] 960Mbps      [6] 960Mbps      [7] 960Mbps
```

Abstract hop hopA is for low latency AND SRLG west, and abstract hop hopB is for excluding SRLG west. Figure 30 on page 538 displays the ingress view of these abstract hops.

Figure 30: Ingress View of Abstract Hops



Using Abstract Hops in Path Constraint

The user associates a unique identifier with each abstract hop definition. This identifier is used for referring to the abstract hop in the path constraint. A sequence of abstract hops can be specified as the path constraint, similar to how real IP hops are used. The path constraint could also be a sequence of abstract hops interleaved by real IP hops.

Using abstract hops or real hops in a path constraint requires more than one Constrained Shortest Path First pass to the destination, typically one pass per hop. When real hops are provided as the path constraint, the constraint computation involves as many passes as the number of hops in the path constraint, where each pass ends on reaching a hop in the constraint list. The starting point for each pass is the destination of the previous pass, with the first pass using the ingress router as the start.

Alternatively, when path constraint uses strict or loose abstract hops, constraint computation comprises passes where each pass processes the subsequent abstract hop in the constraint list. In such a case,

more than one node qualifies to be the destination for the pass. The set of nodes is called the viable router set for the pass.

An abstract hop traverses member nodes by using the following:

- Links that satisfy the logical combination of defined constituent attributes
- Any kind of links

The means of abstract hops traversing the member nodes is controlled by the use of the abstract hop qualifiers—strict, loose, and loose-link—in defining the path constraint. Taking for example, abstract hop hopA is processed differently with different qualifiers:

- **Strict**—After the last processed hop in the constraint list, the path traverses only links or nodes having membership of abstract hop hopA, before reaching a node with hopA's membership that is a feasible starting point for processing the next abstract hop.
- **Loose**—After the last processed hop in the constraint list, the path can traverse any real nodes that do not have abstract hop membership of hopA, before reaching a node with abstract hop membership hopA, which is a feasible starting point for processing the next abstract hop.
- **Loose-link**—After the last processed hop in the constraint list, the path can traverse any real nodes that do not have abstract hop membership of hopA, before reaching a node with abstract hop membership hopA, which is a feasible starting point for processing the next abstract hop. But the path should have traversed at least one link of abstract hop hopA membership in the course of the same.

  In other words, the abstract hop of type loose-link is said to be processed only if any of the viable routers in the constraint is reachable through a link of associated abstract hop membership.

Sample Abstract Hops Specification

Table 13 on page 540 provides sample use case for using abstract hops in path constraints.

Table 13: Using Abstract Hops in Path Constraints

Purpose of Path Constraint	Abstract Hop Qualifier	Configuration	Viable Router Set	Affinity

Traverse nodes that are members of hopA taking only links that satisfy hopA.	Strict	`[edit protocols mpls]`
		`Path path_hopA_s {`
		`  hopA abstract strict;`
		`}`
			All members of abstract hopA. That is, A1, A2...A*n*.	hopA (pick only links that satisfy abstract hopA).

Traverse nodes that are members of hopA but not necessarily links that satisfy hopA	Loose	`[edit protocols mpls]`
		`Path path_hopA_l {`
		`  hopA abstract loose;`
		`}`
			All members of abstract hopA. That is, A1, A2...A*n*.	None (any kind of links).

Traverse nodes that are members of hopA by taking at least one link that satisfies hopA.	Loose-link
	
	NOTE: The loose-link qualifier is viewed as loose followed by strict for the same abstract hop. In other words, hopA loose-link is the same as hopA loose and hopA strict.
		`[edit protocols mpls]`
		`Path path_hopA_ll {`
		`  hopA abstract loose-link;`
		`}`
			In this case, there are two computation passes associated with hopA in the path constraint. The viable router set for both passes is:
			
			All members of abstract hopA. That is, A1, A2...A*n*.
			
			NOTE: During path computation, a router is traversed only once.
				In this case, there are two computation passes associated with hopA in the path constraint. The affinity for the two passes is:
				
				• Pass 1—None (any kind of links).
				
				• Pass 2—hopA (pick only links that satisfy abstract hopA).

Table 13: Using Abstract Hops in Path Constraints *(Continued)*

Purpose of Path Constraint	Abstract Hop Qualifier	Configuration	Viable Router Set	Affinity

Traverse nodes that are members of hopA, taking only links that satisfy hopA, followed by nodes that are members of hopB taking only links that satisfy hopB.	Strict	`[edit protocols mpls]`
		`Path path_hopA_hopB_s {`
		`  hopA abstract strict;`
		`  hopB abstract strict;`
		`}`
			• hopA—Intersection of member set of hopA and hopB.
			NOTE: When an abstract hop is followed by a strict abstract hop, the intersection of the two member sets is considered as viable router set.
			• hopB—All members of abstract hopB. That is, B1, B2… B*n*.
				• hopA—hopA (pick only links that satisfy abstract hopA).
				• hopB—hopB (pick only links that satisfy abstract hopB).

Traverse nodes that are members of hopA taking only links that satisfy hopA, followed by nodes that are members of hopB taking any kind of links.	Strict and loose	`[edit protocols mpls]`
		`Path path_hopA_s_hopB_l {`
		`  hopA abstract strict;`
		`  hopB abstract loose;`
		`}`
			• hopA—All members of abstract hopA. That is, A1, A2… A*n*.
			• hopB—All members of abstract hopB. That is, B1, B2… B*n*.
				• hopA—hopA (pick only links that satisfy abstract hopA).
				• hopB—None (pick any links).

Table 13: Using Abstract Hops in Path Constraints *(Continued)*

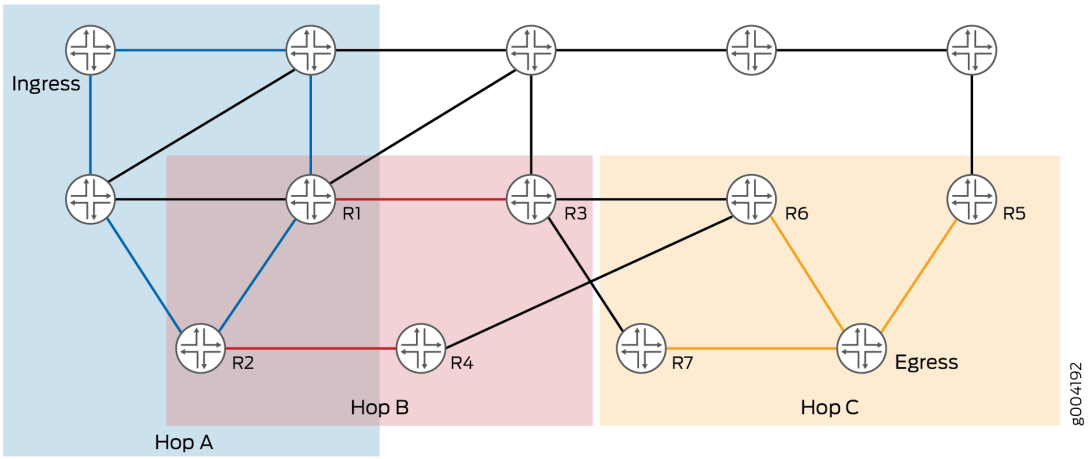
Purpose of Path Constraint	Abstract Hop Qualifier	Configuration	Viable Router Set	Affinity

Traverse nodes that are members of hopA by taking any kinds of links, followed by nodes that are members of hopB taking any kind of links.	Loose	`[edit protocols mpls]`
		`Path path_hopA_l_hopB_l {`
		`  hopA abstract loose;`
		`  hopB abstract loose;`
		`}`
			• hopA—All members of abstract hopA. That is, A1, A2… A*n*.
			• hopB—All members of abstract hopB. That is, B1, B2… B*n*.
				None (pick any links).

Traverse nodes that are members of hopA by taking any kinds of links, followed by nodes that are members of hopB taking only links that satisfy hopB.	Loose and strict	`[edit protocols mpls]`
		`Path path_hopA_l_hopB_s {`
		`  hopA abstract loose;`
		`  hopB abstract strict;`
		`}`
			• hopA—Intersection of the members of hopA and hopB.
			When an abstract hop is followed by a strict abstract hop, the intersection of the two member sets is considered as viable router set.
			• hopB—All members of abstract hopB. That is, B1, B2… B*n*.
				• hopA—None (pick any links).
				• hopB—hopB (pick only links that satisfy abstract hopB).

Figure 31 on page 543 displays path constraints for abstract hops hopA, hopB, and hopC with loose, strict, and loose abstract hop qualifiers, respectively.

Figure 31: Sample Path Constraints for Abstract Hops



The Constrained Shortest Path First passes for the abstract hops are as follows:

- Pass 1 associated with hopA
  - Viable routers—Routers R1 and R2 (intersection of hopA and hopB, as hopB is a strict abstract hop).
  - Affinity—None (as hopA is loose).
- Pass 2 associated with hopB
  - Viable routers—Routers R1, R2, R3, and R4
  - Affinity—Pick only hopB-compliant links (as hopB is a strict abstract hop).
- Pass 3 associated with hopC
  - Viable routers—Routers R5, R6, R7, and the egress router.
  - Affinity—None (as hopC is a loose abstract hop).

Path Computation and Backtracking

In each Constrained Shortest Path First pass, when the nearest router from a viable router set is reached using links satisfying the affinity figured for the pass, the abstract hop associated with the pass is said to be processed. The viable router thus reached serves as the start for the next constraint pass. If any constraint pass fails, and it is not the one with the ingress router as start router, then the pass is backtracked to the previous pass and the process is repeated.

Sample Backtracking

When a Constrained Shortest Path First pass p (other than the first one) fails, the exit router of the previous pass (p – 1) that served as start for the current pass p is disqualified in the viable router set of the previous pass (p – 1). Then the previous pass (p – 1) is re-executed to find the next best exit router or destination for the pass p – 1 from the viable router set.

The router thus determined serves as the new start router for the pass p. This procedure is repeated as long as there are failures and there are viable routers that are not explored.

The `show mpls lsp abstract-hop-computation name` *`lsp-name`* command provides the various computation passes involved per LSP and the qualifying exit routers for each pass. The command output also gives the affinity per pass, and shows the current start router chosen for the pass. For each viable router, the state of backtracking is displayed, where it can be either valid or disqualified.

```
user@host> show mpls lsp abstract-computation
Path computation using abstract hops for LSP: lsp1
Path type: Primary, Path name: path1

Credibility: 0, Total no of CSPF passes: 2
CSPF pass no: 0 Start address of the pass: 128.102.173.123
Affinity: hopA
CSPF pass no: 1 Start address of the pass: 0.0.0.0
Destination: 128.102.172.157, , State: VALID

Path type: Standby, Path name: path2

Credibility: 0, Total no of CSPF passes: 3
CSPF pass no: 0 Start address of the pass: 128.102.173.123
Destination: 128.102.166.237, , State: VALID
Affinity: hopA
CSPF pass no: 1 Start address of the pass: 128.102.166.237
Destination: 128.102.160.211, , State: VALID
Destination: 128.102.165.5, , State: VALID
Destination: 128.102.166.237, , State: VALID
Destination: 128.102.172.157, , State: VALID
Destination: 128.102.172.196, , State: VALID
Affinity: hopB
CSPF pass no: 2 Start address of the pass: 128.102.172.196
Destination: 128.102.172.157, , State: VALID
```

The output field `Credibility` indicates the credibility associated with the interior gateway protocol in use.

Example: Configuring Abstract Hops for MPLS LSPs

IN THIS SECTION

- Requirements | 545
- Overview | 546
- Configuration | 548
- Verification | 565

This example shows how to configure abstract hops for MPLS label-switched paths (LSPs). Abstract hops combine the key features of existing traffic engineering constraints that enables the user to specify an order-aware and resilient path constraint for MPLS LSPs.

Requirements

This example uses the following hardware and software components:

- Six devices that can be a combination of M Series Multiservice Edge Routers, MX Series 5G Universal Routing Platforms, T Series Core Routers, and PTX Series Packet Transport Routers.
- Junos OS Release 17.1 or later running on all the devices.

Before you begin:

- Configure the device interfaces.
- Configure the device router ID and assign an autonomous system (AS) number.
- Configure RSVP on all the devices.
- Configure OSPF or any other interior gateway protocol on all the devices.
- Configure administrative groups, extended administrative groups, and Shared Risk Link Groups (SRLGs) on all the devices.

Overview

IN THIS SECTION

- Topology | 547

Junos OS Release 17.1 introduces abstract hops, which are user-defined router clusters or groups. Similar to the sequence of real-hop constraints (strict or loose), a sequence of abstract hops can be used for setting up a label-switched path (LSP). A path can use a combination of real and abstract hops as constraints.

An abstract hop is a logical combination of the existing traffic engineering constraints, such as administrative groups, extended administrative groups, and SRLGs, along with the ordering property of real hops. As a result, when a sequence of abstract hops is used in a path constraint, ordering is achieved among the groups of routers that meet a logical combination of link or node attributes called constituent attributes.

To configure abstract hops:

- Create constituent lists with constituent traffic engineering attributes by including the `constituent-list` *`list-name`* statement at the `[edit protocols mpls]` hierarchy level.
- Include the constituent lists in the abstract hop definition at the `[edit protocols mpls abstract-hop` *`abstract-hop-name`*`]` hierarchy level.
- Define path constraints that use abstract hops at the `[edit protocols mpls path` *`path-name`*`]` hierarchy level.

Take the following guidelines under consideration when configuring abstract hops for MPLS LSPs:

- Abstract hops are supported only in the master routing instance of a device.
- IPv6 destinations are not supported in abstract hop constraints (only IPv4 destinations work).
- Abstract hops can be strict or loose constraints.
- Abstract hops support in Junos OS Release 17.1 is provided only for intra-area MPLS LSPs and not for inter-domain, or inter-area LSPs.
- Abstract hop constraints is enabled for regular point-to-point LSPs only. Other types of MPLS LSPs, such as point-to-multipoint LSPs, externally controlled bidirectional LSPs, dynamic container LSPs, RSVP automesh LSPs, and inter-area LSPs are not supported with abstract hops configuration.
- Abstract hops do not enable computation of overall shortest path for LSPs.

- An abstract hop must not be referred to more than once in the same path constraint.
- Abstract hop constraint specifications do not affect the support for Graceful Routing Engine switchover (GRES), unified in-service software upgrade (ISSU), and nonstop routing (NSR).
- Abstract hop constraint specifications do not affect overall network performance. However, the time taken for constrained shortest path first computation increases with abstract hop configuration. The setup time for an abstract hop LSP is more than the time taken to set up an LSP without abstract hop configuration.

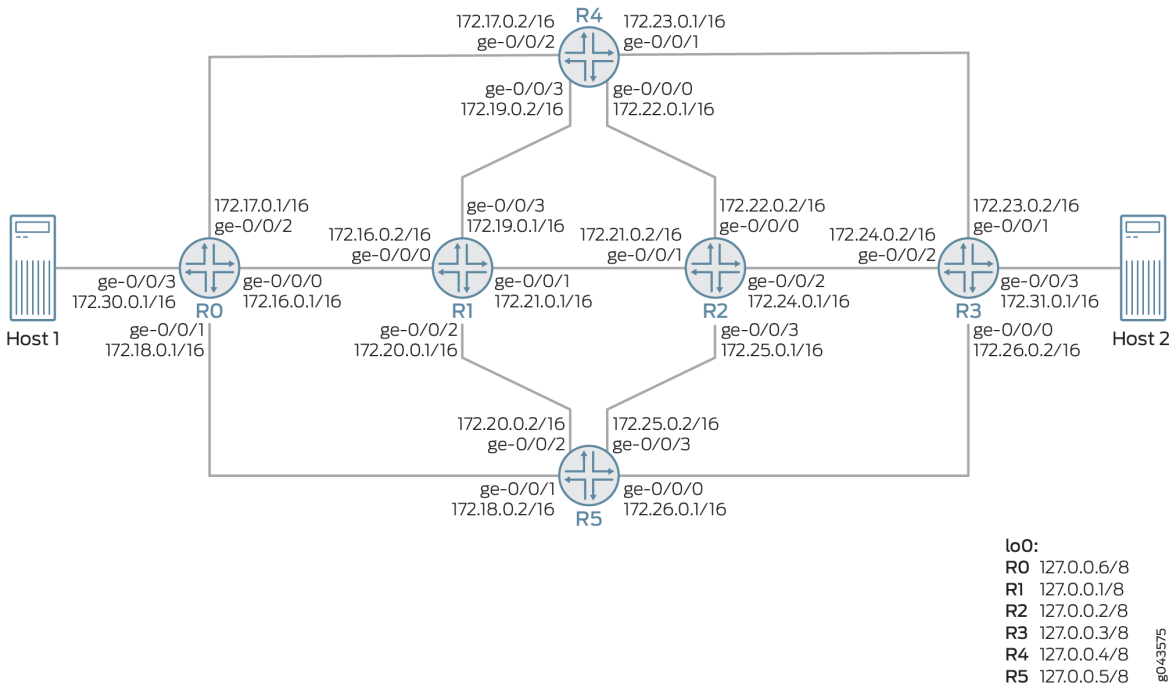
Topology

Figure 32 on page 548 illustrates a sample network topology configured with abstract hops. Devices R0 and R3 are each connected to hosts (Host 1 and Host 2). Devices R4 and R5 are each connected to Devices R0, R1, R2, and R3. Devices R1 and R2 are also directly connected to each other.

Devices R0 and R3 are configured under the same autonomous system—AS 64496. An MPLS LSP is configured from Device R0 through Device R3 with one primary path and two secondary paths (standby and nonstandby secondary paths).

Four constituent lists—c1, c2, c3, and c4—are created using three SRLGs (g1, g2, and g3), three administrative groups (green, blue, and red), and one extended administrative group (gold). Three abstract hops (ah1, ah2, and ah3) are defined using the configured constituent lists, and are specified as path constraints. Abstract hop ah1 is specified as constraint for the primary path, while abstract hops ah2 and ah3 are specified as constraints for the secondary standby path and the secondary nonstandby path, respectively.

Figure 32: Configuring Abstract Hop Path Constraint



Configuration

IN THIS SECTION

- CLI Quick Configuration | 548
- Procedure | 555

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

Device R0

```
set chassis network-services ip
set interfaces ge-0/0/0 unit 0 family inet address 172.16.0.1/16
```

```
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 172.18.0.1/16
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 172.17.0.1/16
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 family inet address 172.30.0.1/16
set interfaces lo0 unit 0 family inet address 127.0.0.6/8
set routing-options srlg g1 srlg-value 100
set routing-options srlg g1 srlg-cost 1000
set routing-options srlg g2 srlg-value 200
set routing-options srlg g2 srlg-cost 2000
set routing-options srlg g3 srlg-value 300
set routing-options srlg g3 srlg-cost 3000
set routing-options administrative-groups-extended-range minimum 50000
set routing-options administrative-groups-extended-range maximum 60000
set routing-options administrative-groups-extended gold group-value 50000
set routing-options router-id 127.0.0.6
set routing-options autonomous-system 64496
set routing-options forwarding-table export test
set protocols rsvp interface all aggregate
set protocols rsvp interface fxp0.0 disable
set protocols rsvp interface ge-0/0/0.0 bandwidth 80m
set protocols rsvp interface ge-0/0/2.0 bandwidth 200m
set protocols rsvp interface ge-0/0/1.0 bandwidth 500m
set protocols mpls administrative-groups green 0
set protocols mpls administrative-groups blue 1
set protocols mpls administrative-groups red 2
set protocols mpls label-switched-path R0-R31 to 127.0.0.3
set protocols mpls label-switched-path R0-R31 primary prim
set protocols mpls label-switched-path R0-R31 secondary stdby standby
set protocols mpls label-switched-path R0-R31 secondary nonstdby
set protocols mpls path path_primary 172.16.0.2 strict
set protocols mpls path path_primary 172.21.0.2 strict
set protocols mpls path path_primary 172.24.0.2 strict
set protocols mpls path path_ter_nonstdby 172.18.0.1 strict
set protocols mpls path path_ter_nonstdby 172.26.0.2 strict
set protocols mpls path path_sec_stdby 172.17.0.2 strict
set protocols mpls path path_sec_stdby 172.23.0.2 strict
set protocols mpls path prim ah1 abstract
set protocols mpls path prim ah1 strict
set protocols mpls path stdby ah2 abstract
set protocols mpls path stdby ah2 strict
set protocols mpls path nonstdby ah3 abstract
```

```
set protocols mpls path nonstdby ah3 strict
set protocols mpls constituent-list c1 srlg g1
set protocols mpls constituent-list c1 administrative-group green
set protocols mpls constituent-list c2 administrative-group green
set protocols mpls constituent-list c2 administrative-group-extended gold
set protocols mpls constituent-list c3 srlg g2
set protocols mpls constituent-list c3 administrative-group red
set protocols mpls constituent-list c3 administrative-group-extended gold
set protocols mpls constituent-list c4 srlg g3
set protocols mpls constituent-list c4 administrative-group blue
set protocols mpls constituent-list c4 administrative-group-extended gold
set protocols mpls abstract-hop ah1 operator AND
set protocols mpls abstract-hop ah1 constituent-list c1 include-all-list
set protocols mpls abstract-hop ah1 constituent-list c2 include-all-list
set protocols mpls abstract-hop ah2 operator AND
set protocols mpls abstract-hop ah2 constituent-list c3 include-all-list
set protocols mpls abstract-hop ah3 operator AND
set protocols mpls abstract-hop ah3 constituent-list c4 include-all-list
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols mpls interface ge-0/0/0.0 srlg g1
set protocols mpls interface ge-0/0/0.0 administrative-group green
set protocols mpls interface ge-0/0/0.0 administrative-group-extended gold
set protocols mpls interface ge-0/0/2.0 srlg g2
set protocols mpls interface ge-0/0/2.0 administrative-group red
set protocols mpls interface ge-0/0/2.0 administrative-group-extended gold
set protocols mpls interface ge-0/0/1.0 srlg g3
set protocols mpls interface ge-0/0/1.0 administrative-group blue
set protocols mpls interface ge-0/0/1.0 administrative-group-extended gold
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set policy-options policy-statement test then load-balance per-packet
```

Device R1

```
set interfaces ge-0/0/0 unit 0 family inet address 172.16.0.2/16
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 172.21.0.1/16
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 172.20.0.1/16
set interfaces ge-0/0/2 unit 0 family mpls
```

```
set interfaces ge-0/0/3 unit 0 family inet address 172.19.0.1/16
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 127.0.0.1/8
set routing-options srlg g1 srlg-value 100
set routing-options srlg g1 srlg-cost 1000
set routing-options srlg g2 srlg-value 200
set routing-options srlg g2 srlg-cost 2000
set routing-options srlg g3 srlg-value 300
set routing-options srlg g3 srlg-cost 3000
set routing-options administrative-groups-extended-range minimum 50000
set routing-options administrative-groups-extended-range maximum 60000
set routing-options administrative-groups-extended gold group-value 50000
set routing-options router-id 127.0.0.1
set protocols rsvp interface fxp0.0 disable
set protocols mpls administrative-groups green 0
set protocols mpls administrative-groups blue 1
set protocols mpls administrative-groups red 2
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols mpls interface ge-0/0/0.0 srlg g1
set protocols mpls interface ge-0/0/0.0 administrative-group green
set protocols mpls interface ge-0/0/0.0 administrative-group-extended gold
set protocols mpls interface ge-0/0/1.0 srlg g1
set protocols mpls interface ge-0/0/1.0 administrative-group green
set protocols mpls interface ge-0/0/1.0 administrative-group-extended gold
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
```

Device R2

```
set interfaces ge-0/0/0 unit 0 family inet address 172.22.0.2/16
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 172.21.0.2/16
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 172.24.0.1/16
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 family inet address 172.25.0.1/16
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 127.0.0.2/8
set routing-options srlg g1 srlg-value 100
set routing-options srlg g1 srlg-cost 1000
```

```
set routing-options srlg g2 srlg-value 200
set routing-options srlg g2 srlg-cost 2000
set routing-options srlg g3 srlg-value 300
set routing-options srlg g3 srlg-cost 3000
set routing-options administrative-groups-extended-range minimum 50000
set routing-options administrative-groups-extended-range maximum 60000
set routing-options administrative-groups-extended gold group-value 50000
set routing-options router-id 127.0.0.2
set protocols rsvp interface all aggregate
set protocols rsvp interface fxp0.0 disable
set protocols mpls administrative-groups green 0
set protocols mpls administrative-groups blue 1
set protocols mpls administrative-groups red 2
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols mpls interface ge-0/0/1.0 srlg g1
set protocols mpls interface ge-0/0/1.0 administrative-group green
set protocols mpls interface ge-0/0/1.0 administrative-group-extended gold
set protocols mpls interface ge-0/0/2.0 srlg g1
set protocols mpls interface ge-0/0/2.0 administrative-group green
set protocols mpls interface ge-0/0/2.0 administrative-group-extended gold
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
```

Device R3

```
set interfaces ge-0/0/0 unit 0 family inet address 172.26.0.2/16
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 172.23.0.2/16
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 172.24.0.2/16
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 family inet address 172.31.0.1/16
set interfaces lo0 unit 0 family inet address 127.0.0.3/8
set routing-options srlg g1 srlg-value 100
set routing-options srlg g1 srlg-cost 1000
set routing-options srlg g2 srlg-value 200
set routing-options srlg g2 srlg-cost 2000
set routing-options srlg g3 srlg-value 300
set routing-options srlg g3 srlg-cost 3000
set routing-options administrative-groups-extended-range minimum 50000
```

```
set routing-options administrative-groups-extended-range maximum 60000
set routing-options administrative-groups-extended gold group-value 50000
set routing-options router-id 127.0.0.3
set routing-options autonomous-system 64496
set protocols rsvp interface all aggregate
set protocols rsvp interface fxp0.0 disable
set protocols mpls administrative-groups green 0
set protocols mpls administrative-groups blue 1
set protocols mpls administrative-groups red 2
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols mpls interface ge-0/0/2.0 srlg g1
set protocols mpls interface ge-0/0/2.0 administrative-group green
set protocols mpls interface ge-0/0/2.0 administrative-group-extended gold
set protocols mpls interface ge-0/0/1.0 srlg g2
set protocols mpls interface ge-0/0/1.0 administrative-group red
set protocols mpls interface ge-0/0/1.0 administrative-group-extended gold
set protocols mpls interface ge-0/0/0.0 srlg g3
set protocols mpls interface ge-0/0/0.0 administrative-group blue
set protocols mpls interface ge-0/0/0.0 administrative-group-extended gold
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
```

Device R4

```
set interfaces ge-0/0/0 unit 0 family inet address 172.22.0.1/16
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 172.23.0.1/16
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 172.17.0.2/16
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 family inet address 172.19.0.2/16
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 127.0.0.4/32
set routing-options srlg g1 srlg-value 100
set routing-options srlg g1 srlg-cost 1000
set routing-options srlg g2 srlg-value 200
set routing-options srlg g2 srlg-cost 2000
set routing-options srlg g3 srlg-value 300
set routing-options srlg g3 srlg-cost 3000
set routing-options administrative-groups-extended-range minimum 50000
```

```
set routing-options administrative-groups-extended-range maximum 60000
set routing-options administrative-groups-extended gold group-value 50000
set routing-options router-id 127.0.0.4
set protocols rsvp interface all aggregate
set protocols rsvp interface fxp0.0 disable
set protocols mpls administrative-groups green 0
set protocols mpls administrative-groups blue 1
set protocols mpls administrative-groups red 2
set protocols mpls icmp-tunneling
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols mpls interface ge-0/0/2.0 srlg g2
set protocols mpls interface ge-0/0/2.0 administrative-group red
set protocols mpls interface ge-0/0/2.0 administrative-group-extended gold
set protocols mpls interface ge-0/0/1.0 srlg g2
set protocols mpls interface ge-0/0/1.0 administrative-group red
set protocols mpls interface ge-0/0/1.0 administrative-group-extended gold
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
```

Device R5

```
set interfaces ge-0/0/0 unit 0 family inet address 172.26.0.1/16
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 172.18.0.2/16
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 172.20.0.2/24
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 family inet address 172.25.0.2/16
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 127.0.0.5/8
set routing-options srlg g1 srlg-value 100
set routing-options srlg g1 srlg-cost 1000
set routing-options srlg g2 srlg-value 200
set routing-options srlg g2 srlg-cost 2000
set routing-options srlg g3 srlg-value 300
set routing-options srlg g3 srlg-cost 3000
set routing-options administrative-groups-extended-range minimum 50000
set routing-options administrative-groups-extended-range maximum 60000
set routing-options administrative-groups-extended gold group-value 50000
set routing-options router-id 127.0.0.5
```

```
set protocols rsvp interface all aggregate
set protocols rsvp interface fxp0.0 disable
set protocols mpls administrative-groups green 0
set protocols mpls administrative-groups blue 1
set protocols mpls administrative-groups red 2
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols mpls interface ge-0/0/1.0 srlg g3
set protocols mpls interface ge-0/0/1.0 administrative-group blue
set protocols mpls interface ge-0/0/1.0 administrative-group-extended gold
set protocols mpls interface ge-0/0/0.0 srlg g3
set protocols mpls interface ge-0/0/0.0 administrative-group blue
set protocols mpls interface ge-0/0/0.0 administrative-group-extended gold
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
```

Procedure

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure Device R0:

1. Enable enhanced IP network services on Device R0.

```
[edit chassis]
user@R0# set network-services ip
```

2. Configure the interfaces on Device R0, including the loopback interface.

```
[edit interfaces]
user@R0# set ge-0/0/0 unit 0 family inet address 172.16.0.1/16
user@R0# set ge-0/0/0 unit 0 family mpls
user@R0# set ge-0/0/1 unit 0 family inet address 172.18.0.1/16
user@R0# set ge-0/0/1 unit 0 family mpls
user@R0# set ge-0/0/2 unit 0 family inet address 172.17.0.1/16
user@R0# set ge-0/0/2 unit 0 family mpls
```

```
user@R0# set ge-0/0/3 unit 0 family inet address 172.30.0.1/16
user@R0# set lo0 unit 0 family inet address 127.0.0.6/8
```

3. Assign the router ID and autonomous system number for Device R0.

```
[edit routing-options]
user@R0# set router-id 127.0.0.6
user@R0# set autonomous-system 64496
```

4. Configure the SRLG definitions.

```
[edit routing-options]
user@R0# set srlg g1 srlg-value 100
user@R0# set srlg g1 srlg-cost 1000
user@R0# set srlg g2 srlg-value 200
user@R0# set srlg g2 srlg-cost 2000
user@R0# set srlg g3 srlg-value 300
user@R0# set srlg g3 srlg-cost 3000
```

5. Configure the extended administrative group definitions.

```
[edit routing-options]
user@R0# set administrative-groups-extended-range minimum 50000
user@R0# set administrative-groups-extended-range maximum 60000
user@R0# set administrative-groups-extended gold group-value 50000
```

6. Configure the administrative group definitions.

```
[edit protocols]
user@R0# set mpls administrative-groups green 0
user@R0# set mpls administrative-groups blue 1
user@R0# set mpls administrative-groups red 2
```

7. Configure MPLS on all the interfaces of Device R0, excluding the management interface.

```
[edit protocols]
user@R0# set mpls interface all
user@R0# set mpls interface fxp0.0 disable
```

8. Assign the interfaces of Device R0 with the configured traffic engineering attributes.

```
[edit protocols]
user@R0# set mpls interface ge-0/0/0.0 srlg g1
user@R0# set mpls interface ge-0/0/0.0 administrative-group green
user@R0# set mpls interface ge-0/0/0.0 administrative-group-extended gold
user@R0# set mpls interface ge-0/0/2.0 srlg g2
user@R0# set mpls interface ge-0/0/2.0 administrative-group red
user@R0# set mpls interface ge-0/0/2.0 administrative-group-extended gold
user@R0# set mpls interface ge-0/0/1.0 srlg g3
user@R0# set mpls interface ge-0/0/1.0 administrative-group blue
user@R0# set mpls interface ge-0/0/1.0 administrative-group-extended gold
```

9. Configure an LSP connecting Device R0 with Device R3, and assign primary and secondary path attributes to the LSP.

```
[edit protocols]
user@R0# set mpls label-switched-path R0-R31 to 127.0.0.3
user@R0# set mpls label-switched-path R0-R31 primary prim
user@R0# set mpls label-switched-path R0-R31 secondary stdby standby
user@R0# set mpls label-switched-path R0-R31 secondary nonstdby
```

10. Define the primary and secondary paths for the R0-R31 LSP.

```
[edit protocols]
user@R0# set mpls path path_primary 172.16.0.2 strict
user@R0# set mpls path path_primary 172.21.0.2 strict
user@R0# set mpls path path_primary 172.24.0.2 strict
user@R0# set mpls path path_ter_nonstdby 172.18.0.1 strict
user@R0# set mpls path path_ter_nonstdby 172.26.0.2 strict
user@R0# set mpls path path_sec_stdby 172.17.0.2 strict
user@R0# set mpls path path_sec_stdby 172.23.0.2 strict
```

11. Create constituent lists with constituent traffic engineering attributes for abstract-hop definitions.

```
[edit protocols]
user@R0# set mpls constituent-list c1 srlg g1
user@R0# set mpls constituent-list c1 administrative-group green
user@R0# set mpls constituent-list c2 administrative-group green
user@R0# set mpls constituent-list c2 administrative-group-extended gold
user@R0# set mpls constituent-list c3 srlg g2
user@R0# set mpls constituent-list c3 administrative-group red
user@R0# set mpls constituent-list c3 administrative-group-extended gold
user@R0# set mpls constituent-list c4 srlg g3
user@R0# set mpls constituent-list c4 administrative-group blue
user@R0# set mpls constituent-list c4 administrative-group-extended gold
```

12. Define abstract hops by assigning the configured constituent lists and respective operators.

```
[edit protocols]
user@R0# set mpls abstract-hop ah1 operator AND
user@R0# set mpls abstract-hop ah1 constituent-list c1 include-all-list
user@R0# set mpls abstract-hop ah1 constituent-list c2 include-all-list
user@R0# set mpls abstract-hop ah2 operator AND
user@R0# set mpls abstract-hop ah2 constituent-list c3 include-all-list
user@R0# set mpls abstract-hop ah3 operator AND
user@R0# set mpls abstract-hop ah3 constituent-list c4 include-all-list
```

13. Define constraints for the configured paths by including abstract hop definitions.

```
[edit protocols]
user@R0# set mpls path prim ah1 abstract
user@R0# set mpls path prim ah1 strict
user@R0# set mpls path stdby ah2 abstract
user@R0# set mpls path stdby ah2 strict
user@R0# set mpls path nonstdby ah3 abstract
user@R0# set mpls path nonstdby ah3 strict
```

14. Configure RSVP on Device R0. Enable RSVP on all the interfaces of Device R0, excluding the management interface and interface connecting to Host1, and assign bandwidth values.

```
[edit protocols]
user@R0# set rsvp interface all aggregate
user@R0# set rsvp interface fxp0.0 disable
user@R0# set rsvp interface ge-0/0/0.0 bandwidth 80m
user@R0# set rsvp interface ge-0/0/2.0 bandwidth 200m
user@R0# set rsvp interface ge-0/0/1.0 bandwidth 500m
```

15. Configure OSPF on all the interfaces of Device R0, excluding the management interface, and assign traffic engineering capabilities.

```
[edit protocols]
user@R0# set ospf traffic-engineering
user@R0# set ospf area 0.0.0.0 interface all
user@R0# set ospf area 0.0.0.0 interface fxp0.0 disable
```

16. Configure a policy on Device R0 to enable load balancing on a per-packet basis.

```
[edit policy-options]
user@R0# set forwarding-table export test
```

17. Export the load-balancing policy to the forwarding table.

```
[edit policy-options]
user@R0# set policy-statement test then load-balance per-packet
```

Results

From configuration mode, confirm your configuration by entering the `show chassis`, `show interfaces`, `show routing-options`, `show protocols`, and `show policy-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R0# show chassis
network-services ip;
```

```
user@R0# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 172.16.0.1/16;
        }
        family mpls;
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 172.18.0.1/16;
        }
        family mpls;
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 172.17.0.1/16;
        }
        family mpls;
    }
}
ge-0/0/3 {
    unit 0 {
        family inet {
            address 172.30.0.1/16;
        }
    }
}
```

```
lo0 {
    unit 0 {
        family inet {
            address 127.0.0.6/8;
        }
    }
}
```

```
user@R0# show routing-options
srlg {
    g1 {
        srlg-value 100;
        srlg-cost 1000;
    }
    g2 {
        srlg-value 200;
        srlg-cost 2000;
    }
    g3 {
        srlg-value 300;
        srlg-cost 3000;
    }
}
administrative-groups-extended-range {
    minimum 50000;
    maximum 60000;
}
administrative-groups-extended {
    gold group-value 50000;
}
```

```
user@R0# show protocols
rsvp {
    interface all;
    interface fxp0.0 {
        disable;
    }
    interface ge-0/0/0.0 {
        bandwidth 80m;
    }
```

```
    interface ge-0/0/2.0 {
        bandwidth 200m;
    }
    interface ge-0/0/1.0 {
        bandwidth 500m;
    }
}
mpls {
    administrative-groups {
        green 0;
        blue 1;
        red 2;
    }
    label-switched-path R0-R31 {
        to 127.0.0.3;
        adaptive;
        auto-bandwidth {
            adjust-interval 300;
            adjust-threshold 5;
            minimum-bandwidth 10m;
            maximum-bandwidth 1g;
        }
        primary prim;
        secondary stdby {
            standby;
        }
        secondary nonstdby;
    }
    path path_primary {
        172.16.0.2 strict;
        172.21.0.2 strict;
        172.24.0.2 strict;
    }
    path path_ter_nonstdby {
        172.18.0.1 strict;
        172.26.0.2 strict;
    }
    path path_sec_stdby {
        172.17.0.2 strict;
        172.23.0.2 strict;
    }
    path prim {
        ah1 abstract strict;
```

```
    }
    path stdby {
        ah2 abstract strict;
    }
    path nonstdby {
        ah3 abstract strict;
    }
    constituent-list c1 {
        srlg g1;
        administrative-group green;
    }
    constituent-list c2 {
        administrative-group green;
        administrative-group-extended gold;
    }
    constituent-list c3 {
        srlg g2;
        administrative-group red;
        administrative-group-extended gold;
    }
    constituent-list c4 {
        srlg g3;
        administrative-group blue;
        administrative-group-extended gold;
    }
    abstract-hop ah1 {
        operator AND;
        constituent-list {
            c1 include-all-list;
            c2 include-all-list;
        }
    }
    abstract-hop ah2 {
        operator AND;
        constituent-list {
            c3 include-all-list;
        }
    }
    abstract-hop ah3 {
        operator AND;
        constituent-list {
            c4 include-all-list;
        }
```

```
    }
    interface all;
    interface fxp0.0 {
        disable;
    }
    interface ge-0/0/0.0 {
        srlg g1;
        administrative-group green;
        administrative-group-extended gold;
    }
    interface ge-0/0/2.0 {
        srlg g2;
        administrative-group red;
        administrative-group-extended gold;
    }
    interface ge-0/0/1.0 {
        srlg g3;
        administrative-group blue;
        administrative-group-extended gold;
    }
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
}
```

```
user@R0# show policy-options
policy-statement test {
    then {
        load-balance per-packet;
    }
}
```

Verification

IN THIS SECTION

- Verifying Abstract Hop Configuration | 565
- Verifying Abstract Hop Path Computation | 566

Confirm that the configuration is working properly.

Verifying Abstract Hop Configuration

Purpose

Verify the members of the abstract hop definition on Device R0 by issuing the `show mpls abstract-hop-membership` command, which displays the abstract hop membership tables.

Action

From operational mode, run the **show mpls abstract-hop-membership** command.

```
user@R0> show mpls abstract-hop-membership
Abstract hop: ah1
   Credibility: 0
Address: 127.0.0.6
Address: 127.0.0.1
Address: 127.0.0.2
Address: 127.0.0.3

Abstract hop: ah2
   Credibility: 0
Address: 127.0.0.6
Address: 127.0.0.3
Address: 127.0.0.4

Abstract hop: ah3
   Credibility: 0
Address: 127.0.0.6
```

```
Address: 127.0.0.3
Address: 127.0.0.5
```

Meaning

The `show mpls abstract-hop-membership` command output provides the abstract hop to traffic engineering database node mapping. The `Credibility` field displays the credibility value associated with the interior gateway protocol in use (OSPF).

Verifying Abstract Hop Path Computation

Purpose

Verify the abstract computation preprocessing for LSPs on Device R0 by issuing the `show mpls lsp abstract-computation` command.

Action

From operational mode, run the **show mpls lsp abstract-computation** command.

```
user@R0> show mpls lsp abstract-computation
Path computation using abstract hops for LSP: R0-R31
   Path type: Primary, Path name: prim

   Credibility: 0, Total no of CSPF passes: 2
     CSPF pass no: 0
      Start address of the pass: 127.0.0.6
       Destination: 127.0.0.1, State: VALID
       Destination: 127.0.0.2, State: VALID
       Destination: 127.0.0.3, State: VALID
       Affinity: ah1
     CSPF pass no: 1
      Start address of the pass: 127.0.0.1
       Destination: 127.0.0.3, State: VALID
   Path type: Secondary, Path name: nonstdby
   Path type: Standby, Path name: stdby

   Credibility: 0, Total no of CSPF passes: 2
     CSPF pass no: 0
      Start address of the pass: 127.0.0.6
       Destination: 127.0.0.3, State: VALID
```

```
        Destination: 127.0.0.4, State: VALID
        Affinity: ah2
      CSPF pass no: 1
       Start address of the pass: 127.0.0.4
        Destination: 127.0.0.3, State: VALID
```

Meaning

The `show mpls lsp abstract-hop-computation` command output provides the various computation passes involved per LSP, and the qualifying exit devces for each pass. The command output also gives the affinity per pass, and shows the current start device chosen for the pass. For each viable router (device), the state of backtracking is displayed, where it can either be valid or disqualified.

The `Credibility` field indicates the credibility value associated with the interior gateway protocol in use (OSPF).

Configuring the Maximum Number of MPLS Labels

For interfaces that you configure for MPLS applications, you can set the maximum number of labels upon which MPLS can operate.

By default, the maximum number of labels is three. You can change the maximum to four labels or five labels for applications that require four or five labels.

Starting in Junos OS Release 19.1R1, the maximum number of labels that can be pushed by the egress Packet Forwarding Engine (PFE) can be leveraged, wherein the number of labels that can be pushed for an MPLS next hop is the number of labels the device is capable of pushing, or the *maximum-labels* configured under `family mpls` of the outgoing interface, whichever is smaller. This support is enabled on MX Series routers with MPC and MIC interfaces, and PTX Series routers with third-generation FPCs.

NOTE: On PTX Series routers, the maximum number of labels that can be pushed by the ingress PFE is 4 and egress PFE is 8.

The increased label push capability is useful for features, such as segment routing traffic-engineering LSPs and RSVP-TE pop-and-forward LSPs. All existing functionality of applications using MPLS next hops continue to work with the increased label push capability. This includes:

- All OAM utilities, such as lsping, traceroute, and BFD for MPLS LSPs.
- Monitoring utilities, such as lspmon, and LM DM for MPLS LSPs.

The `show route table` and `show route forwarding-table` command outputs are enhanced to display up to 16 labels per next hop component.

For example:

```
user@host> show route table inet.3

inet.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

11.0.0.17/32       *[SPRING-TE/8] 00:02:16, metric 1
                    > to 192.1.2.2 via ge-0/0/2.0, Push 1000115, Push 1000114, Push 1000113,
Push 1000112, Push 1000111, Push 1000110, Push 1000109, Push 1000108, Push 1000107, Push
1000106, Push 1000105, Push 1000104, Push 1000103, Push 1000102, Push 1000101(top)
                      to 192.1.3.2 via ge-0/0/4.0, Push 1000115, Push 1000114, Push 1000113,
Push 1000112, Push 1000111, Push 1000110, Push 1000109, Push 1000108, Push 1000107, Push
1000106, Push 1000105, Push 1000104, Push 1000103, Push 1000102, Push 1000101(top)
```

NOTE: When the maximum number of MPLS labels of an interface is modified, the MPLS interface is bounced. All LDP and RSVP sessions on that interface are restarted, resulting in all LSPs over that interface to flap.

For example, suppose you configure a two-tier carrier-of-carriers VPN service for customers who provide VPN service. A carrier-of-carrier VPN is a two-tiered relationship between a provider carrier (Tier 1 ISP) and a customer carrier (Tier 2 ISP). In a carrier-of-carrier VPN, the provider carrier provides a VPN backbone network for the customer carrier. The customer carrier in turn provides Layer 3 VPN service to its end customers. The customer carrier sends labeled traffic to the provider carrier to deliver it to the next hop on the other side of the provider carrier's network. This scenario requires a three-label stack: one label for the provider carrier VPN, another label for the customer carrier VPN, and a third label for the transport route.

If you add fast reroute service, the PE routers in the provider carrier's network must be configured to support a fourth label (the reroute label). If the customer carrier is using LDP as its signaling protocol and the provider carrier is using RSVP, the provider carrier must support LDP over RSVP tunnel service. This additional service requires an additional label, for a total of five labels.

To the customer carrier, the router it uses to connect to the provider carrier's VPN is a PE router. However, the provider carrier views this device as a CE router.

Table 14 on page 569 summarizes the label requirements.

Table 14: Sample Scenarios for Using 3, 4, or 5 MPLS Labels

Number of Labels Required	Scenarios
3	Carrier-of-carriers VPN or a VPN with two labels and fast reroute
4	Combination of carrier-of-carriers and fast reroute
5	Carrier-of-carriers with fast reroute and the customer carrier running LDP, with the provider carrier running RSVP

To configure and monitor the maximum number of labels:

1. Specify the maximum on the logical interface. Apply this configuration to the carrier's PE routers.

```
[edit interfaces ge-0/1/3 unit 0 family mpls]
user@switch# set maximum-labels maximum-limit
```

2. Verify the configuration.

```
[edit system]
user@switch# show interfaces ge-0/1/3.0
Logical interface ge-0/1/3.0 (Index 77) (SNMP ifIndex 507)
    Flags: SNMP-Traps Encapsulation: ENET2
    Input packets : 0
    Output packets: 0
    Protocol mpls, MTU: 1480, Maximum labels: 8
      Flags: Is-Primary
```

The command output includes the `Maximum labels: 5` field under the logical interface unit 0.

Configuring MPLS to Pop the Label on the Ultimate-Hop Router

You can control the label value advertised on the egress router of a label-switched path (LSP). The default advertised label is label 3 (Implicit Null Label). If label 3 is advertised, the penultimate-hop router removes the label and sends the packet to the egress router. By enabling ultimate-hop popping, label 0 (IPv4 Explicit Null Label) is advertised. Ultimate-hop popping ensures that any packets traversing an MPLS network include a label.

NOTE: Juniper Networks routers queue packets based on the incoming label. Routers from other vendors might queue packets differently. Keep this in mind when working with networks containing routers from multiple vendors.

To configure MPLS to pop the label on the ultimate-hop router, include the `explicit-null` statement:

```
explicit-null;
```

You can configure this statement at the following hierarchy levels:

- `[edit protocols mpls]`
- `[edit logical-systems` *`logical-system-name`* `protocols mpls]`

Advertising Explicit Null Labels to BGP Peers

For the IPv4 (`inet`) family only, BGP peers in a routing group can send an explicit NULL label for a set of connected routes (direct and loopback routes) for the inet labeled-unicast and inet6 labeled-unicast NLRI. By default, peers advertise label 3 (implicit NULL). If the `explicit-null` statement is enabled, peers advertise label 0 (explicit NULL). The explicit NULL labels ensures that labels are always present on packets traversing an MPLS network. If the implicit NULL label is used. the penultimate hop router removes the label and sends the packet as a plain IP packet to the egress router. This might cause issues in queuing the packet properly on the penultimate hop router if the penultimate hop is another vendor's router. Some other vendors queue packets based on the CoS bits in the outgoing label rather than the incoming label.

To advertise an explicit null label, include the following statements in the configuration:

```
family inet {
    labeled-unicast {
        aggregate-label {
            community community-name:
        }
        explicit-null {
            connected-only;
        }
    }
}
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

The `connected-only` statement is required to advertise explicit null labels.

To verify that the explicit NULL label is being advertised for connected routes, use the `show route advertising-protocol bgp` *`neighbor-address`* command.

Understanding MPLS Label Operations on EX Series Switches

IN THIS SECTION

- MPLS Label-Switched Paths and MPLS Labels on the Switches | **571**
- Reserved Labels | **572**
- MPLS Label Operations on the Switches | **573**
- Penultimate-Hop Popping and Ultimate-Hop Popping | **574**

In the traditional packet-forwarding paradigm, as a packet travels from one switch to the next, an independent forwarding decision is made at each hop. The IP network header is analyzed and the next hop is chosen based on this analysis and on the information in the routing table. In an MPLS environment, the analysis of the packet header is made only once, when a packet enters the MPLS tunnel (that is, the path used for MPLS traffic).

When an IP packet enters a label-switched path (LSP), the ingress provider edge (PE) switch examines the packet and assigns it a label based on its destination, placing the label in the packet's header. The label transforms the packet from one that is forwarded based on its IP routing information to one that is forwarded based on information associated with the label. The packet is then forwarded to the next provider switch in the LSP. This switch and all subsequent switches in the LSP do not examine any of the IP routing information in the labeled packet. Rather, they use the label to look up information in their label forwarding table. They then replace the old label with a new label and forward the packet to the next switch in the path. When the packet reaches the egress PE switch, the label is removed, and the packet again becomes a native IP packet and is again forwarded based on its IP routing information.

This topic describes:

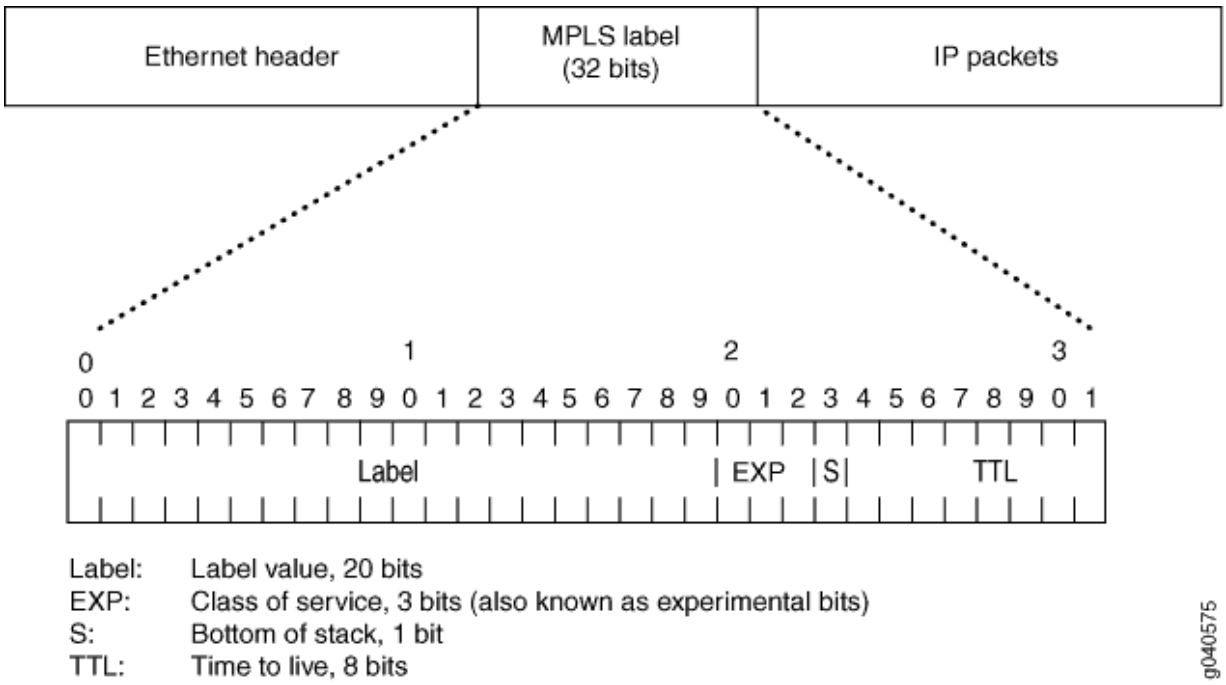
MPLS Label-Switched Paths and MPLS Labels on the Switches

When a packet enters the MPLS network, it is assigned to an LSP. Each LSP is identified by a label, which is a short (20-bit), fixed-length value at the front of the MPLS label (32 bits). Labels are used as lookup indexes for the label forwarding table. For each label, this table stores forwarding information. Because no additional parsing or lookup is done on the encapsulated packet, MPLS supports the transmission of any other protocols within the packet payload.

NOTE: The implementation of MPLS on Juniper Networks EX3200 and EX4200 Ethernet Switches supports only single-label packets. However, MPLS on Juniper Networks EX8200 Ethernet Switches supports packets with as many as three labels.

Figure 33 on page 572 shows the encoding of a single label. The encoding appears after data link layer headers, but before any network layer header.

Figure 33: Label Encoding



Reserved Labels

Labels range from 0 through 1,048,575. Labels 0 through 999,999 are for internal use.

Some of the reserved labels (in the range 0 through 15) have well-defined meanings. The following reserved labels are used by the switches:

- 0, IPv4 Explicit Null label—This value is valid only when it is the sole label entry (no label stacking). It indicates that the label must be popped on receipt. Forwarding continues based on the IP version 4 (IPv4) packet.
- 1, Router Alert label—When a packet is received with a top label value of 1, it is delivered to the local software module for processing.

- 2, IPv6 Explicit Null label—This value is legal only when it is the sole label entry (no label stacking). It indicates that the label must be popped on receipt.
- 3, Implicit Null label—This label is used in the signaling protocol (RSVP) only to request label popping by the downstream switch. It never actually appears in the encapsulation. Labels with a value of 3 must not be used in the data packet as real labels. No payload type (IPv4 or IPv6) is implied with this label.

MPLS Label Operations on the Switches

EX Series switches support the following label operations:

- Push
- Pop
- Swap

The push operation affixes a new label to the top of the IP packet. For IPv4 packets, the new label is the first label. The time to live (TTL) field value in the packet header is derived from the IP packet header. The push operation cannot be applied to a packet that already has an MPLS label.

The pop operation removes a label from the beginning of the packet. Once the label is removed, the TTL is copied from the label into the IP packet header, and the underlying IP packet is forwarded as a native IP packet

The swap operation removes an existing MPLS label from an IP packet and replaces it with a new MPLS label, based on the following:

- Incoming interface
- Label
- Label forwarding table

Figure 34 on page 574 shows an IP packet without a label arriving on the customer edge interface (**ge-0/0/1**) of the ingress PE switch. The ingress PE switch examines the packet and identifies that packet's destination as the egress PE switch. The ingress PE switch applies label 100 to the packet and sends the MPLS packet to its outgoing MPLS core interface (**ge-0/0/5**). The MPLS packet is transmitted on the MPLS tunnel through the provider switch, where it arrives at interface **ge-0/0/5** with label 100. The provider switch swaps label 100 to label 200 and forwards the MPLS packet through its core interface (**ge-0/0/7**) to the next hop on the tunnel, which is the egress PE switch. The egress PE switch receives the MPLS packet through its core interface (**ge-0/0/7**), removes the MPLS label, and sends the IP packet out of its customer edge interface (**ge-0/0/1**) to a destination that is beyond the tunnel.

Figure 34: MPLS Label Swapping

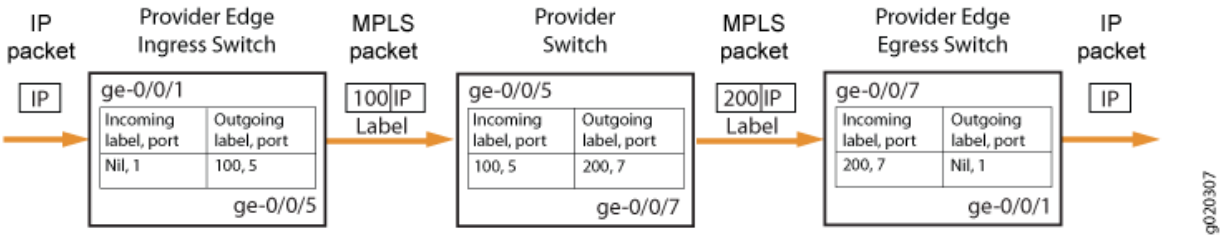


Figure 34 on page 574 shows the path of a packet as it passes in one direction from the ingress PE switch to the egress PE switch. However, the MPLS configuration also allows traffic to travel in the reverse direction. Thus, each PE switch operates as both an ingress switch and an egress switch.

Penultimate-Hop Popping and Ultimate-Hop Popping

The switches enable penultimate-hop popping (PHP) by default with IP over MPLS configurations. With PHP, the penultimate provider switch is responsible for popping the MPLS label and forwarding the traffic to the egress PE switch. The egress PE switch then performs an IP route lookup and forwards the traffic. This reduces the processing load on the egress PE switch, because it is not responsible for popping the MPLS label.

On EX8200 switches, you can choose to use either the default, PHP, or to configure ultimate-hop popping.

- The default advertised label is label 3 (Implicit Null label). If label 3 is advertised, the penultimate-hop switch removes the label and sends the packet to the egress PE switch.
- If ultimate-hop popping is enabled, label 0 (IPv4 Explicit Null label) is advertised and the egress PE switch of the LSP removes the label.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
19.1R1	Starting in Junos OS Release 19.1R1, the maximum number of labels that can be pushed by the egress Packet Forwarding Engine (PFE) can be leveraged, wherein the number of labels that can be pushed for an MPLS next hop is the number of labels the device is capable of pushing, or the *maximum-labels* configured under `family mpls` of the outgoing interface, whichever is smaller. This support is enabled on MX Series routers with MPC and MIC interfaces, and PTX Series routers with third-generation FPCs.
14.2	Starting with Junos OS Release 14.2, entropy label is supported in mixed mode chassis where the entropy label can be configured without enhanced-ip configuration.

RELATED DOCUMENTATION

MPLS Overview | 2

LSP Routes

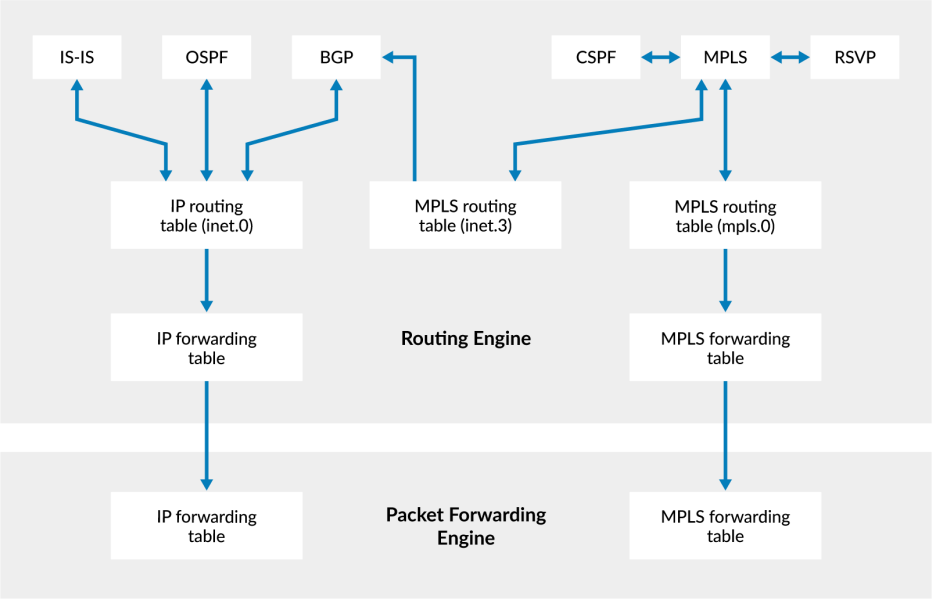
IN THIS SECTION

- MPLS and Routing Tables | 575
- Fast Reroute Overview | 577
- Configuring Fast Reroute | 580
- Detour Merging Process | 581
- Detour Computations | 582
- Fast Reroute Path Optimization | 582
- Configuring the Optimization Interval for Fast Reroute Paths | 583
- Adding LSP-Related Routes to the inet.3 or inet6.3 Routing Table | 583

MPLS and Routing Tables

The IGPs and BGP store their routing information in the inet.0 routing table, the main IP routing table. If the `traffic-engineering bgp` command is configured, thereby allowing only BGP to use MPLS paths for forwarding traffic, MPLS path information is stored in a separate routing table, inet.3. Only BGP accesses the inet.3 routing table. BGP uses both inet.0 and inet.3 to resolve next-hop addresses. If the `traffic-engineering bgp-igp` command is configured, thereby allowing the IGPs to use MPLS paths for forwarding traffic, MPLS path information is stored in the inet.0 routing table. (Figure 35 on page 576 and Figure 36 on page 577 illustrate the routing tables in the two traffic engineering configurations.)

Figure 35: Routing and Forwarding Tables, traffic-engineering bgp



The inet.3 routing table contains the host address of each LSP's egress router. This routing table is used on ingress routers to route packets to the destination egress router. BGP uses the inet.3 routing table on the ingress router to help in resolving next-hop addresses.

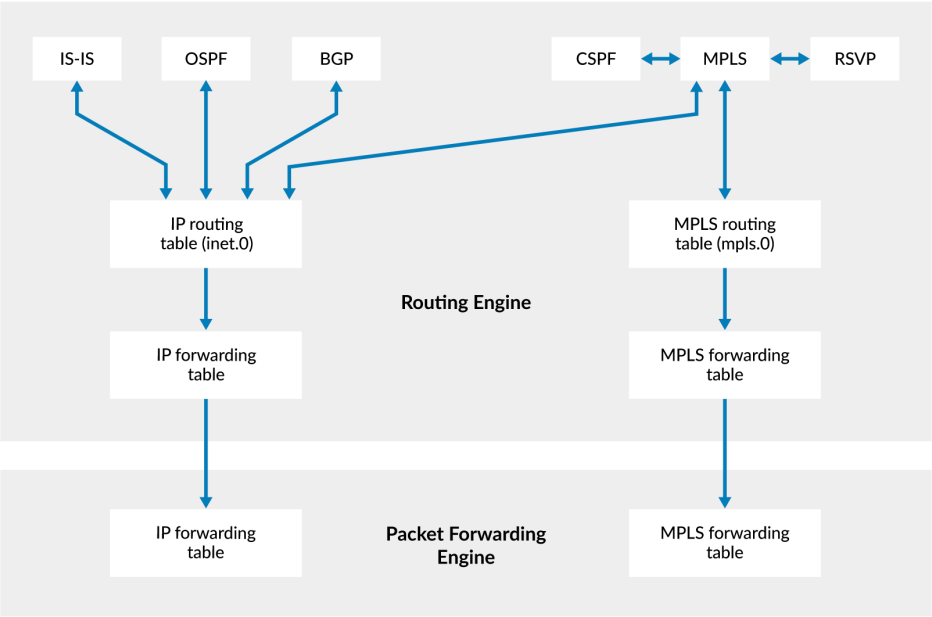
MPLS also maintains an MPLS path routing table (mpls.0), which contains a list of the next label-switched router in each LSP. This routing table is used on transit routers to route packets to the next router along an LSP.

Typically, the egress router in an LSP does not consult the mpls.0 routing table. (This router does not need to consult mpls.0 because the penultimate router in the LSP either changes the packet's label to a value of 0 or pops the label.) In either case, the egress router forwards it as an IPv4 packet, consulting the IP routing table, inet.0, to determine how to forward the packet.

When a transit or egress router receives an MPLS packet, information in the MPLS forwarding table is used to determine the next transit router in the LSP or to determine that this router is the egress router.

When BGP resolves a next-hop prefix, it examines both the inet.0 and inet.3 routing tables, seeking the next hop with the lowest preference. If it finds a next-hop entry with an equal preference in both routing tables, BGP prefers the entry in the inet.3 routing table.

Figure 36: Routing and Forwarding Tables, traffic-engineering bgp-igp



Generally, BGP selects next-hop entries in the inet.3 routing table because their preferences are always lower than OSPF and IS-IS next-hop preferences. When you configure LSPs, you can override the default preference for MPLS LSPs, which might alter the next-hop selection process.

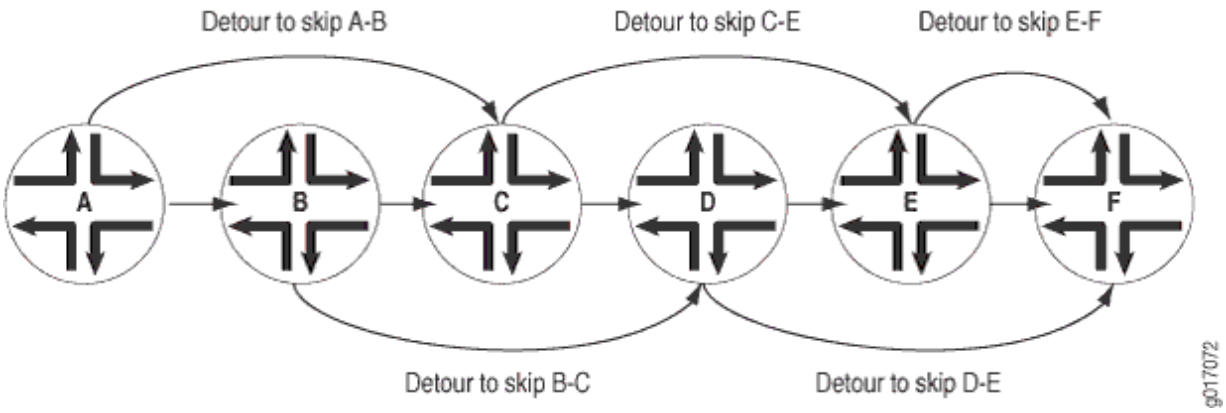
When BGP selects a next-hop entry from the inet.3 routing table, it installs that LSP into the forwarding table in the Packet Forwarding Engine, which causes packets destined for that next hop to enter and travel along the LSP. If the LSP is removed or fails, the path is removed from the inet.3 routing table and from the forwarding table, and BGP reverts to using a next hop from the inet.0 routing table.

Fast Reroute Overview

Fast reroute provides redundancy for an LSP path. When you enable fast reroute, detours are precomputed and preestablished along the LSP. In case of a network failure on the current LSP path, traffic is quickly routed to one of the detours. Figure 37 on page 578 illustrates an LSP from Router A to Router F, showing the established detours. Each detour is established by an upstream node to avoid the link toward the immediate downstream node and the immediate downstream node itself. Each detour might traverse through one or more label-switched routers (or switches) that are not shown in the figure.

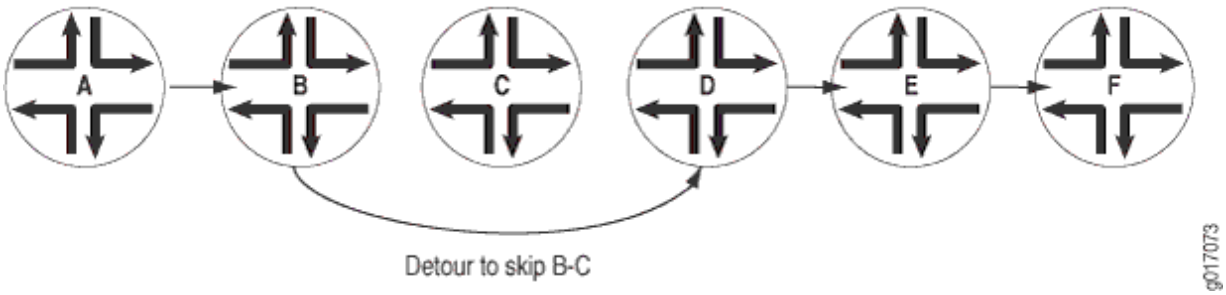
Fast reroute protects traffic against any single point of failure between the ingress and egress routers (or switches). If there is a failure in a scaled fast reroute scenario, the devices lose reachability to all the peers that were connected through the failed link. This leads to traffic interruption, as the BGP session among the devices goes down. If there are multiple failures along an LSP, fast reroute itself might fail. Also, fast reroute does not protect against failure of the ingress or egress routers.

Figure 37: Detours Established for an LSP Using Fast Reroute



If a node detects that a downstream link has failed (using a link-layer-specific liveness detection mechanism) or that a downstream node has failed (for example, using the RSVP neighbor hello protocol), the node quickly switches the traffic to the detour and, at the same time, signals the ingress router about the link or node failure. Figure 38 on page 578 illustrates the detour taken when the link between Router B and Router C fails.

Figure 38: Detour After the Link from Router B to Router C Fails



If the network topology is not rich enough (there are not enough routers with sufficient links to other routers), some of the detours might not succeed. For example, the detour from Router A to Router C in Figure 37 on page 578 cannot traverse link A-B and Router B. If such a path is not possible, the detour does not occur.

Note that after the node switches traffic to the detour, it might switch the traffic again to a newly calculated detour soon after. This is because the initial detour route might not be the best route. To make rerouting as fast as possible, the node switches traffic onto the initial detour without first verifying that the detour is valid. Once the switch is made, the node recomputes the detour. If the node determines that the initial detour is still valid, traffic continues to flow over this detour. If the node determines that the initial detour is no longer valid, it again switches the traffic to a newly computed detour.

NOTE: If you issue `show` commands after the node has switched traffic to the initial detour, the node might indicate that the traffic is still flowing over the original LSP. This situation is temporary and should correct itself quickly.

The time required for a fast-rerouting detour to take effect depends on two independent time intervals:

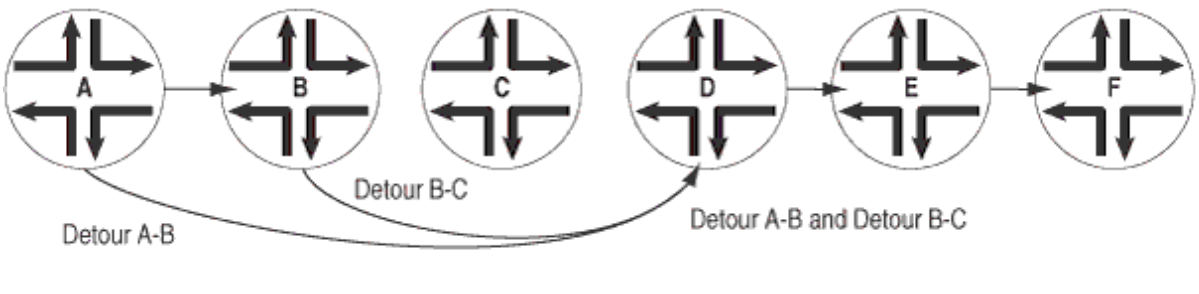
- Amount of time to detect that there is a link or node failure—This interval depends greatly on the link layer in use and the nature of the failure. For example, failure detection on an SONET/SDH link typically is much faster than on a Gigabit Ethernet link, and both are much faster than detection of a router failure.
- Amount of time required to splice the traffic onto the detour—This operation is performed by the Packet Forwarding Engine, which requires little time to splice traffic onto the detour. The time needed can vary depending on the number of LSPs being switched to detours.

Fast reroute is a short-term patch to reduce packet loss. Because detour computation might not reserve adequate bandwidth, the detours might introduce congestion on the alternate links. The ingress router is the only router that is fully aware of LSP policy constraints and, therefore, is the only router able to come up with adequate long-term alternate paths.

Detours are created by use of RSVP and, like all RSVP sessions, they require extra state and overhead in the network. For this reason, each node establishes at most one detour for each LSP that has fast reroute enabled. Creating more than one detour for each LSP increases the overhead, but serves no practical purpose.

To reduce network overhead further, each detour attempts to merge back into the LSP as soon as possible after the failed node or link. If you can consider an LSP that travels through *n* router nodes, it is possible to create *n* – 1 detours. For instance, in Figure 39 on page 579, the detour tries to merge back into the LSP at Router D instead of at Router E or Router F. Merging back into the LSP makes the detour scalability problem more manageable. If topology limitations prevent the detour from quickly merging back into the LSP, detours merge with other detours automatically.

Figure 39: Detours Merging into Other Detours



Configuring Fast Reroute

Fast reroute provides a mechanism for automatically rerouting traffic on an LSP if a node or link in an LSP fails, thus reducing the loss of packets traveling over the LSP.

To configure fast reroute on an LSP, include the `fast-reroute` statement on the ingress router (or switch):

```
fast-reroute {
    (bandwidth bps | bandwidth-percent percentage);
    (exclude [ group-names ] | no-exclude );
    hop-limit number;
    (include-all [ group-names ] | no-include-all);
    (include-any [ group-names ] | no-include-any);
}
```

You can include this statement at the following hierarchy levels:

- `[edit protocols mpls label-switched-path lsp-name]`
- `[edit logical-systems logical-system-name protocols mpls label-switched-path lsp-name]`

You do not need to configure fast reroute on the LSP's transit and egress routers (or switches). Once fast reroute is enabled, the ingress router (or switch) signals all the downstream routers (or switches) that fast reroute is enabled on the LSP, and each downstream router does its best to set up detours for the LSP. If a downstream router does not support fast reroute, it ignores the request to set up detours and continues to support the LSP. A router that does not support fast reroute will cause some of the detours to fail, but otherwise has no impact on the LSP.

NOTE: To enable PFE fast reroute, configure a routing policy statement with the `load-balance per-packet` statement at the `[edit policy-options policy-statement policy-name then]` hierarchy level on each of the routers where traffic might be rerouted. See also "Configuring Load Balancing Across RSVP LSPs" on page 1265.

By default, no bandwidth is reserved for the rerouted path. To allocate bandwidth for the rerouted path, include either the `bandwidth` statement or the `bandwidth-percent` statement. You can only include one of these statements at a time. If you do not include either the `bandwidth` statement or the `bandwidth-percent` statement, the default setting is to not reserve bandwidth for the detour path.

When you include the `bandwidth` statement, you can specify the specific amount of bandwidth (in bits per second [bps]) you want to reserve for the detour path. The bandwidth does not need to be identical to that allocated for the LSP.

When you specify a bandwidth percent using the `bandwidth-percent` statement, the detour path bandwidth is computed by multiplying the bandwidth percentage by the bandwidth configured for the main traffic-

engineered LSP. For information about how to configure the bandwidth for a traffic-engineered LSP, see "Configuring Traffic-Engineered LSPs" on page 1837.

Hop-limit constraints define how many more routers a detour is allowed to traverse compared with the LSP itself. By default, the hop limit is set to 6. For example, if an LSP traverses 4 routers, any detour for the LSP can be up to 10 (that is, 4 + 6) router hops, including the ingress and egress routers.

By default, a detour inherits the same administrative (coloring) group constraints as its parent LSP when CSPF is determining the alternate path. Administrative groups, also known as link coloring or resource class, are manually assigned attributes that describe the “color” of links, such that links with the same color conceptually belong to the same class. If you specify the `include-any` statement when configuring the parent LSP, all links traversed by the alternate session must have at least one color found in the list of groups. If you specify the `include-all` statement when configuring the parent LSP, all links traversed by the alternate session must have all of the colors found in the list of groups. If you specify the `exclude` statement when configuring the parent LSP, none of the links must have a color found in the list of groups. For more information about administrative group constraints, see "Configuring Administrative Groups for LSPs" on page 613.

Detour Merging Process

This section describes the process used by a router to determine which LSP to select when the router receives path messages from different interfaces with identical Session and Sender Template objects. When this occurs, the router needs to merge the path states.

The router employs the following process to determine when and how to merge path states:

- When all the path messages do not include a fast reroute or a detour object, or when the router is the egress of the LSP, no merging is required. The messages are processed according to RSVP traffic engineering.
- Otherwise, the router *must* record the path state in addition to the incoming interface. If the path messages do not share the same outgoing interface and next-hop router, the router considers them to be independent LSPs and does not merge them.
- For all the path messages that share the same outgoing interface and next-hop router, the router uses the following process to select the final LSP:
  - If only one LSP originates from this node, select it as the final LSP.
  - If only one LSP contains a fast reroute object, select it as the final LSP.
  - If there are several LSPs and some of them have a detour object, eliminate those containing a detour object from the final LSP selection process.
  - If several final LSP candidates remain (that is, there are still both detour and protected LSPs), select the LSPs with fast reroute objects.

- If none of the LSPs have fast reroute objects, select the ones without detour objects. If all the LSPs have detour objects, select them all.
- Of the remaining LSP candidates, eliminate from consideration those that traverse nodes that other LSPs avoid.
- If several candidate LSPs still remain, select the one with the shortest explicit route object (ERO) path length. If more than one LSP has the same path length, select one randomly.

- Once the final LSP has been identified, the router must transmit only the path messages that correspond to this LSP. All other LSPs are considered merged at this node.

Detour Computations

Computing and setting up detours is done independently at each node. On a node, if an LSP has fast reroute enabled and if a downstream link or node can be identified, the router performs a Constrained Shortest Path First (CSPF) computation using the information in the local traffic engineering database. For this reason, detours rely on your IGP supporting traffic engineering extensions. Without the traffic engineering database, detours cannot be established.

CSPF initially attempts to find a path that skips the next downstream node. Attempting to find this path provides protection against downstream failures in either nodes or links. If a node-skipping path is not available, CSPF attempts to find a path on an alternate link to the next downstream node. Attempting to find an alternate link provides protection against downstream failures in links only. Detour computations might not succeed the first time. If a computation fails, the router recomputes detours approximately once every refresh interval until the computation succeeds. The RSVP metric for each detour is set to a value in the range from 10,000 through 19,999.

Fast Reroute Path Optimization

A fast reroute protection path is nondeterministic. The actual protection path of a particular node depends on the history of the LSP and the network topology when the fast reroute path was computed. The lack of deterministic behavior can lead to operational difficulties and poorly optimized paths after multiple link flaps in a network. Even in a small network, after a few link flaps fast reroute paths can traverse an arbitrarily large number of nodes and can remain in that state indefinitely. This is inefficient and makes the network less predictable.

Fast reroute optimization addresses this deficiency. It provides a global path optimization timer, allowing you to optimize all LSPs that have fast reroute enabled and a detour path up and running. The timer value can be varied depending on the expected RE processing load.

The fast reroute optimization algorithm is based on the IGP metric only. As long as the new path's IGP metric is lower than the old path's, the CSPF result is accepted, even if the new path might be more congested (higher bandwidth utilization) or traverses more hops.

In conformance with RFC 4090, *Fast Reroute Extensions to RSVP-TE for LSP Tunnels*, when a new path is computed and accepted for fast reroute optimization, the existing detour is destroyed first and then the new detour is established. To prevent traffic loss, detours actively protecting traffic are not optimized.

Configuring the Optimization Interval for Fast Reroute Paths

You can enable path optimization for fast reroute by configuring the fast reroute optimize timer. The optimize timer triggers a periodic optimization process that recomputes the fast reroute detour LSPs to use network resources more efficiently.

To enable fast reroute path optimization, specify the number of seconds using the optimize-timer option for the `fast-reroute` statement:

```
fast-reroute seconds;
```

You can include this statement at the following hierarchy levels:

- `[edit protocols rsvp]`
- `[edit logical-systems logical-system-name protocols rsvp]`

Adding LSP-Related Routes to the inet.3 or inet6.3 Routing Table

By default, a host route toward the egress router is installed in the inet.3 or inet6.3 routing table. (The host route address is the one you configure in the `to` statement.) Installing the host route allows BGP to perform next-hop resolution. It also prevents the host route from interfering with prefixes learned from dynamic routing protocols and stored in the inet.0 or inet6.0 routing table.

Unlike the routes in the inet.0 or inet6.0 table, routes in the inet.3 or inet6.3 table are not copied to the Packet Forwarding Engine, and hence they cause no changes in the system forwarding table directly. You cannot use the `ping` or `traceroute` command through these routes. The only use for inet.3 or inet6.3 is to permit BGP to perform next-hop resolution. To examine the inet.3 or inet6.3 table, use the `show route table inet.3` or `show route table inet6.3` command.

To inject additional routes into the inet.3 or inet6.3 routing table, include the `install` statement:

```
install {
    destination-prefix <active>;
}
```

You can include this statement at the following hierarchy levels:

- `[edit protocols mpls label-switched-path lsp-name]`

- [edit protocols mpls static-label-switched-path *lsp-name*]
- [edit logical-systems *logical-system-name* protocols mpls label-switched-path *lsp-name*]
- [edit logical-systems *logical-system-name* protocols mpls static-label-switched-path *lsp-name*]

The specified routes are installed as aliases into the routing table when the LSP is established. Installing additional routes allows BGP to resolve next hops within the specified prefix and to direct additional traffic for these next hops to a particular LSP.

Including the active option with the install statement installs the specified prefix into the inet.0 or inet6.0 routing table, which is the primary forwarding table. The result is a route that is installed in the forwarding table any time the LSP is established, which means you can ping or trace the route. Use this option with care, because this type of prefix is very similar to a static route.

You use alias routes for routers that have multiple addresses being used as BGP next hops, or for routers that are not MPLS capable. In either of these cases, the LSP can be configured to another MPLS capable system within the local domain, which then acts as a "border" router. The LSP then terminates on the border router and, from that router, Layer 3 forwarding takes the packet to the true next-hop router.

In the case of an interconnect, the domain's border router can act as the proxy router and can advertise the prefix for the interconnect if the border router is not setting the BGP next hop to itself.

In the case of a point of presence (POP) that has routers that do not support MPLS, one router (for example, a core router) that supports MPLS can act as a proxy for the entire POP and can inject a set of prefixes that cover the POP. Thus, all routers within the POP can advertise themselves as interior BGP (IBGP) next hops, and traffic can follow the LSP to reach the core router. This means that normal IGP routing would prevail within the POP.

You cannot use the ping or traceroute commands on routes in the inet.3 or inet6.3 routing table.

For BGP next-hop resolution, it makes no difference whether a route is in inet.0/inet6.0 or inet.3/inet6.3; the route with the best match (longest mask) is chosen. Among multiple best-match routes, the one with the highest preference value is chosen.

NOTE: The install *destination-prefix* active statement is not supported on static LSPs. When the install *destination-prefix* active statement is configured for a static LSP, the MPLS routes do not get installed into the inet.0 routing table.

RELATED DOCUMENTATION

MPLS Overview | 2

LSP Computation

IN THIS SECTION

- Constrained-Path LSP Computation | 585
- How CSPF Selects a Path | 586
- CSPF Path Selection Tie-Breaking | 587
- Computing CSPF Paths Offline | 588
- Configuring CSPF Tie Breaking | 588
- Disabling Constrained-Path LSP Computation | 589

Constrained-Path LSP Computation

The Constrained Shortest Path First (CSPF) algorithm is an advanced form of the shortest-path-first (SPF) algorithm used in OSPF and IS-IS route computations. CSPF is used in computing paths for LSPs that are subject to multiple constraints. When computing paths for LSPs, CSPF considers not only the topology of the network, but also the attributes of the LSP and the links, and it attempts to minimize congestion by intelligently balancing the network load.

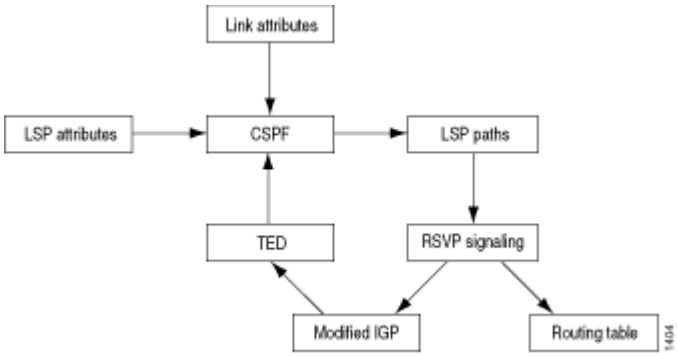
The constraints that CSPF considers include:

- LSP attributes
  - Administrative groups (that is, link color requirements)
  - Bandwidth requirements
  - Explicit route (strict or loose)
  - Hop limitations
  - Priority (setup and hold)
- Link attributes
  - Administrative groups (that is, link colors assigned to the link)
  - Reservable bandwidth of the links (static bandwidth minus the currently reserved bandwidth)

The data that CSPF considers comes from the following sources:

- Traffic engineering database—Provides CSPF with up-to-date topology information, the current reservable bandwidth of links, and the link colors. For the CSPF algorithm to perform its computations, a link-state IGP (such as OSPF or IS-IS) with special extensions is needed. For CSPF to be effective, the link-state IGP on all routers must support the special extensions. While building the topology database, the extended IGP must take into consideration the current LSPs and must flood the route information everywhere. Because changes in the reserved link bandwidth and link color cause database updates, an extended IGP tends to flood more frequently than a normal IGP. See Figure 40 on page 586 for a diagram of the relationships between these components.
- Currently active LSPs—Includes all the LSPs that should originate from the router and their current operational status (up, down, or timeout).

Figure 40: CSPF Computation Process



This section discusses the following topics:

- "How CSPF Selects a Path" on page 586
- "CSPF Path Selection Tie-Breaking" on page 587
- "Computing CSPF Paths Offline" on page 588

How CSPF Selects a Path

To select a path, CSPF follows certain rules. The rules are as follows:

1. Computes LSPs one at a time, beginning with the highest priority LSP (the one with the lowest setup priority value). Among LSPs of equal priority, CSPF services the LSPs in alphabetical order of the LSP names.
2. Prunes the traffic engineering database of all the links that are not full duplex and do not have sufficient reservable bandwidth.
3. If the LSP configuration includes the `include` statement, prunes all links that do not share any included colors.

4. If the LSP configuration includes the `exclude` statement, prunes all links that contain excluded colors. If the link does not have a color, it is accepted.
5. If several paths have equal cost, chooses the one whose last-hop address is the same as the LSP's destination.
6. If several equal cost paths remain, selects the one with the fewest number of hops.
7. If several equal-cost paths remain, applies the CSPF load-balancing rule configured on the LSP (least fill, most fill, or random).

CSPF finds the shortest path toward the LSP's egress router, taking into account explicit-path constraints. For example, if the path must pass through Router A, two separate SPFs are computed, one from the ingress router to Router A, the other from Router A to the egress router. All CSPF rules are applied to both computations.

CSPF Path Selection Tie-Breaking

If more than one path is still available after the CSPF rules ("How CSPF Selects a Path" on page 586) have been applied, a tie-breaking rule is applied to choose the path for the LSP. The rule used depends on the configuration. There are three tie-breaking rules:

- Random—One of the remaining paths is picked at random. This rule tends to place an equal number of LSPs on each link, regardless of the available bandwidth ratio. This is the default behavior.
- Least fill—The path with the largest minimum available bandwidth ratio is preferred. This rule tries to equalize the reservation on each link.
- Most fill—The path with the smallest minimum available bandwidth ratio is preferred. This rule tries to fill a link before moving on to alternative links.

The following definitions describe how a figure for minimum available bandwidth ratio is derived for the least fill and most fill rules:

- Reservable bandwidth = bandwidth of link x subscription factor of link
- Available bandwidth = reservable bandwidth – (sum of the bandwidths of the LSPs traversing the link)
- Available bandwidth ratio = available bandwidth/reservable bandwidth
- Minimum available bandwidth ratio (for a path) = the smallest available bandwidth ratio of the links in a path

NOTE: For the least fill or most fill behaviors to be used, the paths must have their bandwidth (specified using the `bandwidth` statement at the `[edit protocols mpls label-switched-path` *`lsp-name`*`]` hierarchy level) or minimum bandwidth (specified using the `minimum-bandwidth` statement at the `[edit protocols mpls label-switched-path` *`lsp-name`* `auto-bandwidth]` hierarchy level) configured to a value greater than 0. If the bandwidth or minimum bandwidth for the paths is either not configured or configured as 0, the minimum available bandwidth cannot be calculated and the random path selection behavior is used instead.

Computing CSPF Paths Offline

The Junos OS provides online, real-time CSPF computation only; each router performs CSPF calculations independent of the other routers in the network. These calculations are based on currently available topology information—information that is usually recent, but not completely accurate. LSP placements are locally optimized, based on current network status.

To optimize links globally across the network, you can use an offline tool to perform the CSPF calculations and determine the paths for the LSPs. You can create such a tool yourself, or you can modify an existing network design tool to perform these calculations. You should run the tool periodically (daily or weekly) and download the results into the router. An offline tool should take the following into account when performing the optimized calculations:

- All the LSP's requirements
- All link attributes
- Complete network topology

Configuring CSPF Tie Breaking

When selecting a path for an LSP, CSPF uses a tie-breaking process if there are several equal-cost paths. For information about how CSPF selects a path, see "How CSPF Selects a Path" on page 586.

You can configure one of the following statements (you can only configure one of these statements at a time) to alter the behavior of CSPF tie-breaking:

- By default, a random tie-breaking rule for CSPF is used to select a path from the set of equal-cost paths. However, you can also explicitly configure this behvior using the `random` statement:

```
random;
```

- To prefer the path with the least-utilized links, include the `least-fill` statement:

```
least-fill;
```

- To prefer the path with the most-utilized links, include the `most-fill` statement:

```
most-fill;
```

You can include each of these statements at the following hierarchy levels:

- `[edit protocols mpls label-switched-path` *`lsp-name`*`]`
- `[edit logical-systems` *`logical-system-name`* `protocols mpls label-switched-path` *`lsp-name`*`]`

Disabling Constrained-Path LSP Computation

If the IGP is a link-state protocol (such as IS-IS or OSPF) and supports extensions that allow the current bandwidth reservation on each router's link to be reported, constrained-path LSPs are computed by default.

The Junos implementations of IS-IS and OSPF include the extensions that support constrained-path LSP computation.

- IS-IS—These extensions are enabled by default. To disable this support, include the `disable` statement at the `[edit protocols isis traffic-engineering]` hierarchy level, as discussed in the Junos OS Routing Protocols Library for Routing Devices.
- OSPF—These extensions are disabled by default. To enable this support, include the `traffic-engineering` statement in the configurations of all routers running OSPF, as described in the Junos OS Routing Protocols Library for Routing Devices.

If IS-IS is enabled on a router or you enable OSPF traffic engineering extensions, MPLS performs the constrained-path LSP computation by default. For information about how constrained-path LSP computation works, see "Constrained-Path LSP Computation" on page 585.

Constrained-path LSPs have a greater chance of being established quickly and successfully for the following reasons:

- The LSP computation takes into account the current bandwidth reservation.
- Constrained-path LSPs reroute themselves away from node failures and congestion.

When constrained-path LSP computation is enabled, you can configure the LSP so that it is periodically reoptimized, as described in "Optimizing Signaled LSPs" on page 622.

When an LSP is being established or when an existing LSP fails, the constrained-path LSP computation is repeated periodically at the interval specified by the retry timer until the LSP is set up successfully. Once the LSP is set up, no recomputation is done. For more information about the retry timer, see "Configuring the Connection Between Ingress and Egress Routers" on page 599.

By default, constrained-path LSP computation is enabled. You might want to disable constrained-path LSP computation when all nodes do not support the necessary traffic engineering extensions. To disable constrained-path LSP computation, include the `no-cspf` statement:

```
no-cspf;
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

If you disable constrained-path LSP computation on LSPs by configuring the `no-cspf` statement and then attempt to advertise other LSPs with lower metrics than the IGPs from this router in either IS-IS or OSPF, new LSPs cannot be established.

RELATED DOCUMENTATION

MPLS Overview | 2

LSP Routers

IN THIS SECTION

- Routers in an LSP | 591
- Configuring the Ingress and Egress Router Addresses for LSPs | 591
- Configuring the Ingress Router for MPLS-Signaled LSPs | 594
- Configuring the Intermediate and Egress Routers for MPLS-Signaled LSPs | 599
- Configuring the Connection Between Ingress and Egress Routers | 599
- Pinging LSPs | 600

Routers in an LSP

Each router in an LSP performs one of the following functions:

- Ingress router—The router at the beginning of an LSP. This router encapsulates IP packets with an MPLS Layer 2 frame and forwards it to the next router in the path. Each LSP can have only one ingress router.
- Egress router—The router at the end of an LSP. This router removes the MPLS encapsulation, thus transforming it from an MPLS packet to an IP packet, and forwards the packet to its final destination using information in the IP forwarding table. Each LSP can have only one egress router. The ingress and egress routers in an LSP cannot be the same router.
- Transit router—Any intermediate router in the LSP between the ingress and egress routers. A transit router forwards received MPLS packets to the next router in the MPLS path. An LSP can contain zero or more transit routers, up to a maximum of 253 transit routers in a single LSP.

A single router can be part of multiple LSPs. It can be the ingress or egress router for one or more LSPs, and it also can be a transit router in one or more LSPs. The functions that each router supports depend on your network design.

Configuring the Ingress and Egress Router Addresses for LSPs

IN THIS SECTION

- Configuring the Ingress Router Address for LSPs | **591**
- Configuring the Egress Router Address for LSPs | **592**
- Preventing the Addition of Egress Router Addresses to Routing Tables | **593**

The following sections describe how to specify the addresses of an LSP's ingress and egress routers:

Configuring the Ingress Router Address for LSPs

The local router always is considered to be the ingress router, which is the beginning of the LSP. The software automatically determines the proper outgoing interface and IP address to use to reach the next router in an LSP.

By default, the router ID is chosen as the address of the ingress router. To override the automatic selection of the source address, specify a source address in the `from` statement:

```
from address;
```

You can include this statement at the following hierarchy levels:

- [edit protocols mpls label-switched-path *lsp-name*]
- [edit logical-systems *logical-system-name* protocols mpls label-switched-path *lsp-name*]

The outgoing interface used by the LSP is not affected by the source address that you configure.

Configuring the Egress Router Address for LSPs

When configuring an LSP, you must specify the address of the egress router by including the `to` statement:

```
to address;
```

You can include this statement at the following hierarchy levels:

- [edit protocols mpls label-switched-path *lsp-name*]
- [edit protocols mpls static-label-switched-path *lsp-name*]
- [edit logical-systems *logical-system-name* protocols mpls label-switched-path *lsp-name*]
- [edit logical-systems *logical-system-name* protocols mpls static-label-switched-path *lsp-name*]

When you are setting up a signaled LSP, the `to` statement is the only required statement. All other statements are optional.

After the LSP is established, the address of the egress router is installed as a host route in the routing table. This route can then be used by BGP to forward traffic.

To have the software send BGP traffic over an LSP, the address of the egress router is the same as the address of the BGP next hop. You can specify the egress router's address as any one of the router's interface addresses or as the BGP router ID. If you specify a different address, even if the address is on the same router, BGP traffic is not sent over the LSP.

To determine the address of the BGP next hop, use the `show route detail` command. To determine the destination address of an LSP, use the `show mpls lsp` command. To determine whether a route has gone through an LSP, use the `show route` or `show route forwarding-table` command. In the output of these last two commands, the `label-switched-path` or `push` keyword included with the route indicates it has passed

through an LSP. Also, use the traceroute command to trace the actual path to which the route leads. This is another indication whether a route has passed through an LSP.

You also can manipulate the address of the BGP next hop by defining a BGP import policy filter that sets the route's next-hop address.

Preventing the Addition of Egress Router Addresses to Routing Tables

You must configure an address using the to statement for all LSPs. This address is always installed as a /32 prefix in the inet.3 or inet.0 routing tables. You can prevent the egress router address configured using the to statement from being added to the inet.3 and inet.0 routing tables by including the no-install-to-address statement.

Some reasons not to install the to statement address in the inet.3 and inet.0 routing tables include the following:

- Allow Constrained Shortest Path First (CSPF) RSVP LSPs to be mapped to traffic intended for secondary loopback addresses. If you configure an RSVP tunnel, including the no-install-to-address statement, and then configure an install pfx/ <active> policy later, you can do the following:
  - Verify that the LSP was set up correctly without impacting traffic.
  - Map traffic to the LSP in incremental steps.
  - Map traffic to the destination loopback address (the BGP next hop) by removing the no-install-to-address statement once troubleshooting is complete.
- Prevent CCC connections from losing IP traffic. When an LSP determines that it does not belong to a connection, it installs the address specified with the to statement in the inet.3 routing table. IP traffic is then forwarded to the CCC remote endpoint, which can cause some types of PICs to fail.

To prevent the egress router address configured using the to statement from being added to the inet.3 and inet.0 routing tables, include the no-install-to-address statement:

```
no-install-to-address;
```

You can include this statement at the following hierarchy levels:

- [edit protocols mpls label-switched-path *lsp-name*]
- [edit protocols mpls static-label-switched-path *lsp-name*]
- [edit logical-systems *logical-system-name* protocols mpls label-switched-path *lsp-name*]
- [edit logical-systems *logical-system-name* protocols mpls static-label-switched-path *lsp-name*]

Configuring the Ingress Router for MPLS-Signaled LSPs

IN THIS SECTION

- Creating Named Paths | 594
- Configuring Alternate Backup Paths Using Fate Sharing | 596

MPLS-signaled label-switched paths (LSPs) run from a specific ingress router to a specific egress router. For basic MPLS-signaled LSP function, you must configure the ingress router, but do not have to configure any other routers.

To configure signaled LSPs, perform the following tasks on the ingress router:

Creating Named Paths

To configure signaled LSPs, you must first create one or more named paths on the ingress router. For each path, you can specify some or all transit routers in the path, or you can leave it empty.

Each pathname can contain up to 32 characters and can include letters, digits, periods, and hyphens. The name must be unique within the ingress router. Once a named path is created, you can use the named path with the `primary` or `secondary` statement to configure LSPs at the `[edit protocols mpls label-switched-path` *`label-path-name`*`]` hierarchy level. You can specify the same named path on any number of LSPs.

To determine whether an LSP is associated with the primary or secondary path in an RSVP session, issue the `show rsvp session detail` command.

To create an empty path, create a named path by including the following form of the `path` statement. This form of the `path` statement is empty, which means that any path between the ingress and egress routers is accepted. In actuality, the path used tends to be the same path as is followed by destination-based, best-effort traffic.

```
path path-name;
```

You can include this statement at the following hierarchy levels:

- `[edit protocols mpls]`
- `[edit logical-systems` *`logical-system-name`* `protocols mpls]`

To create a path in which you specify some or all transit routers in the path, include the following form of the path statement, specifying one address for each transit router:

```
path path-name {
    (address | hostname) <strict | loose>;
}
```

You can include this statement at the following hierarchy levels:

- [edit protocols mpls]
- [edit logical-systems *logical-system-name* protocols mpls]

In this form of the path statement, you specify one or more transit router addresses. Specifying the ingress or egress routers is optional. You can specify the address or hostname of each transit router, although you do not need to list each transit router if its type is loose. Specify the addresses in order, starting with the ingress router (optional) or the first transit router, and continuing sequentially along the path up to the egress router (optional) or the router immediately before the egress router. You need to specify only one address per router hop. If you specify more than one address for the same router, only the first address is used; the additional addresses are ignored and truncated.

For each router address, you specify the type, which can be one of the following:

- strict—(Default) The route taken from the previous router to this router is a direct path and cannot include any other routers. If *address* is an interface address, this router also ensures that the incoming interface is the one specified. Ensuring that the incoming interface is the one specified is important when there are parallel links between the previous router and this router. It also ensures that routing can be enforced on a per-link basis.

  For strict addresses, you must ensure that the router immediately preceding the router you are configuring has a direct connection to that router. The address can be a loopback interface address, in which case the incoming interface is not checked.

- loose—The route taken from the previous router to this router need not be a direct path, can include other routers, and can be received on any interface. The address can be any interface address or the address of the loopback interface.

Examples: Creating Named Paths

Configure a path, to-hastings, to specify the complete strict path from the ingress to the egress routers through 10.14.1.1, 10.13.1.1, 10.12.1.1, and 10.11.1.1, in that order. There cannot be any intermediate routers except the ones specified. However, there can be intermediate routers between 10.11.1.1 and the

egress router because the egress router is not specifically listed in the `path` statement. To prevent intermediate routers before egress, configure the egress router as the last router, with a `strict` type.

```
[edit protocols mpls]
path to-hastings {
    10.14.1.1 strict;
    10.13.1.1 strict;
    10.12.1.1 strict;
    10.11.1.1 strict;
}
```

Create a path, `alt-hastings`, to allow any number of intermediate routers between routers `10.14.1.1` and `10.11.1.1`. In addition, intermediate routers are permitted between `10.11.1.1` and the egress router.

```
[edit protocols mpls]
path alt-hastings {
    10.14.1.1 strict;
    10.11.1.1 loose;
}
```

Configuring Alternate Backup Paths Using Fate Sharing

You can create a database of information that Constrained Shortest Path First (CSPF) uses to compute one or more backup paths in case the primary path becomes unstable. The database describes the relationships between elements of the network, such as routers and links. Because these network elements share the same fate, this relationship is called fate sharing.

You can configure backup paths that minimize the number of shared links and fiber paths with the primary paths as much as possible to ensure that, if a fiber is cut, the minimum amount of data is lost and a path still exists to the destination.

For a backup path to work optimally, it must not share links or physical fiber paths with the primary path. This ensures that a single point of failure will not affect the primary and backup paths at the same time.

The following sections describe how to configure fate sharing and how it affects CSPF, and provides a fate sharing configuration example:

Configuring Fate Sharing

To configure fate sharing, include the `fate-sharing` statement:

```
fate-sharing {
    group group-name {
        cost value;
        from address <to address>;
    }
}
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

Each fate-sharing group must have a name, which can be up to 32 characters long and can contain letters, digits, periods (.) and hyphens (-). You can define up to 512 groups.

Fate-sharing groups contain three types of objects:

- Point-to-point links—Identified by the IP addresses at each end of the link. Unnumbered point-to-point links are typically identified by borrowing IP addresses from other interfaces. Order is not important; `from 10.1.3.4 to 10.1.3.5` and `from 10.1.3.5 to 10.1.3.4` have the same meaning.
- Non-point-to-point links—Include links on a LAN interface (such as Gigabit Ethernet interfaces) or nonbroadcast multiaccess (NBMA) interfaces (such as Asynchronous Transfer Mode [ATM] or Frame Relay). You identify these links by their individual interface address. For example, if the LAN interface `192.168.200.0/24` has four routers attached to it, each router link is individually identified:

  ```
  from 192.168.200.1;     # LAN interface of router 1
  from 192.168.200.2;     # LAN interface of router 2
  from 192.168.200.3;     # LAN interface of router 3
  from 192.168.200.4;     # LAN interface of router 4
  ```

  You can list the addresses in any order.
- A router node—Identified by its configured router ID.

All objects in a group share certain similarities. For example, you can define a group for all fibers that share the same fiber conduit, all optical channels that share the same fiber, all links that connect to the same LAN switch, all equipment that shares the same power source, and so on. All objects are treated as /32 host addresses.

For a group to be meaningful, it should contain at least two objects. You can configure groups with zero or one object; these groups are ignored during processing.

An object can be in any number of groups, and a group can contain any number of objects. Each group has a configurable cost attributed to it, which represents the level of impact this group has on CSPF computations. The higher the cost, the less likely a backup path will share with the primary path any objects in the group. The cost is directly comparable to traffic engineering metrics. By default, the cost is 1. Changing the fate-sharing database does not affect established LSPs until the next reoptimization of CSPF. The fate-sharing database does influence fast-reroute computations.

Implications for CSPF

When CSPF computes the primary paths of an LSP (or secondary paths when the primary path is not active), it ignores the fate-sharing information. You always want to find the best possible path (least IGP cost) for the primary path.

When CSPF computes a secondary path while the primary path (of the same LSP) is active, the following occurs:

1. CSPF identifies all fate-sharing groups that are associated with the primary path. CSPF does this by identifying all links and nodes that the primary path traverses and compiling group lists that contain at least one of the links or nodes. CSPF ignores the ingress and egress nodes in the search.
2. CSPF checks each link in the traffic engineering database against the compiled group list. If the link is a member of a group, the cost of the link is increased by the cost of the group. If a link is a member of multiple groups, all group costs are added together.
3. CSPF performs the check for every node in the traffic engineering database, except the ingress and egress node. Again, a node can belong to multiple groups, so costs are additive.
4. The router performs regular CSPF computation with the adjusted topology.

Implications for CSPF When Fate Sharing with Bypass LSPs

When fate sharing is enabled with link protection or link-node protection, CSPF operates as follows when calculating the bypass LSP path:

- CSPF identifies the fate-sharing groups that are associated with the primary LSP path. CSPF does this by identifying the immediate downstream link and immediate downstream nodes that the bypass is trying to protect. CSPF compiles group lists that contain the immediate downstream link and immediate downstream nodes.
- CSPF checks each link (from ingress to the immediate downstream node) in the traffic engineering database against the compiled group list. If the link is a member of a group, the cost of the link is increased by the cost of the group.
- CSPF identifies the downstream link that is not in the fate-shared path.

This calculation prevents bypasses from using the same physical link as the primary LSP path when viable alternatives are available.

Example: Configuring Fate Sharing

Configure fate-sharing groups `east` and `west`. Because `west` has no objects, it is ignored during processing.

```
[edit routing-options]
fate-sharing {
    group east {
        cost 20;           # Optional, default value is 1
        from 10.1.3.4 to 10.1.3.5; # A point-to-point link
        from 192.168.200.1;    # LAN interface
        from 192.168.200.2;     # LAN interface
        from 192.168.200.3;     # LAN interface
        from 192.168.200.4;   # LAN interface
        from 10.168.1.220;     # Router ID of a router node
        from 10.168.1.221;      # Router ID of a router node
    }
    group west {
        .....
    }
}
```

Configuring the Intermediate and Egress Routers for MPLS-Signaled LSPs

To configure signaled LSPs on all MPLS routers that should participate in MPLS, you need to enable MPLS and RSVP on these routers.

Configuring the Connection Between Ingress and Egress Routers

The ingress router might make many attempts to connect and reconnect to the egress router using the primary path. You can control how often the ingress router tries to establish a connection using the primary path and how long it waits between retry attempts.

The retry timer configures how long the ingress router waits before trying to connect again to the egress router using the primary path. The default retry time is 30 seconds. The time can be from 1 through 600 seconds. To modify this value, include the `retry-timer` statement:

```
retry-timer seconds;
```

You can configure this statement at the following hierarchy levels:

- [edit protocols mpls label-switched-path *lsp-name*]
- [edit logical-systems *logical-system-name* protocols mpls label-switched-path *lsp-name*]

By default, no limit is set to the number of times an ingress router attempts to establish or reestablish a connection to the egress router using the primary path. To limit the number of attempts, include the retry-limit statement:

```
retry-limit number;
```

You can configure this statement at the following hierarchy levels:

- [edit protocols mpls label-switched-path *lsp-name*]
- [edit logical-systems *logical-system-name* protocols mpls label-switched-path *lsp-name*]

The limit can be a value up to 10,000. When the retry limit is exceeded, no more attempts are made to establish a path connection. At this point, intervention is required to restart the primary path.

If you set a retry limit, it is reset to 1 each time a successful primary path is created.

Pinging LSPs

IN THIS SECTION

- Pinging MPLS LSPs | **600**
- Pinging Point-to-Multipoint LSPs | **601**
- Pinging the Endpoint Address of MPLS LSPs | **601**
- Pinging CCC LSPs | **601**
- Pinging Layer 3 VPNs | **602**
- Support for LSP Ping and Traceroute Commands Based on RFC 4379 | **602**

The following sections describe how to use the ping mpls command to confirm LSP functioning.

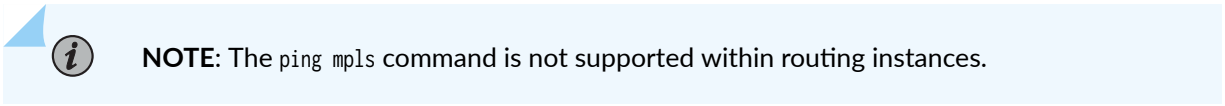
Pinging MPLS LSPs

You can ping a specific LSP. Echo requests are sent over the LSP as MPLS packets. The payload is a User Datagram Protocol (UDP) packet forwarded to an address in the 127/8 range (127.0.0.1 by default, this address is configurable) and port 8503. The label and interface information for building and sending this information as an MPLS packet is the same as for standard LSP traffic.

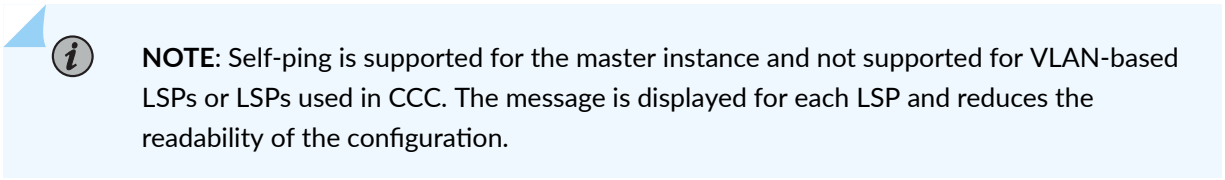
When the echo request arrives at the egress node, the receiver checks the contents of the packet and sends a reply containing the correct return value, by using UDP. The router sending the echo request waits to receive an echo reply after a timeout of 2 seconds (you cannot configure this value).

You must configure MPLS at the `[edit protocols mpls]` hierarchy level on the remote router to be able to ping an LSP terminating there. You must configure MPLS even if you intend to ping only LDP forwarding equivalence classes (FECs).

To ping an MPLS LSP use the `ping mpls <count count> <ldp <fec>> <rsvp <exp forwarding-class> <lsp-name>>` command. To ping a secondary MPLS LSP, use the `ping mpls <count count> <rsvp <lsp-name>> standby path-name` command. For a detailed description of this command, see the CLI Explorer.



NOTE: The `ping mpls` command is not supported within routing instances.



NOTE: Self-ping is supported for the master instance and not supported for VLAN-based LSPs or LSPs used in CCC. The message is displayed for each LSP and reduces the readability of the configuration.

Pinging Point-to-Multipoint LSPs

To ping a point-to-multipoint LSP, use the `ping mpls rsvp lsp-name multipoint` or `ping mpls rsvp egress address` commands. The `ping mpls rsvp lsp-name multipoint` command returns a list of all of the egress router identifiers and the current status of the point-to-multipoint LSP egress routers. The `ping mpls rsvp lsp-name multipoint egress address` command returns the current status of the specified egress router.

Pinging the Endpoint Address of MPLS LSPs

To determine whether an LSP between two provider edge (PE) routers is up and running, you can ping the endpoint address of the LSP. To ping an MPLS LSP endpoint, use the `ping mpls lsp-end-point address` command. This command tells you what type of LSP (RSVP or LDP) terminates at the address specified and whether that LSP is up or down.

For a detailed description of this command, see the CLI Explorer.

Pinging CCC LSPs

You can ping a specific CCC LSP. The CCC LSP ping command is identical to the one used for MPLS LSPs. The command you use is `ping mpls <count count> <rsvp <lsp-name>>`. You can also ping a secondary standby CCC LSP by using the `ping mpls <count count> <rsvp <lsp-name>> standby path-name` command.

For a detailed description of this command, see the CLI Explorer.

Pinging Layer 3 VPNs

You can use a similar command, `ping mpls l3vpn` *`vpn-name`* `prefix` *`prefix`* `<count` *`count`*`>`, to ping a Layer 3 VPN. For more information about this command, see the Junos OS VPNs Library for Routing Devices and the CLI Explorer.

Support for LSP Ping and Traceroute Commands Based on RFC 4379

The Junos OS supports LSP `ping` and `traceroute` commands based on RFC 4379, *Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures*.

LSP `ping` and `traceroute` commands based on RFC 4379 attempt to trace the path taken by an LSP by relying on MPLS TTL expiration. An LSP can take multiple paths from ingress to egress. This occurs in particular with Equal Cost Multipath (ECMP). The LSP `traceroute` command can trace all possible paths to an LSP node.

RELATED DOCUMENTATION

Basic MPLS Configuration | 46

CHAPTER 9

Configuring MPLS LSPs

IN THIS CHAPTER

- Basic LSP Configuration | **603**
- Primary, Secondary, and Static LSP Configuration | **683**
- Adaptive LSP Configuration | **709**
- Container LSP Configuration | **710**
- Multiclass LSP Configuration | **778**
- Point-to-Multipoint LSP Configuration | **780**
- Pop-and-Forward LSP Configuration | **826**
- Segment Routing LSP Configuration | **833**
- Express Segment LSP Configuration | **926**

Basic LSP Configuration

IN THIS SECTION

- Configuring RSVP-signaled MPLS Label-Switched Path (LSP) Metrics | **604**
- Configuring a Text Description for RSVP-signaled MPLS Label-Switched Paths (LSP) and Static LSPs | **609**
- Configuring RSVP-signaled MPLS Label-Switched Path Soft Preemption | **611**
- Configuring Priority and Preemption for RSVP-signaled MPLS Label-Switched Paths | **612**
- Configuring Administrative Groups for LSPs | **613**
- Configuring Extended Administrative Groups for LSPs | **616**
- Configuring the Junos Routing Preference Value for RSVP-signaled MPLS LSPs | **618**
- Disabling Path Route Recording by RSVP-signaled MPLS LSPs | **618**
- Achieving a Make-Before-Break, Hitless Switchover for RSVP-signaled MPLS LSPs | **618**
- Optimizing Signaled LSPs | **622**

- Configuring the Smart Optimize Timer for LSPs | **625**
- Limiting the Number of Hops in the Path of an RSVP-signaled MPLS LSP | **627**
- Configuring a Manually-Defined Bandwidth Value for RSVP-signaled MPLS LSPs | **627**
- Automatic Bandwidth Allocation for LSPs | **627**
- Configuring Automatic Bandwidth Allocation for LSPs | **628**
- Configuring Optimized Auto-bandwidth Adjustments for MPLS LSPs | **629**
- Configuring Reporting of Automatic Bandwidth Allocation Statistics for LSPs | **633**
- Using Polling Profiles for Automatic Bandwidth Allocation for RSVP LSPs | **637**
- Configuring an RSVP-signaled MPLS LSP to Calculate a Path Across Autonomous Systems | **639**
- Damping IGP Advertisements of RSVP-signaled MPLS LSP State Changes | **641**
- Configuring Corouted Bidirectional RSVP-signaled MPLS LSPs | **641**
- Configuring the Entropy Label for LSPs | **644**
- Example: Configuring an Entropy Label for a BGP Labeled Unicast LSP | **646**
- Configuring Ultimate-Hop Popping for RSVP-signaled MPLS LSPs | **671**
- Configuring Explicit-Path LSPs | **675**
- Example: Configuring an Explicit-Path LSP | **676**
- LSP Bandwidth Oversubscription Overview | **677**
- LSP Size Oversubscription | **678**
- LSP Link Size Oversubscription | **678**
- Class Type Oversubscription and Local Oversubscription Multipliers | **679**
- Configuring the Bandwidth Subscription Percentage for LSPs | **679**
- Detecting MPLS MTU Exceed Errors | **681**

Configuring RSVP-signaled MPLS Label-Switched Path (LSP) Metrics

IN THIS SECTION

- Configuring Dynamic LSP Metrics | **605**
- Configuring Static LSP Metrics | **605**
- RSVP LSP Conditional Metrics | **606**
- Preserve the IGP Metric in RSVP LSP Routes | **607**

- Sample: Configuring RSVP LSP Conditional Metrics and Preserving the IGP Metric in RSVP LSP Routes | **607**

The LSP metric is used to indicate the ease or difficulty of sending traffic over a particular LSP. Lower LSP metric values (lower cost) increase the likelihood of an LSP being used. Conversely, high LSP metric values (higher cost) decrease the likelihood of an LSP being used.

The LSP metric can be specified dynamically by the router or explicitly by the user as described in the following sections:

Configuring Dynamic LSP Metrics

If no specific metric is configured, an LSP attempts to track the IGP metric toward the same destination (the `to` address of the LSP). IGP includes OSPF, IS-IS, Routing Information Protocol (RIP), and static routes. BGP and other RSVP or LDP routes are excluded.

For example, if the OSPF metric toward a router is 20, all LSPs toward that router automatically inherit metric 20. If the OSPF toward a router later changes to a different value, all LSP metrics change accordingly. If there are no IGP routes toward the router, the LSP raises its metric to 65,535.

Note that in this case, the LSP metric is completely determined by IGP; it bears no relationship to the actual path the LSP is currently traversing. If LSP reroutes (such as through reoptimization), its metric does not change, and thus it remains transparent to users. Dynamic metric is the default behavior; no configuration is required.

Configuring Static LSP Metrics

You can manually assign a fixed metric value to an LSP. Once configured with the `metric` statement, the LSP metric is fixed and cannot change:

```
metric number;
```

You can include this statement at the following hierarchy levels:

- [edit protocols mpls label-switched-path *lsp-name*]
- [edit protocols mpls static-label-switched-path *lsp-name*]
- [edit logical-systems *logical-system-name* protocols mpls label-switched-path *lsp-name*]
- [edit logical-systems *logical-system-name* protocols mpls static-label-switched-path *lsp-name*]

The LSP metric has several uses:

- When there are parallel LSPs with the same egress router, the metrics are compared to determine which LSP has the lowest metric value (the lowest cost) and therefore the preferred path to the destination. If the metrics are the same, the traffic is shared.

  Adjusting the metric values can force traffic to prefer some LSPs over others, regardless of the underlying IGP metric.

- When an IGP shortcut is enabled (see *Using Labeled-Switched Paths to Augment SPF to Compute IGP Shortcuts*), an IGP route might be installed in the routing table with an LSP as the next hop, if the LSP is on the shortest path to the destination. In this case, the LSP metric is added to the other IGP metrics to determine the total path metric. For example, if an LSP whose ingress router is X and egress router is Y is on the shortest path to destination Z, the LSP metric is added to the metric for the IGP route from Y to Z to determine the total cost of the path. If several LSPs are potential next hops, the total metrics of the paths are compared to determine which path is preferred (that is, has the lowest total metric). Or, IGP paths and LSPs leading to the same destination could be compared by means of the metric value to determine which path is preferred.

  By adjusting the LSP metric, you can force traffic to prefer LSPs, prefer the IGP path, or share the load among them.

- If router X and Y are BGP peers and if there is an LSP between them, the LSP metric represents the total cost to reach Y from X. If for any reason the LSP reroutes, the underlying path cost might change significantly, but X's cost to reach Y remains the same (the LSP metric), which allows X to report through a BGP multiple exit discriminator (MED) a stable metric to downstream neighbors. As long as Y remains reachable through the LSP, no changes are visible to downstream BGP neighbors.

It is possible to configure IS-IS to ignore the configured LSP metric by including the `ignore-lsp-metrics` statement at the `[edit protocols isis traffic-engineering shortcuts]` hierarchy level. This statement removes the mutual dependency between IS-IS and MPLS for path computation. For more information, see the Junos OS Routing Protocols Library for Routing Devices.

RSVP LSP Conditional Metrics

Conditional metric provides the capability to use different metric values conditionally for local statically configured label-switched paths (LSPs). The conditional metrics are based on the dynamically changing IGP metric. Junos OS changes the LSP metric to the configured conditional metric that corresponds to the highest threshold reached by the IGP metric. If there are no matching conditions, the LSP uses the IGP metric of the route. You can configure up to four conditional metrics for an LSP and they will be in sorted order.

If you configure the `track-igp-metric` statement with the conditional metric configuration, Junos OS uses the IGP metric of the installed routes to evaluate the configured conditional metric. You cannot configure static metric along with conditional metric.

Preserve the IGP Metric in RSVP LSP Routes

When you use the `conditional-metric` statement to configure RSVP LSPs, the resulting metric might be different from the actual IGP metric for the LSP destination. RSVP programs the LSP ingress route with this conditional metric as the route's metric. But in certain situations, there may be a need to preserve the actual IGP metric used by conditional metric for later use, such as calculating the BGP MED value.

Use the `include-igp-metric` statement in conjunction with the `conditional-metric` statement to include the IGP metric information in the RSVP route.

Run the `show route protocol rsvp extensive` command to view the updated actual IGP cost.

NOTE: This is only applicable to RSVP routes using the conditional metric. RSVP routes that use dynamic IGP include the IGP metric by default.

For more information, see the *include-igp-metric* configuration statement.

Sample: Configuring RSVP LSP Conditional Metrics and Preserving the IGP Metric in RSVP LSP Routes

Here is a sample configuring using a simple four-router topology with an ingress device that can reach an egress device through two transit devices. Three RSVP LSPs are configured from the ingress to egress device with OSPF as the IGP. The LSPs are as follows:

- LSP1 (strict ERO, metric = 80)
- LSP2 (strict ERO, metric = 120)
- LSP3 (conditional metric based on OSPF)

To preserve the IGP metric, include the **include-igp-metric** CLI statement on the ingress device. The routing decision is as follows:

- When OSPF metric is low, LSP3 may be preferred over LSP2.
- As OSPF metric increases, LSP3 becomes less preferred due to higher conditional metrics.
- This dynamic behavior allows LSP3 to act as a fallback path, while LSP1 and LSP2 remain preferred static paths.

On the ingress device:

1. Configure interfaces and enable MPLS and RSVP.

```
[edit]
user@host# set interfaces ge-0/0/0 unit 0 description "Connected to Transit 1"
```

```
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.0.0.1/30
user@host# set interfaces ge-0/0/1 unit 0 description "Connected to Transit 2"
user@host# set interfaces ge-0/0/1 unit 0 family inet address 20.0.0.1/30
user@host# set interfaces lo0 unit 0 family inet address 192.168.100.1/32
user@host# set protocols mpls interface ge-0/0/0
user@host# set protocols mpls interface ge-0/0/1
user@host# set protocols rsvp interface ge-0/0/0
user@host# set protocols rsvp interface ge-0/0/1
```

2. Configure OSPF.

```
[edit]
user@host# set protocols ospf area 0.0.0.0 interface ge-0/0/0
user@host# set protocols ospf area 0.0.0.0 interface ge-0/0/1
```

3. Define RSVP LSPs.

```
[edit]
user@host# set protocols mpls label-switched-path LSP1 to 192.168.100.4
user@host# set protocols mpls label-switched-path LSP1 strict
user@host# set protocols mpls label-switched-path LSP1 explicit-route 10.0.0.2
user@host# set protocols mpls label-switched-path LSP1 explicit-route 30.0.0.2
user@host# set protocols mpls label-switched-path LSP1 priority 7 7

user@host# set protocols mpls label-switched-path LSP2 to 192.168.100.4
user@host# set protocols mpls label-switched-path LSP2 strict
user@host# set protocols mpls label-switched-path LSP2 explicit-route 20.0.0.2
user@host# set protocols mpls label-switched-path LSP2 explicit-route 40.0.0.2
user@host# set protocols mpls label-switched-path LSP2 priority 7

user@host# set protocols mpls label-switched-path LSP3 to 192.168.100.4
user@host# set protocols mpls label-switched-path LSP3 track-igp-metric
user@host# set protocols mpls label-switched-path LSP3 conditional-metric threshold 50 metric
100
user@host# set protocols mpls label-switched-path LSP3 conditional-metric threshold 100
metric 200
user@host# set protocols mpls label-switched-path LSP3 conditional-metric threshold 150
metric 300
user@host# set protocols mpls label-switched-path LSP3 conditional-metric threshold 200
metric 400
```

```
user@host# set protocols mpls label-switched-path LSP3 include-igp-metric
user@host# set protocols mpls label-switched-path LSP3 priority 0 0
```

Verify the configuration:

To verify if the configuration is working, following these verification steps:

- Verify that interfaces are up and have correct IP addresses. Confirm correct subnetting and peer IPs.
- Check MPLS and RSVP is enabled on all relevant interfaces.
- Confirm OSPF neighbors are established.
- Verify all LSPs are up. Confirm LSP paths match explicit routes for LSP1 and LSP2.
- Verify RSVP LSP Behavior use the show route protocol rsvp extensive command, where the 'Actual IGP metric' value is 2.

Junos OS tracks the IGP metric to the LSP destination (192.168.100.4). If the IGP metric crosses a threshold (100), Junos applies the corresponding conditional metric (200). The `include-igp-metric` ensures the original IGP metric is preserved in the RSVP route for use in BGP MED or other calculations.

Configuring a Text Description for RSVP-signaled MPLS Label-Switched Paths (LSP) and Static LSPs

You can provide a textual description for an LSP by enclosing any descriptive text that includes spaces within quotation marks (" "). The descriptive text you include is displayed in the detail output of the `show mpls lsp` or the `show mpls container-lsp` command.

Adding a text description for an LSP has no effect on the operation of the LSP. The LSP text description can be no more than 80 characters in length.

To provide a textual description for an LSP, include the `description` statement at any of the following hierarchy levels:

- [edit protocols mpls label-switched-path *lsp-name*]
- [edit protocols mpls container-label-switched-path *lsp-name*]
- [edit protocols mpls static-label-switched-path *lsp-name*]
- [edit logical-systems *logical-system-name* protocols mpls label-switched-path *lsp-name*]
- [edit logical-systems *logical-system-name* protocols mpls static-label-switched-path *lsp-name*]

Before you begin:

- Configure the device interfaces.

- Configure the device for network communication.
- Enable MPLS on the device interfaces.
- Configure an LSP in the MPLS domain.

To add a text description for an LSP:

1. Enter any text describing the LSP.

```
[edit protocols mpls lsp lsp-name]
user@host# set description text
```

For example:

```
[edit protocols mpls lsp LSP1]
user@host# set description "Connecting remote device"
```

2. Verify and commit the configuration.

For example:

```
[edit protocols mpls lsp]
user@host# set protocols mpls label-switched-path LSP1 to 10.1.1.1
user@host# set protocols mpls label-switched-path LSP1 description "Connecting remote device"
user@host# set protocols mpls interface ge-1/0/8.0
```

```
[edit]
user@host# commit
commit complete
```

3. View the description of an LSP using the `show mpls lsp detail` or `show mpls container-lsp detail` command, depending on the type of LSP configured.

```
user@host> show mpls lsp detail

Ingress LSP: 1 sessions

10.1.1.1
  From: 0.0.0.0, State: Up, ActiveRoute: 1, LSPname: LSP1
```

```
  Description: Connecting remote device
  ActivePath: (none)
  LSPtype: Static Configured, Penultimate hop popping
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
  Primary                   State: Up
    Priorities: 7 0
    SmartOptimizeTimer: 180
        No computed ERO.
Total 1 displayed, Up 1, Down 0

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Configuring RSVP-signaled MPLS Label-Switched Path Soft Preemption

Soft preemption attempts to establish a new path for a preempted LSP before tearing down the original LSP. The default behavior is to tear down a preempted LSP first, signal a new path, and then reestablish the LSP over the new path. In the interval between when the path is taken down and the new LSP is established, any traffic attempting to use the LSP is lost. Soft preemption prevents this type of traffic loss. The trade-off is that during the time when an LSP is being soft preempted, two LSPs with their corresponding bandwidth requirements are used until the original path is torn down.

MPLS soft preemption is useful for network maintenance. For example, you can move all LSPs away from a particular interface, then take the interface down for maintenance without interrupting traffic. MPLS soft preemption is described in detail in RFC 5712, *MPLS Traffic Engineering Soft Preemption*.

Soft preemption is a property of the LSP and is disabled by default. You configure it at the ingress of an LSP by including the `soft-preemption` statement:

```
soft-preemption;
```

You can include this statement at the following hierarchy levels:

- [edit protocols mpls label-switched-path *lsp-name*]
- [edit logical-systems *logical-system-name* protocols mpls label-switched-path *lsp-name*]

You can also configure a timer for soft preemption. The timer designates the length of time the router should wait before initiating a hard preemption of the LSP. At the end of the time specified, the LSP is torn down and resignaled. The soft-preemption cleanup timer has a default value of 30 seconds; the

range of permissible values is 0 through 180 seconds. A value of 0 means that soft preemption is disabled. The soft-preemption cleanup timer is global for all LSPs.

Configure the timer by including the `cleanup-timer` statement:

```
cleanup-timer seconds;
```

You can include this statement at the following hierarchy levels:

- `[edit protocols rsvp preemption soft-preemption]`
- `[edit logical-systems logical-system-name protocols rsvp preemption soft-preemption]`

NOTE: Soft preemption cannot be configured on LSPs for which fast reroute has been configured. The configuration fails to commit. However, you can enable soft preemption in conjunction with node and link protection.

NOTE: The counter value for *SoftPreemptionCnt* initializes with a value of 0 (zero), visible in the command `show rsvp interface detail` output.

Configuring Priority and Preemption for RSVP-signaled MPLS Label-Switched Paths

When there is insufficient bandwidth to establish a more important LSP, you might want to tear down a less important existing LSP to free the bandwidth. You do this by preempting the existing LSP.

Whether an LSP can be preempted is determined by two properties associated with the LSP:

- Setup priority—Determines whether a new LSP that preempts an existing LSP can be established. For preemption to occur, the setup priority of the new LSP must be higher than that of the existing LSP. Also, the act of preempting the existing LSP must produce sufficient bandwidth to support the new LSP. That is, preemption occurs only if the new LSP can be set up successfully.
- Reservation priority—Determines the degree to which an LSP holds on to its session reservation after the LSP has been set up successfully. When the reservation priority is high, the existing LSP is less likely to give up its reservation, and hence it is unlikely that the LSP can be preempted.

You cannot configure an LSP with a high setup priority and a low reservation priority, because permanent preemption loops might result if two LSPs are allowed to preempt each other. You must configure the reservation priority to be higher than or equal to the setup priority.

The setup priority also defines the relative importance of LSPs on the same ingress router. When the software starts, when a new LSP is established, or during fault recovery, the setup priority determines

the order in which LSPs are serviced. Higher-priority LSPs tend to be established first and hence enjoy more optimal path selection.

To configure the LSP's preemption properties, include the `priority` statement:

```
priority setup-priority reservation-priority;
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

Both *setup-priority* and *reservation-priority* can be a value from 0 through 7. The value 0 corresponds to the highest priority, and the value 7 to the lowest. By default, an LSP has a setup priority of 7 (that is, it cannot preempt any other LSPs) and a reservation priority of 0 (that is, other LSPs cannot preempt it). These defaults are such that preemption does not happen. When you are configuring these values, the setup priority should always be less than or equal to the hold priority.

Configuring Administrative Groups for LSPs

Administrative groups, also known as link coloring or resource class, are manually assigned attributes that describe the "color" of links, such that links with the same color conceptually belong to the same class. You can use administrative groups to implement a variety of policy-based LSP setups.

Administrative groups are meaningful only when constrained-path LSP computation is enabled.

You can assign up to 32 names and values (in the range 0 through 31), which define a series of names and their corresponding values. The administrative names and values must be identical across all routers within a single domain.

NOTE: The administrative value is distinct from the priority. You configure the priority for an LSP using the `priority` statement. See "Configuring Priority and Preemption for LSPs" on page 612.

To configure administrative groups, follow these steps:

1. Define multiple levels of service quality by including the `admin-groups` statement:

   ```
   admin-groups {
       group-name group-value;
   }
   ```

   You can include this statement at the following hierarchy levels:

   - `[edit protocols mpls]`

- [edit logical-systems *logical-system-name* protocols mpls]

The following configuration example illustrates how you might configure a set of administrative names and values for a domain:

```
[edit protocols mpls]
admin-groups {
    gold 1;
    silver 2;
    copper 3;
    best-effort 4;
}
```

2. Define the administrative groups to which an interface belongs. You can assign multiple groups to an interface. Include the `interface` statement:

```
interface interface-name {
    admin-group [ group-names ];
}
```

You can include this statement at the following hierarchy levels:

- [edit protocols mpls]
- [edit logical-systems *logical-system-name* protocols mpls]

If you do not include the `admin-group` statement, an interface does not belong to any group.

IGPs use the group information to build link-state packets, which are then flooded throughout the network, providing information to all nodes in the network. At any router, the IGP topology, as well as administrative groups of all the links, is available.

Changing the interface's administrative group affects only new LSPs. Existing LSPs on the interface are not preempted or recomputed to keep the network stable. If LSPs need to be removed because of a group change, issue the `clear rsvp session` command.

NOTE: When configuring administrative groups and extended administrative groups together for a link, both the types of administrative groups must be configured on the interface.

3. Configure an administrative group constraint for each LSP or for each primary or secondary LSP path. Include the `label-switched-path` statement:

```
label-switched-path lsp-name {
    to address;
    ...
    admin-group {
        exclude [ group-names ];
        include-all [ group-names ];
        include-any [ group-names ];
    }
    primary path-name {
        admin-group {
            exclude [ group-names ];
            include-all [ group-names ];
            include-any [ group-names ];
        }
    }
    secondary path-name {
        admin-group {
            exclude [ group-names ];
            include-all [ group-names ];
            include-any [ group-names ];
        }
    }
}
```

You can include the `label-switched-path` statement at the following hierarchy levels:

- `[edit protocols mpls]`
- `[edit logical-systems logical-system-name protocols mpls]`

If you omit the `include-all`, `include-any`, or `exclude` statements, the path computation proceeds unchanged. The path computation is based on the constrained-path LSP computation. For information about how the constrained-path LSP computation is calculated, see "How CSPF Selects a Path" on page 586.

NOTE: Changing the LSP's administrative group causes an immediate recomputation of the route; therefore, the LSP might be rerouted.

Configuring Extended Administrative Groups for LSPs

In MPLS traffic engineering, a link can be configured with a set of administrative groups (also known as colors or resource classes). Administrative groups are carried in the interior gateway protocol (IGP) (OSPFv2 and IS-IS) as a 32-bit value assigned to each link. Juniper Networks routers normally interpret this 32-bit value as a bit mask with each bit representing a group, limiting each network to a total of 32 distinct administrative groups (value range 0 through 31).

You configure extended administrative groups, represented by a 32-bit value, expanding the number of administrative groups supported in the network beyond just 32. The original range of values available for administrative groups is still supported for backwards compatibility.

The extended administrative groups configuration accepts a set of interfaces with a corresponding set of extended administrative group names. It converts the names into a set of 32-bit values and propagates this information into the IGP. The extended administrative group values are global and must be identically configured on all the supported routers participating in the network. The domain-wide extended administrative groups database, learned from other routers through IGP flooding, is used by Constrained Shortest Path First (CSPF) for path computation.

The following procedure describes how to configure extended administrative groups:

1. Configure the `admin-groups-extended-range` statement:

```
admin-groups-extended-range {
    maximum maximum-number;
    mininum minimum-number;
}
```

   You can include this statement at the following hierarchy levels:

   - `[edit routing-options]`
   - `[edit logical-systems logical-system-name routing-options]`

   The `admin-groups-extended-range` statement includes the `minimum` and `maximum` options. The range maximum must be greater than the range minimum.

2. Configure the `admin-groups-extended` statement:

```
admin-groups-extended group-name {
    group-value group-identifier;
}
```

   You can include this statement at the following hierarchy levels:

- [edit routing-options]
- [edit logical-systems *logical-system-name* routing-options]

The admin-groups-extended statement enables you to configure a group name and group value for the administrative group. The group value must be within the range of values configured using the admin-groups-extended-range statement.

3. The extended administrative groups for an MPLS interface consist of the set of extended administrative group names assigned for the interface. The interface extended administrative group names must be configured for the global extended administrative groups.

   To configure an extended administrative group for an MPLS interface, specify the administrative group name within the MPLS interface configuration using the admin-groups-extended statement:

   ```
   admin-groups-extended group-name;
   ```

   You can include this statement at the following hierarchy levels:

   - [edit protocols mpls interface *interface-name*]
   - [edit logical-systems *logical-system-name* protocols mpls interface *interface-name*]

4. The LSP extended administrative groups define the set of include and exclude constraints for an LSP and for a path's primary and secondary paths. The extended administrative group names must be configured for the global extended administrative groups.

   To configure extended administrative groups for an LSP, include the admin-group-extended statement at an LSP hierarchy level:

   ```
   admin-group-extended {
       apply-groups group-value;
       apply-groups-except group-value;
       exclude group-value;
       include-all  group-value;
       include-any  group-value;
   }
   ```

   The admin-group-extended statement includes the following options: apply-groups, apply-groups-except, exclude, include-all, and include-any. Each option enables you to configure one or more extended administrative groups.

   For the list of the hierarchy levels at which you can configure this statement, see the statement summary for this statement.

5. To display the currently configured extended administrative groups, issue the show mpls admin-groups-extended command.

NOTE: When configuring administrative groups and extended administrative groups together for a link, both the types of administrative groups must be configured on the interface.

Configuring the Junos Routing Preference Value for RSVP-signaled MPLS LSPs

As an option, you can configure multiple LSPs between the same pair of ingress and egress routers. This is useful for balancing the load among the LSPs because all LSPs, by default, have the same preference level. To prefer one LSP over another, set different preference levels for individual LSPs. The LSP with the lowest preference value is used. The default preference for RSVP LSPs is 7 and for LDP LSPs is 9. These preference values are lower (more preferred) than all learned routes except direct interface routes.

To change the default preference value, include the `preference` statement:

```
preference preference;
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

Disabling Path Route Recording by RSVP-signaled MPLS LSPs

The Junos implementation of RSVP supports the Record Route object, which allows an LSP to actively record the routers through which it transits. You can use this information for troubleshooting and to prevent routing loops. By default, path route information is recorded. To disable recording, include the `no-record` statement:

```
no-record;
```

For a list of hierarchy levels at which you can include the `record` and `no-record` statements, see the statement summary section for the statement.

Achieving a Make-Before-Break, Hitless Switchover for RSVP-signaled MPLS LSPs

IN THIS SECTION

- Specifying the Amount of Time the Router Waits to Switch Over to New Paths | **620**
- Specifying the Amount of Time to Delay the Tear Down of Old Paths | **620**

Achieving a Hitless, MBB Switchover Without Artificial Delays | 621

Adaptive label-switched paths (LSPs) might need to establish a new LSP instance and transfer traffic from an old LSP instance onto the new LSP instance before tearing down the old one. This type of configuration is referred to as a *make before break* (MBB).

RSVP-TE is a protocol used to establish LSPs in MPLS networks. The Junos OS implementation of RSVP-TE to achieve a hitless (no traffic loss) MBB switchover has relied on configuring the timer values in the following configuration statements:

- `optimize-switchover-delay`—Amount of time to wait before switching to the new LSP instance.
- `optimize-hold-dead-delay`—Amount of time to wait after switchover and before deletion of the old LSP instance.

Both the `optimize-switchover-delay` and `optimize-hold-dead-delay` statements apply to all LSPs that use the make-before-break behavior for LSP setup and teardown, not just for LSPs for which the `optimize-timer` statement has also been configured. The following MPLS features cause LSPs to be set up and torn down using make-before-break behavior:

- Adaptive LSPs
- Automatic bandwidth allocation
- BFD for LSPs
- Graceful Routing Engine switchover
- Link and node protection
- Nonstop active routing
- Optimized LSPs
- Point-to-multipoint (P2MP) LSPs
- Soft preemption
- Standby secondary paths

Both the `optimize-switchover-delay` and `optimize-hold-dead-delay` statements when configured add an artificial delay to the MBB process. The value of the `optimize-switchover-delay` statement varies with the size of the Explicit Route Objects (EROs). An ERO is an extension to RSVP that allows an RSVP PATH message to traverse an explicit sequence of routers that is independent of conventional shortest-path IP routing.

The value of the optimize-switchover-delay statement also depends on the CPU load on each of the routers on the path. Customers set the optimize-switchover-delay statement by trial and error.

The value of the optimize-hold-dead-delay statement depends on how fast the ingress router moves all application prefixes to point to the new LSP. This is determined by the Packet Forwarding Engine load, which can vary from platform to platform. Customers have to set the optimize-hold-dead-delay statement by trial and error.

However, as of Release 15.1, Junos OS is able to achieve a hitless MBB switchover without configuring the artificial delays that such timer values introduce.

This topic summarizes the three methods of achieving a MBB switchover from an old LSP to a new LSP using Junos OS:

Specifying the Amount of Time the Router Waits to Switch Over to New Paths

To specify the amount of time the router waits to switch over LSP instances to newly optimized paths, use the optimize-switchover-delay statement. You only need to configure this statement on routers acting as the ingress for the affected LSPs (you do not need to configure this statement on transit or egress routers). The timer in this statement helps to ensure that the new optimized paths have been established before traffic is switched over from the old paths. This timer can only by enabled or disabled for all of the LSPs configured on the router.

To configure the amount of time the router waits to switch over LSP instances to newly optimized paths, specify the time in seconds by using the optimize-switchover-delay statement:

```
optimize-switchover-delay seconds;
```

You can include this statement at the following hierarchy levels:

- [edit protocols mpls]
- [edit logical-systems *logical-system-name* protocols mpls]

Specifying the Amount of Time to Delay the Tear Down of Old Paths

To specify the amount of time to delay the tear down of old paths after the router has switched traffic to new optimized paths, use the optimize-hold-dead-delay statement. You only need to configure this statement on routers acting as the ingress for the affected LSPs (you do not need to configure this statement on transit or egress routers). The timer in this statement helps to ensure that old paths are not torn down before all routes have been switched over to the new optimized paths. This timer can be enabled for specific LSPs or for all of the LSPs configured on the router.

To configure the amount of time in seconds to delay the tear down of old paths after the router has switched traffic to new optimized paths, use the `optimize-hold-dead-delay` statement:

```
optimize-hold-dead-delay seconds;
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

Achieving a Hitless, MBB Switchover Without Artificial Delays

As of Junos OS Release 15.1, there is another way to relinquish the old LSP instances after MBB switchover without relying on the arbitrary time intervals set up by the `optimize-switchover-delay` or `optimize-hold-dead-delay` statement. For example, if you use the `optimize-hold-dead-delay` statement, you configure a time you think it is safe to wait before tearing down the old LSP instance after MBB. However, some routes might still be in the process of shifting to the new instance. Tearing down the old LSP instance prematurely results in one of the transit nodes dropping the traffic for those routes that have not shifted to the new LSP instance.

To avoid traffic loss, instead of using the `optimize-switchover-delay` statement, you can use MPLS-OAM (lsp ping), which confirms that the LSP data plane is established end-to-end. Instead of using the `optimize-hold-dead-delay` statement, you can use a feedback mechanism from the rpd infrastructure that confirms that all prefixes referring to the old LSP have been switched over. The feedback mechanism is sourced from the Tag library and relies on the routing protocol process (rpd) infrastructure to determine when all the routes using the old LSP instance have fully shifted to the new LSP instance after MBB switchover.

The feedback mechanism is always in place, and it is optional. Configure the `optimize-adaptive-teardown` statement to have the feedback mechanism used during MBB switchover. This feature is not supported for RSVP point-to-multipoint (P2MP) LSP instances. Global configuration of the `optimize-adaptive-teardown` statement only affects the point-to-point LSPs that are configured in the system.

You only need to configure the `optimize-adaptive-teardown` statement on routers acting as the ingress for the affected LSPs (you do not need to configure this statement on transit or egress routers). This feedback mechanism ensures that old paths are not torn down before all routes have been switched over to the new optimized paths. The global configuration of this configuration statement affects only the point-to-point LSPs that are configured in the system.

```
optimize-adaptive-teardown {
    p2p:
}
```

You can include this statement at the `[edit protocols mpls]` hierarchy level.

Optimizing Signaled LSPs

Once an LSP has been established, topology or resources changes might, over time, make the path suboptimal. A new path might have become available that is less congested, has a lower metric, and traverses fewer hops. You can configure the router to recompute paths periodically to determine whether a more optimal path has become available.

If reoptimization is enabled, an LSP can be rerouted through different paths by constrained-path recomputations. However, if reoptimization is disabled, the LSP has a fixed path and cannot take advantage of newly available network resources. The LSP is fixed until the next topology change breaks the LSP and forces a recomputation.

Reoptimization is not related to failover. A new path is always computed when topology failures occur that disrupt an established path.

Because of the potential system overhead involved, you need to carefully control the frequency of reoptimization. Network stability might suffer when reoptimization is enabled. By default, the `optimize-timer` statement is set to 0 (that is, it is disabled).

LSP optimization is meaningful only when constrained-path LSP computation is enabled, which is the default behavior. For more information about constrained-path LSP computation, see "Disabling Constrained-Path LSP Computation" on page 589. Also, LSP optimization is only applicable to ingress LSPs, so it is only necessary to configure the `optimize-timer` statement on the ingress router. The transit and egress routers require no specific configuration to support LSP optimization (other than to have MPLS enabled).

To enable path reoptimization, include the `optimize-timer` statement:

```
optimize-timer seconds;
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

Once you have configured the `optimize-timer` statement, the reoptimization timer continues its countdown to the configured value even if you delete the `optimize-timer` statement from the configuration. The next optimization uses the new value. You can force the Junos OS to use a new value immediately by deleting the old value, committing the configuration, configuring the new value for the `optimize-timer` statement, and then committing the configuration again.

After reoptimization is run, the result is accepted only if it meets the following criteria:

1. The new path is not higher in IGP metric. (The metric for the old path is updated during computation, so if a recent link metric changed somewhere along the old path, it is accounted for.)
2. If the new path has the same IGP metric, it is not more hops away.

3. The new path does not cause preemption. (This is to reduce the ripple effect of preemption causing more preemption.)

4. The new path does not worsen congestion overall.

   The relative congestion of the new path is determined as follows:

   a. The percentage of available bandwidth on each link traversed by the new path is compared to that for the old path, starting from the most congested links.

   b. For each current (old) path, the software stores the four smallest values for bandwidth availability for the links traversed in ascending order.

   c. The software also stores the four smallest bandwidth availability values for the new path, corresponding to the links traversed in ascending order.

   d. If any of the four new available bandwidth values are smaller than any of the corresponding old bandwidth availability values, the new path has at least one link that is more congested than the link used by the old path. Because using the link would cause more congestion, traffic is not switched to this new path.

   e. If none of the four new available bandwidth values is smaller than the corresponding old bandwidth availability values, the new path is less congested than the old path.

When all the above conditions are met, then:

1. If the new path has a lower IGP metric, it is accepted.

2. If the new path has an equal IGP metric and lower hop count, it is accepted.

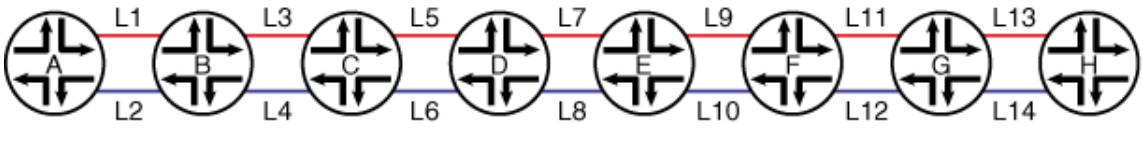
3. If you choose `least-fill` as a load balancing algorithm, LSPs are load balanced as follows:

   a. The LSP is moved to a new path that is utilized at least 10% less than the current path. This might reduce congestion on the current path by only a small amount. For example, if an LSP with 1 MB of bandwidth is moved off a path carrying a minimum of 200 MB, congestion on the original path is reduced by less than 1%.

   b. For `random` or `most-fill` algorithms, this rule does not apply.

   The following example illustrates how the `least-fill` load balancing algorithm works.

Figure 41: least-fill Load Balancing Algorithm Example



As shown in Figure 41 on page 623, there are two potential paths for an LSP to traverse from router A to router H, the odd links from L1 through L13 and the even links from L2 through L14. Currently, the router is using the even links as the active path for the LSP. Each link between the same two routers (for example, router A and router B) has the same bandwidth:

- L1, L2 = 10GE
- L3, L4 = 1GE
- L5, L6 = 1GE
- L7, L8 = 1GE
- L9, L10 = 1GE
- L11, L12 = 10GE
- L13, L14 = 10GE

The 1GE links are more likely to be congested. In this example, the odd 1GE links have the following available bandwidth:

- L3 = 41%
- L5 = 56%
- L7 = 66%
- L9 = 71%

The even 1GE links have the following available bandwidth:

- L4 = 37%
- L6 = 52%
- L8 = 61%
- L10 = 70%

Based on this information, the router would calculate the difference in available bandwidth between the odd links and the even links as follows:

- L4 - L3 = 41% - 37% = 4%
- L6 - L5 = 56% - 52% = 4%
- L8 - L7 = 66% - 61% = 5%
- L10 - L9 = 71% - 70% = 1%

The total additional bandwidth available over the odd links is 14% (4% + 4% + 5% + 1%). Since 14% is greater than 10% (the least-fill algorithm minimum threshold), the LSP is moved to the new path over the odd links from the original path using the even links.

4. Otherwise, the new path is rejected.

You can disable the following reoptimization criteria (a subset of the criteria listed previously):

- If the new path has the same IGP metric, it is not more hops away.
- The new path does not cause preemption. (This is to reduce the ripple effect of preemption causing more preemption.)
- The new path does not worsen congestion overall.
- If the new path has an equal IGP metric and lower hop count, it is accepted.

To disable them, either issue the `clear mpls lsp optimize-aggressive` command or include the `optimize-aggressive` statement:

```
optimize-aggressive;
```

You can include this statement at the following hierarchy levels:

- `[edit protocols mpls]`
- `[edit logical-systems` *`logical-system-name`* `protocols mpls]`

Including the `optimize-aggressive` statement in the configuration causes the reoptimization procedure to be triggered more often. Paths are rerouted more frequently. It also limits the reoptimization algorithm to the IGP metric only.

Configuring the Smart Optimize Timer for LSPs

Because of network and router resource constraints, it is typically inadvisable to configure a short interval for the optimize timer. However, under certain circumstances, it might be desirable to reoptimize a path sooner than would normally be provided by the optimize timer.

For example, an LSP is traversing a preferred path that subsequently fails. The LSP is then switched to a less desirable path to reach the same destination. Even if the original path is quickly restored, it could take an excessively long time for the LSP to use it again, because it has to wait for the optimize timer to reoptimize the network paths. For such situations, you might want to configure the smart optimize timer.

When you enable the smart optimize timer, an LSP is switched back to its original path so long as the original path has been restored within 3 minutes of going down. Also, if the original path goes down again within 60 minutes, the smart optimize timer is disabled, and path optimization behaves as it

normally does when the optimize timer alone is enabled. This prevents the router from using a flapping link.

The smart optimize timer is dependant on other MPLS features to function properly. For the scenario described here in which an LSP is switched to an alternate path in the event of a failure on the original path, it is assumed that you have configured one or more of the MPLS traffic protection features, including fast reroute, link protection, and standby secondary paths. These features help to ensure that traffic can reach its destination in the event of a failure.

At the least, you must configure a standby secondary path for the smart optimize timer feature to work properly. Fast reroute and link protection are more temporary solutions to a network outage. A secondary path ensures that there is a stable alternate path in the event the primary path fails. If you have not configured any sort of traffic protection for an LSP, the smart optimize timer by itself does not ensure that traffic can reach its destination. For more information about MPLS traffic protection, see "MPLS and Traffic Protection" on page 360.

When a primary path fails and the smart optimize timer switches traffic to the secondary path, the router might continue to use the secondary path even after the primary path has been restored. If the ingress router completes a CSPF calculation, it might determine that the secondary path is the better path.

This might be undesirable if the primary path should be the active path and the secondary path should be used as a backup only. Also, if the secondary path is being used as the active path (even though the primary path has been reestablished) and the secondary path fails, the smart optimize timer feature will not automatically switch traffic back to the primary path. However, you can enable protection for the secondary path by configuring node and link protection or an additional standby secondary path, in which case, the smart optimize timer can be effective.

Specify the time in seconds for the smart optimize timer using the `smart-optimize-timer` statement:

NOTE: You can apply the `smart-optimize-timer` configuration statement only if you enable periodic LSP re-optimziation by using the `optimize-timer` statement.

```
smart-optimize-timer seconds;
```

You can include this statement at the following hierarchy levels:

- `[edit protocols mpls]`
- `[edit logical-systems logical-system-name protocols mpls]`

Limiting the Number of Hops in the Path of an RSVP-signaled MPLS LSP

By default, each LSP can traverse a maximum of 255 hops, including the ingress and egress routers. To modify this value, include the `hop-limit` statement:

```
hop-limit number;
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

The number of hops can be from 2 through 255. (A path with two hops consists of the ingress and egress routers only.)

Configuring a Manually-Defined Bandwidth Value for RSVP-signaled MPLS LSPs

Each LSP has a bandwidth value. This value is included in the sender's Tspec field in RSVP path setup messages. You can specify a bandwidth value in bits per second. If you configure more bandwidth for an LSP, it should be able to carry a greater volume of traffic. The default bandwidth is 0 bits per second.

A nonzero bandwidth requires that transit and egress routers reserve capacity along the outbound links for the path. The RSVP reservation scheme is used to reserve this capacity. Any failure in bandwidth reservation (such as failures at RSVP policy control or admission control) might cause the LSP setup to fail. If there is insufficient bandwidth on the interfaces for the transit or egress routers, the LSP is not established.

To specify a bandwidth value for a signaled LSP, include the `bandwidth` statement:

```
bandwidth bps;
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

Automatic Bandwidth Allocation for LSPs

Automatic bandwidth allocation allows an MPLS tunnel to automatically adjust its bandwidth allocation based on the volume of traffic flowing through the tunnel. You can configure an LSP with minimal bandwidth; this feature can dynamically adjust the LSP's bandwidth allocation based on current traffic patterns. The bandwidth adjustments do not interrupt traffic flow through the tunnel.

You set a sampling interval on an LSP configured with automatic bandwidth allocation. The average bandwidth is monitored during this interval. At the end of the interval, an attempt is made to signal a new path for the LSP with the bandwidth allocation set to the maximum average value for the preceding sampling interval. If the new path is successfully established and the original path is removed, the LSP is switched over to the new path. If a new path is not created, the LSP continues to use its current path

until the end of the next sampling interval, when another attempt is made to establish a new path. Note that you can set minimum and maximum bandwidth values for the LSP.

During the automatic bandwidth allocation interval, the router might receive a steady increase in traffic (increasing bandwidth utilization) on an LSP, potentially causing congestion or packet loss. To prevent this, you can define a second trigger to prematurely expire the automatic bandwidth adjustment timer before the end of the current adjustment interval.

Configuring Automatic Bandwidth Allocation for LSPs

Automatic bandwidth allocation allows an MPLS tunnel to automatically adjust its bandwidth allocation based on the volume of traffic flowing through the tunnel. You can configure an LSP with minimal bandwidth, and this feature can dynamically adjust the LSP's bandwidth allocation based on current traffic patterns. The bandwidth adjustments do not interrupt traffic flow through the tunnel.

At the end of the automatic bandwidth allocation time interval, the current maximum average bandwidth usage is compared with the allocated bandwidth for the LSP. If the LSP needs more bandwidth, an attempt is made to set up a new path where bandwidth is equal to the current maximum average usage. If the attempt is successful, the LSP's traffic is routed through the new path and the old path is removed. If the attempt fails, the LSP continues to use its current path.

NOTE: In calculating the value for `Max AvgBW` (relative to the ingress LSP), the sample collected during make before break (MBB) is ignored to prevent inaccurate results. The first sample after a bandwidth adjustment, or after a change in the LSP ID (regardless of path change), is also ignored.

If you have configured link and node protection for the LSP and traffic has been switched to the bypass LSP, the automatic bandwidth allocation feature continues to operate and take bandwidth samples from the bypass LSP. For the first bandwidth adjustment cycle, the maximum average bandwidth usage taken from the original link and node-protected LSP is used to resignal the bypass LSP if more bandwidth is needed. (Link and node protection are not supported on QFX Series switches.)

If you have configured fast-reroute for the LSP, you might not be able to use this feature to adjust the bandwidth. Because the LSPs use a fixed filter (FF) reservation style, when a new path is signaled, the bandwidth might be double-counted. Double-counting can prevent a fast-reroute LSP from ever adjusting its bandwidth when automatic bandwidth allocation is enabled. (Fast reroute is not supported on QFX Series switches.)

To configure automatic bandwidth allocation, complete the steps in the following sections:

NOTE: On the QFX10000 switches, you can only configure automatic bandwidth allocation at the `edit protocols mpls` hierarchy level. Logical systems are not supported.

Configuring Optimized Auto-bandwidth Adjustments for MPLS LSPs

Auto-bandwidth functionality enables the RSVP-TE LSPs, either directly configured or automatically created using auto-mesh, to re-size based on the traffic rate. The traffic rate carried on each LSP is measured by periodically collecting samples of the traffic rate. The frequency of traffic statistics collection is controlled through the `set protocols mpls statistics interval` configuration statement. The re-sizing of the LSPs is called adjustment and the frequency of adjustments is controlled through the `adjust-interval` statement. The minimum configurable value of adjust-interval is one second.

Starting in Junos OS Release 20.4R1, the minimum `adjust-interval` for an `auto-bandwidth` adjustment is decreased to 150 seconds if the `adjust-threshold-overflow-limit` or `adjust-threshold-underflow-limit` statements cross the configured overflow or underflow threshold values.

However, the minimum `adjust-interval` for an `auto-bandwidth` adjustment is 300 seconds if no overflow or underflow sample is detected.

In releases earlier than Junos OS Release 20.4R1, the `adjust-interval` is 300 seconds under overflow or underflow conditions.

With the implementation of auto-bandwidth adjustment optimization, the `auto-bandwidth` decreases the bandwidth of the LSP faster. The ingress label edge router (LER) is able to resize within 150 seconds because of the reduction in `adjust-threshold-overflow-limit`, provided the tear down of an old LSP instance post make-before-break (MBB) is accomplished within 150 seconds.

The requirements for auto-bandwidth optmization are:

- Reduce the probability of LSP route change—This is to reduce the probability of LSP route change when an auto-bandwidth adjustment occurs.
- Reduce the probability of LSP reroute—This is to reduce the probability of the LSP reroute because of the higher priority LSPs that demand the same resource.

In order to fulfil these requirements, the auto-bandwidth adjustments optimization supports the following:

1. **In-place LSP Bandwidth Update**—Enables the ingress label edge router (LER) to re-use the LSP ID when performing bandwidth change on an intra-domain LSP.

   NOTE: In-place LSP bandwidth update is not applicable for an inter-domain LSP.

   In certain scenarios, the LSP route next hop carries the LSP bandwidth either directly or indirectly. Even though in-place LSP bandwidth update is supported in these scenarios, the performance improvement from the functionality is limited because of the LSP route change. That is, because of the change in the inet.3 route table after *auto-bandwidth (MPLS Tunnel)*. For example, performance enhancement is limited when you configure either or both the statements:

- `auto-policing` configured under MPLS.
- The option `bandwidth` under the statement `load-balance` configured under RSVP.

> **NOTE**: In-place LSP bandwidth update through LSP-ID re-use fails and the ingress LER immediately triggers MBB with a new LSP-ID if:
>
> - `no-cspf` is configured for the LSP.
> - LSP is controlled by the Path Computation Element (PCE).
> - LSP optimization timer fires.
> - `clear mpls lsp optimize-aggressive` command is executed.

2. **Per-priority Subscription**—In order to utilize the network resources more efficiently, per-priority subscription enables you to configure a lower RSVP subscription percentage for LSPs of lower priorities and higher RSVP subscription percentage for LSPs of higher priorities.

   For example, instead of setting RSVP subscription percentage as 90% for LSPs for all priorities, you can configure a lower RSVP subscription percentage (say 75%) for LSPs of lower priorities

> **NOTE**: Per-priority subscription does not interoperate with Differentiated Services (DiffServ)-aware traffic engineering (TE). Differentiated Services (DiffServ)-aware traffic engineering offers more flexible and statistical sharing of TE link bandwidth than per-priority subscription.

To Configure In-place LSP Auto-bandwidth Resizing:

1. Configure the device interface to enable MPLS.

```
[edit]
user@host# set interfaces interface-name unit logical-unit-number family mpls
user@host# set protocols mpls interface et-0/0/0:1.0 unit 0 family mpls
```

2. Configure MPLS protocol on the interface.

```
[edit]
user@host# set protocols mpls interfaceinterface-name
user@host# set protocols mpls interface et-0/0/0:1.0
```

3. Configure MPLS and the LSPs and configure link protection for the LSP.

```
[edit]
user@host# set protocols mpls label-switched-path lsp-name to address
```

```
user@host# set protocols mpls label-switched-path lsp1 to 10.2.5.1
```

4. Configure `in-place-bandwidth-update` for the LSP to enable automatic bandwidth LSP resizing.

```
[edit]
user@host# set protocols mpls label-switched-path lsp-name in-place-lsp-bandwidth-update
```

```
user@host# set protocols mpls label-switched-path lsp1 in-place-lsp-bandwidth-update
```

5. Enter commit from the configuration mode.

Verification

From configuration mode, confirm your configuration by entering the, `show protocols show interfaces` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
    interfaces {
        et-0/0/0:1 {
            unit 0 {
                family {
                    mpls;
                }
            }
        }
        }
protocols {
    mpls {
        label-switched-path lsp1 {
            to 10.2.5.1;
            in-place-lsp-bandwidth-update;
        }
```

```
    }
}
```

To Configure Per-priority Subscription:

1. Configure RSVP protocol on the interface.

```
[edit]
user@host# set protocols rsvp interfaceinterface-name
user@host# set protocols rsvp interface et-0/0/0:1.0
```

2. Configure the bandwidth subscription value for the interface. It can be a value from 0 through 65,000 percent. The default subscription value is 100 percent.

```
[edit]
user@host# set protocols rsvp interface interface-name subscription percentage
```

```
user@host# set protocols rsvp et-0/0/0:1.0 subscription 11
```

3. Configure the subscription priority over the interface.

```
[edit]
user@host# set protocols rsvp interface interface-name subscription percentage priority
```

```
user@host# set protocols rsvp et-0/0/0:1.0 subscription 11 priority 7
```

4. Configure the subscription percentage for the priority.

```
[edit]
user@host# set protocols rsvp interface interface-name subscription percentage priority
percentage
```

```
user@host# set protocols rsvp et-0/0/0:1.0 subscription 11 priority 7 percent 10
```

5. Enter commit from the configuration mode.

Verification

From configuration mode, confirm your configuration by entering the, `show protocols show interfaces` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
protocols {
    rsvp {
        interface et-0/0/0:1.0 {
            subscription 11{
            priority 7 {
            percent 10;
        }
    }
}
```

SEE ALSO

> *in-place-lsp-bandwidth-update (Protocols MPLS)*
>
> *subscription*

Configuring Reporting of Automatic Bandwidth Allocation Statistics for LSPs

Automatic bandwidth allocation allows an MPLS tunnel to automatically adjust its bandwidth allocation based on the volume of traffic flowing through the tunnel. You can configure the device to collect statistics related to automatic bandwidth allocation by completing the following steps:

1. To collect statistics related to automatic bandwidth allocation, configure the `auto-bandwidth` option for the `statistics` statement at the `[edit protocols mpls]` hierarchy level. These settings apply to all LSPs configured on the router on which you have also configured the `auto-bandwidth` statement at the `[edit protocols mpls label-switched-path` *`label-switched-path-name`*`]` hierarchy level.

```
statistics {
    auto-bandwidth (MPLS Statistics);
    file filename <files number> <size size> <world-readable | no-world-readable>;
    interval seconds;
    no-transit-statistics;
    transit-statistics-polling;
}
```

2. Specify the *filename* for the files used to store the MPLS trace operation output using the `file` option. All files are placed in the directory **/var/log**. We recommend that you place MPLS tracing output in the file **mpls-log**.
3. Specify the maximum number of trace files using the `files` *number* option. When a trace file named ***trace-file*** reaches its maximum size, it is renamed ***trace-file*.0**, then ***trace-file*.1**, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten.
4. Specify the interval for calculating the average bandwidth usage by configuring a time in seconds using the `interval` option. You can also set the adjustment interval on a specific LSP by configuring the `interval` option at the `[edit protocols mpls label-switch-path` *label-switched-path-name* `statistics]` hierarchy level.

   NOTE: To prevent unnecessary resignaling of LSPs, it is best to configure an LSP adjustment interval that is at least three times longer than the MPLS automatic bandwidth statistics interval. For example, if you configure a value of 30 seconds for the MPLS automatic bandwidth statistics interval (`interval` statement at the `[edit protocols mpls statistics]` hierarchy level), you should configure a value of at least 90 seconds for the LSP adjustment interval (`adjust-interval` statement at the `[edit protocols mpls label-switched-path` *label-switched-path-name* `auto-bandwidth]` hierarchy level).

5. To trace automatic bandwidth allocation, include the `autobw-state flag` for the MPLS `traceoptions` statement at the `[edit protocols mpls]` hierarchy level.

   The following configuration enables the MPLS traceoptions for automatic bandwidth allocation. The trace records are stored in a file called `auto-band-trace` (the filename is user configurable):

   ```
   [edit protocols mpls]
   traceoptions {
       file auto-band-trace size 10k files 10 world-readable;
       flag autobw-state;
   }
   ```

6. Using the `show log` command, you can display the automatic bandwidth allocation statistics file generated when you configure the *auto-bandwidth (MPLS Statistics)* statement. The following shows sample log file output taken from an MPLS statistics file named `auto-band-stats` on a router configured with an LSP named `E-D`. The log file shows that LSP `E-D` is operating over its reserved bandwidth limit initially. Before `Oct 30 17:14:57`, the router triggered an automatic bandwidth adjustment (you might see two sessions for an LSP undergoing an automatic bandwidth adjustment). By `Oct 30 17:16:57`, the

LSP has been reestablished at a higher bandwidth and is now shown using less than 100 percent of its `Reserved Bw` (reserved bandwidth).

```
user@host> show log auto-band-stats
E-D          (LSP ID 5, Tunnel ID 6741)          209 pkt          17094 Byte      1 pps
90 Bps Util 240.01% Reserved Bw       37 Bps
decr nh 0x952c224, type 4, flags 0x0, n_gw 1, nhid 0 to refcount 1Oct 30 17:13:57 Total 1
sessions: 1 success, 0 fail, 0 ignored
E-D          (LSP ID 5, Tunnel ID 6741)          241 pkt          19737 Byte      1 pps
88 Bps Util 234.67% Reserved Bw       37 Bps
decr nh 0x952c224, type 4, flags 0x0, n_gw 1, nhid 0 to refcount 1Oct 30 17:14:27 Total 1
sessions: 1 success, 0 fail, 0 ignored
E-D          (LSP ID 5, Tunnel ID 6741)          276 pkt          22607 Byte      1 pps
95 Bps Util 253.34% Reserved Bw       37 Bps
decr nh 0x952c224, type 4, flags 0x0, n_gw 1, nhid 0 to refcount 1Oct 30 17:14:57 Total 1
sessions: 1 success, 0 fail, 0 ignored
E-D          (LSP ID 5, Tunnel ID 6741)            0 pkt              0 Byte      0
pps        0 Bps Util  0.00% Reserved Bw       37 Bps
E-D          (LSP ID 6, Tunnel ID 6741)            0 pkt              0 Byte      0
pps        0 Bps Util  0.00% Reserved Bw      101 Bps
decr nh 0x952c224, type 4, flags 0x0, n_gw 1, nhid 0 to refcount 1decr nh 0x952c308, type 4,
flags 0x0, n_gw 1, nhid 0 to refcount 1Oct 30 17:15:27 Total 2 sessions: 2 success, 0 fail, 0
ignored
E-D          (LSP ID 5, Tunnel ID 6741)            0 pkt              0 Byte      0
pps        0 Bps Util  0.00% Reserved Bw       37 Bps
E-D          (LSP ID 6, Tunnel ID 6741)           33 pkt           2695 Byte      1 pps
89 Bps Util 87.69% Reserved Bw      101 Bps
decr nh 0x952c224, type 4, flags 0x0, n_gw 1, nhid 0 to refcount 1decr nh 0x952c308, type 4,
flags 0x0, n_gw 1, nhid 0 to refcount 1Oct 30 17:15:57 Total 2 sessions: 2 success, 0 fail, 0
ignored
E-D          (LSP ID 5, Tunnel ID 6741)            0 pkt              0 Byte      0
pps        0 Bps Util  0.00% Reserved Bw       37 Bps
E-D          (LSP ID 6, Tunnel ID 6741)           65 pkt           5338 Byte      1 pps
88 Bps Util 86.70% Reserved Bw      101 Bps
decr nh 0x952c224, type 4, flags 0x0, n_gw 1, nhid 0 to refcount 1decr nh 0x952c308, type 4,
flags 0x0, n_gw 1, nhid 0 to refcount 1Oct 30 17:16:27 Total 2 sessions: 2 success, 0 fail, 0
ignored
E-D          (LSP ID 6, Tunnel ID 6741)           97 pkt           7981 Byte      1 pps
88 Bps Util 86.70% Reserved Bw      101 Bps
```

```
decr nh 0x952c308, type 4, flags 0x0, n_gw 1, nhid 0 to refcount 1Oct 30 17:16:57 Total 1
sessions: 1 success, 0 fail, 0 ignored
```

7. Issue the *show mpls lsp autobandwidth* command to display current information about automatic bandwidth allocation. The following shows sample output from the `show mpls lsp autobandwidth` command taken at about the same time as the log file shown previously:

```
user@host> show mpls lsp autobandwidth
Lspname               Last         Requested    Reserved     Highwater    AdjustTime
LastAdjust
                      BW           BW           BW           mark         Left
(sec)
E-D                   300bps      812.005bps  812bps      1.56801kbps 294 sec    Wed Oct 30
17:15:26 2013
```

8. Issue the `file show` command to display the MPLS trace file. You need to specify the file location and file name (the file is located in **/var/log/**. The following shows sample trace file output is taken from an MPLS trace file named `auto-band-trace.0.gz` on a router configured with an LSP named `E-D`. The trace file shows that LSP `E-D` is operating over its reserved bandwidth limit initially. At `Oct 30 17:15:26`, the router triggers an automatic bandwidth adjustment (you might see two sessions for an LSP undergoing an automatic bandwidth adjustment). By `Oct 30 17:15:57`, the LSP has been reestablished at a higher bandwidth and is now shown using less than 100 percent of its `Reserved Bw` (reserved bandwidth).

```
user@host> file show /var/log/auto-band-trace.0.gz
Oct 30 17:13:57 trace_on: Tracing to "/var/log/E/auto-band-trace" started
Oct 30 17:13:57.466825 LSP E-D (id 5) new bytes arrived           2714 in 29 sec
Oct 30 17:14:27.466713 E-D          (LSP ID 5, Tunnel ID 6741)          241 pkt
19737 Byte      1 pps       88 Bps Util 234.67% Reserved Bw       37 Bps
Oct 30 17:14:27.466962 LSP E-D (id 5, old id 5); sampled bytes      19737 > bytes
recorded      17094
Oct 30 17:14:27.467035 LSP E-D (id 5) new bytes arrived           2643 in 29 sec
Oct 30 17:14:57.466599 E-D          (LSP ID 5, Tunnel ID 6741)          276 pkt
22607 Byte      1 pps       95 Bps Util 253.34% Reserved Bw       37 Bps
Oct 30 17:14:57.466758 LSP E-D (id 5, old id 5); sampled bytes      22607 > bytes
recorded      19737
Oct 30 17:14:57.466825 LSP E-D (id 5) new bytes arrived           2870 in 29 sec
Oct 30 17:15:26.265816 Adjust Autobw: LSP E-D (id 5) curr adj bw 300bps updated with
812.005bps
Oct 30 17:15:26.266064 mpls LSP E-D Autobw change 512.005bps >= threshold 75bps
Oct 30 17:15:26.363372 Autobw Success: LSP E-D ()  (old id 5 new id 6) update prev active bw
300 bps with 812 bps
```

```
Oct 30 17:15:26.363686 RPD_MPLS_PATH_BANDWIDTH_CHANGE: MPLS path  (lsp E-D) bandwidth
changed, path bandwidth 812 bps
Oct 30 17:15:27.364751 RPD_MPLS_LSP_BANDWIDTH_CHANGE: MPLS LSP E-D bandwidth changed, lsp
bandwidth 812 bps
Oct 30 17:15:27.466849 E-D          (LSP ID 5, Tunnel ID 6741)            0 pkt
0 Byte      0 pps        0 Bps Util  0.00% Reserved Bw       37 Bps
Oct 30 17:15:27.467050 E-D          (LSP ID 6, Tunnel ID 6741)            0 pkt
0 Byte      0 pps        0 Bps Util  0.00% Reserved Bw      101 Bps
Oct 30 17:15:57.466858 E-D          (LSP ID 5, Tunnel ID 6741)            0 pkt
0 Byte      0 pps        0 Bps Util  0.00% Reserved Bw       37 Bps
Oct 30 17:15:57.467106 E-D          (LSP ID 6, Tunnel ID 6741)           33 pkt
2695 Byte      1 pps       89 Bps Util 87.69% Reserved Bw      101 Bps
Oct 30 17:15:57.467201 LSP E-D (id 6, old id 5); LSP up after autobw adjustment and active
for 30 sec
Oct 30 17:15:57.467398 LSP E-D (id 6) psb bytes           2695 < bytes recorded
22607 total bytes           2695 in 30 sec
Oct 30 17:15:57.467461 First sample of the adjust interval after automatic bw adjustment
Oct 30 17:15:57.467594 Update curr max avg bw 0bps of LSP E-D with new bw 716.225bps
Oct 30 17:16:27.466830 E-D          (LSP ID 5, Tunnel ID 6741)            0 pkt
0 Byte      0 pps        0 Bps Util  0.00% Reserved Bw       37 Bps
Oct 30 17:16:27.467079 E-D          (LSP ID 6, Tunnel ID 6741)           65 pkt
5338 Byte      1 pps       88 Bps Util 86.70% Reserved Bw      101 Bps
Oct 30 17:16:27.467171 LSP E-D (id 6, old id 6); sampled bytes       5338 > bytes
recorded       2695
Oct 30 17:16:27.467237 LSP E-D (id 6) new bytes arrived           2643 in 29 sec
Oct 30 17:16:57.466712 E-D          (LSP ID 6, Tunnel ID 6741)           97 pkt
7981 Byte      1 pps       88 Bps Util 86.70% Reserved Bw      101 Bps
Oct 30 17:16:57.466870 LSP E-D (id 6, old id 6); sampled bytes       7981 > bytes
recorded       5338
```

Using Polling Profiles for Automatic Bandwidth Allocation for RSVP LSPs

SUMMARY

Optimize bandwidth allocation for RSVP label-switched paths (LSPs) based on traffic demands using MPLS Automatic Bandwidth polling profiles.

IN THIS SECTION

- Benefits of Using Polling Profiles for MPLS Auto-Bandwidth | **638**
- Configuration | **638**

You can use polling profiles to optimize bandwidth allocation for RSVP label-switched paths (LSPs) based on traffic demands. Use polling profiles to configure distinct polling intervals per LSP, enabling precise bandwidth adjustments tailored to varying service requirements. This flexibility supports differentiated service levels, allowing for both aggressive and conservative bandwidth tuning. You can further refine control with independent overflow and underflow thresholds through the `adjust-threshold-options` configuration. The feature also integrates comprehensive statistics capture, enhancing your ability to monitor performance and make informed network configuration decisions.

Benefits of Using Polling Profiles for MPLS Auto-Bandwidth

- Tailor bandwidth adjustments to specific service needs through configurable polling profiles, enhancing service differentiation.
- Achieve granular control over bandwidth management with independent settings for overflow and underflow thresholds, ensuring precise resource allocation.
- Enhance network stability by supporting different polling intervals, which can be adjusted to suit the traffic demands of each LSP.
- Improve monitoring and decision-making with detailed statistics capture, providing better insight into bandwidth usage patterns and adjustments.
- Maintain compatibility with existing configurations, ensuring seamless integration and operation within current network setups.

Configuration

You can configure polling profile with a specific polling interval and apply to an auto-bandwidth LSP.

To configure polling profile, include the `polling-profile` CLI statement at the [`edit protocols mpls statistics auto-bandwidth`] hierarchy level. You must also configure a polling interval to the polling profile as follows:

```
[edit protocol mpls]
user@host# set statistics auto-bandwidth polling-profile profile-name interval sec
```

When an auto-bandwidth LSP is configured, by default the auto-bandwidth LSP uses the default global statistics polling interval configured as follows:

```
[edit protocol mpls]
user@host# set statistics interval sec
```

You can associate the polling profile with an LSP. This ensures that each LSP is monitored and adjusted according to its designated interval, which enhances service differentiation and stability. Use the following configuration statement to associate the polling profile with an LSP:

```
[edit protocols mpls]
user@host# set label-switched-path lsp-name auto-bandwidth polling-profile profile-name
```

When the polling profile is associated with an LSP, the configuration restriction on overflow and underflow statements are relaxed only for those LSPs. This allows for aggressive auto-bandwidth adjustments. Other LSPs which do not need aggressive adjustments do not require any changes in configuration.

Additionally, the `adjust-threshold-options` statement under the [`edit protocols mpls label-switched-path` *`lsp-name`* `auto-bandwidth`] hierarchy provides a structured approach to managing bandwidth adjustments.

For precise bandwidth adjustments, you can set independent thresholds for overflow and underflow conditions. The `overflow-threshold` *`percent`* and `underflow-threshold` *`percent`* statements under the [`edit protocols mpls label-switched-path` *`lsp-name`* `auto-bandwidth adjust-threshold-options`] hierarchy enables you to specify the bandwidth percentage that triggers these adjustments. This granular control ensures that your network responds appropriately to fluctuating traffic patterns, maintaining operational efficiency and avoiding unnecessary bandwidth changes. When evaluating whether a statistics sample is a valid overflow or underflow, the sample is compared against the global threshold percentage configured at [`edit protocols mpls label-switched-path` *`lsp-name`* `auto-bandwidth adjust-threshold-options threshold` *`percent`*] hierarchy level.

Furthermore, the integration of comprehensive statistics capture enhances your ability to monitor and analyze bandwidth usage patterns. You can use the `show mpls lsp autobandwidth detail` and `show mpls lsp extensive` commands to view the polling profiles, their intervals, as well as adjustments made due to overflow or underflow conditions. This detailed information supports informed decision-making, allowing you to optimize the configuration of your network to meet current and future demands. These enhancements, maintaining backward compatibility, ensure seamless integration into existing setups without disrupting current operations.

Configuring an RSVP-signaled MPLS LSP to Calculate a Path Across Autonomous Systems

You can configure an LSP to traverse multiple areas in a network by including the `inter-domain` statement as a part of the LSP configuration. This statement allows the router to search for routes in the IGP database. You need to configure this statement on routers that might be unable to locate a path using intra-domain CSPF (by looking in the traffic engineering database (TED)). When you configure inter-area LSPs, the `inter-domain` statement is required.

Before you begin:

- Configure the device interfaces with family MPLS.
- Configure the device router ID and autonomous system number.
- Enable MPLS and RSVP on the router and transit interfaces.
- Configure your IGP to support traffic engineering.
- Set up an LSP from the ingress to the egress router.

To configure an LSP across multiple ASs on the ingress label-switched router (LER):

1. Enable MPLS on all the interfaces (excluding the management interface).

```
[edit protocols]
user@LER# set mpls interface all
user@LER# set mpls interface fxp0.0 disable
```

2. Enable RSVP on all the interfaces (excluding the management interface).

```
[edit protocols]
user@LER# set rsvp interface all
user@LER# set rsvp interface fxp0.0 disable
```

3. Configure the inter-area LSP.

```
[edit protocols]
user@LER# set mpls label-switched-path inter-area-LSP-name to egress-LER-ip-address
user@LER# set mpls label-switched-path inter-area-LSP-name inter-domain
```

4. Verify and commit the configuration.

```
[edit protocols]
user@LER# set rsvp interface ge-0/0/0.0
user@LER# set rsvp interface lo0.0
user@LER# set rsvp interface fxp0.0 disable
user@LER# set mpls statistics traffic-class-statistics
user@LER# set mpls label-switched-path R1-R2 to 20.0.0.1
user@LER# set mpls label-switched-path R1-R2 inter-domain
user@LER# set mpls interface ge-0/0/0.0
user@LER# set mpls interface lo0.0
user@LER# set mpls interface fxp0.0 disable
user@LER# set ospf traffic-engineering
```

```
user@LER# set ospf area 0.0.0.0 interface ge-0/0/0.0
user@LER# set ospf area 0.0.0.0 interface lo0.0
```

Damping IGP Advertisements of RSVP-signaled MPLS LSP State Changes

When an LSP changes from being up to being down, or from down to up, this transition takes effect immediately in the router software and hardware. However, when advertising LSPs into IS-IS and OSPF, you may want to damp LSP transitions, thereby not advertising the transition until a certain period of time has transpired (known as the hold time). In this case, if the LSP goes from up to down, the LSP is not advertised as being down until it has remained down for the hold-time period. Transitions from down to up are advertised into IS-IS and OSPF immediately. Note that LSP damping affects only the IS-IS and OSPF advertisements of the LSP; other routing software and hardware react immediately to LSP transitions.

To damp LSP transitions, include the `advertisement-hold-time` statement:

```
advertisement-hold-time seconds;
```

seconds can be a value from 0 through 65,535 seconds. The default is 5 seconds.

You can include this statement at the following hierarchy levels:

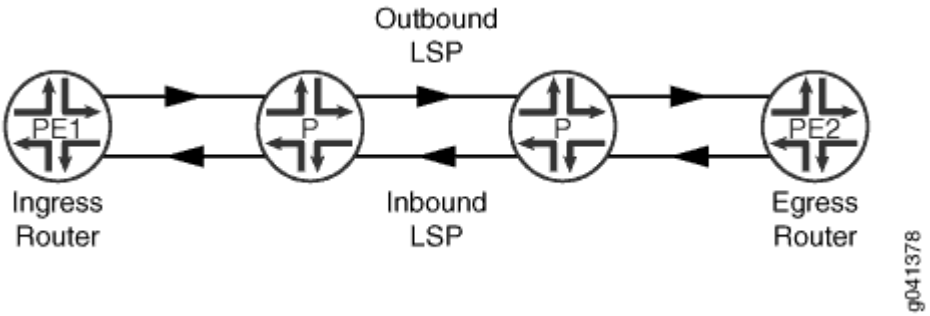
- `[edit protocols mpls]`
- `[edit logical-systems logical-system-name protocols mpls]`

Configuring Corouted Bidirectional RSVP-signaled MPLS LSPs

A corouted bidirectional packet LSP is a combination of two LSPs sharing the same path between a pair of ingress and egress nodes, as shown in Figure 42 on page 642. It is established using the GMPLS extensions to RSVP-TE. This type of LSP can be used to carry any of the standard types of MPLS-based traffic, including Layer 2 VPNs, Layer 2 circuits, and Layer 3 VPNs. You can configure a single BFD session for the bidirectional LSP (you do not need to configure a BFD session for each LSP in each direction). You can also configure a single standby bidirectional LSP to provide a backup for the primary bidirectional LSP. Corouted bidirectional LSPs are supported for both penultimate hop popping (PHP) and ultimate hop popping (UHP).

High availability is available for bidirectional LSPs. You can enable graceful restart and nonstop active routing. Graceful restart and nonstop active routing are supported when the restarting router is the ingress, egress, or transit router for the bidirectional LSP.

Figure 42: Corouted Bidirectional LSP



To configure a corouted bidirectional LSP:

1. In configuration mode, configure the ingress router for the LSP and include the `corouted-bidirectional` statement to specify that the LSP be established as a corouted bidirectional LSP.

   The path is computed using CSPF and initiated using RSVP signaling (just like a unidirectional RSVP signaled LSP). Both the path to the egress router and the reverse path from the egress router are created when this configuration is committed.

   ```
   [edit protocols mpls]
   user@PE1# set label-switched-path sample-lsp corouted-bidirectional
   ```

2. (Optional) For a reverse path, configure an LSP on the egress router and include the `corouted-bidirectional-passive` statement to associate the LSP with another LSP.

   No path computation or signaling is used for this LSP since it relies on the path computation and signaling provided by the ingress LSP. You cannot configure both the `corouted-bidirectional` statement and the `corouted-bidirectional-passive` statement on the same LSP.

   ```
   [edit protocols mpls]
   user@PE1# set label-switched-path sample-lsp-reverse-path corouted-bidirectional-passive
   ```

   This statement also makes it easier to debug corouted bidirectional LSPs. If you configure the `corouted-bidirectional-passive` statement (again, on the egress router), you can issue `ping mpls lsp-end-point`, `ping mpls ldp`, `ping mpls rsvp`, `traceroute mpls ldp`, and `traceroute mpls rsvp` commands to test the corouted bidirectional LSP from the egress router.

3. Use the `show mpls lsp extensive` and the `show rsvp session extensive` commands to display information about the bidirectional LSP.

The following shows output for the `show rsvp session extensive` command when run on an ingress router with a bidirectional LSP configured:

```
user@PE1> show rsvp session extensive
Ingress RSVP: 2 sessions

10.255.14.39
  From: 10.255.14.43, LSPstate: Up, ActiveRoute: 0
  LSPname: l-to-h, LSPpath: Primary
  LSPtype: Static Configured
  Bidirectional, Upstream label in: 3, Upstream label out: -
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 300032
  Resv style: 1 FF, Label in: -, Label out: 300032
  Time left:    -, Since: Tue May 31 08:49:25 2011
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 24617 protocol 0
  PATH rcvfrom: localclient
  Adspec: sent MTU 1500
  Path MTU: received 1500
  PATH sentto: 10.1.1.2 (ge-0/0/0.0) 3396 pkts
  RESV rcvfrom: 10.1.1.2 (ge-0/0/0.0) 3394 pkts
  PATH notifyto: localclient
  RESV notifyto: 10.255.14.39
  Protection attributes: primary, working, 1:N protection
  Association attributes: recovery, src 10.255.14.43, id 1
  Explct route: 10.1.1.2 10.1.2.2 10.1.3.2
  Record route: 10.1.1.2 10.1.2.2 10.1.3.2

10.255.14.39
  From: 10.255.14.43, LSPstate: Up, ActiveRoute: 0
  LSPname: l-to-h, LSPpath: Secondary
  LSPtype: Static Configured
  Bidirectional, Upstream label in: 3, Upstream label out: -
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 300032
  Resv style: 1 FF, Label in: -, Label out: 300032
  Time left:    -, Since: Tue May 31 08:49:25 2011
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 2 receiver 24617 protocol 0
  PATH rcvfrom: localclient
  Adspec: sent MTU 1500
```

```
  Path MTU: received 1500
  PATH sentto: 10.1.1.2 (ge-0/0/0.0) 3396 pkts
  RESV rcvfrom: 10.1.1.2 (ge-0/0/0.0) 3394 pkts
  Protection attributes: primary, protecting
  Association attributes: recovery, src 10.255.14.43, id 1
  Explct route: 10.2.1.2 10.2.2.2 10.2.3.2
  Record route: 10.2.1.2 10.2.2.2 10.2.3.2
Total 2 displayed, Up 2, Down 0

Egress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Configuring the Entropy Label for LSPs

The insertion of entropy labels for an LSP enables transit routers to load-balance MPLS traffic across ECMP paths or Link Aggregation groups using just the MPLS label stack as a hash input without having to rely on deep packet inspection. Deep packet inspection requires more of the router's processing power and different routers have differing deep-packet inspection capabilities.

To configure the entropy label for an LSP, complete the following steps:

1. On the ingress router, include the `entropy-label` statement at the `[edit protocols mpls labeled-switched-path` *`labeled-switched-path-name`*`]` hierarchy level or at the `[edit protocols mpls static-labeled-switched-path` *`labeled-switched-path-name`* `ingress]` hierarchy level. The entropy label is added to the MPLS label stack and can be processed in the forwarding plane.

```
entropy-label;
```

NOTE: This is only applicable for RSVP and static LSPs.

2. On the ingress router, you can configure an ingress policy for LDP-signaled LSPs:

```
entropy-label {
    ingress-policy policy-name;
}
```

Configure the ingress policy at the `[edit policy-options]` hierarchy level:

```
policy-options {
    policy-statement policy-name {
        term term-name {
            from {
                prefix-list prefix-list-name;
            }
            then actions;
        }
    }
}
```

The following shows an example of an entropy label ingress policy.

```
policy-options {
    policy-statement entropy-policy {
        term no-insert-entropy-label {
            from {
                prefix-list no-entropy-label-fec;
            }
            then accept;
        }
    }
}
```

3. (Optional) By default, routers that support the pushing and popping of entropy labels are configured with the `load-balance-label-capability` statement at the `[edit forwarding-options]` hierarchy level to signal the labels on a per-LSP basis. If the peer router is not equipped to handle load-balancing labels, you can prevent the provider edge (PE) router from signaling the entropy label capability by configuring the `no-load-balance-label-capability` statement at the `[edit forwarding-options]` hierarchy level.

```
[edit forwarding-options]
user@PE no-load-balance-label-capability;
```

Transit routers require no configuration. The presence of the entropy label indicates to the transit router to load balance based solely on the MPLS label stack.

Penultimate hop routers pop the entropy label by default.

Example: Configuring an Entropy Label for a BGP Labeled Unicast LSP

IN THIS SECTION

- Requirements | 646
- Overview | 647
- Configuration | 648
- Verification | 663

This example shows how to configure an entropy label for a BGP labeled unicast to achieve end-to-end load balancing using entropy labels. When an IP packet has multiple paths to reach its destination, Junos OS uses certain fields of the packet headers to hash the packet to a deterministic path. This requires an entropy label, a special load-balancing label that can carry the flow information. LSRs in the core simply use the entropy label as the key to hash the packet to the correct path. An entropy label can be any label value between 16 to 1048575 (regular 20-bit label range). Since this range overlaps with the existing regular label range, a special label called entropy label indicator (ELI) is inserted before the entropy label. ELI is a special label assigned by IANA with the value of 7.

BGP labeled unicasts generally concatenate RSVP or LDP LSPs across multiple IGP areas or multiple autonomous systems. RSVP or LDP entropy labels are popped at the penultimate hop node, together with the RSVP or LDP label. This feature enables the use of entropy labels at the stitching points to bridge the gap between the penultimate hop node and the stitching point, in order to achieve end-to-end entropy label load balancing for BGP traffic.

Requirements

This example uses the following hardware and software components:

- Seven MX Series routers with MPCs
- Junos OS Release 15.1 or later running on all the devices
  - Revalidated using Junos OS Relese 22.4

Before you configure an entropy label for BGP labeled unicast, make sure you:

1. Configure the device interfaces.
2. Configure OSPF or any other IGP protocol.
3. Configure BGP.

4. Configure RSVP.

5. Configure MPLS.

Overview

IN THIS SECTION

- Topology | 648

When BGP labeled unicasts concatenate RSVP or LDP LSPs across multiple IGP areas or multiple autonomous systems, RSVP or LDP entropy labels are popped at the penultimate hop node, together with the RSVP or LDP label. However, there are no entropy labels at the stitching points, that is, the routers between two areas. Therefore, the routers at the stitching points used the BGP labels to forward packets.

Beginning with Junos OS Release 15.1, you can configure an entropy label for BGP labeled unicast to achieve end-to-end entropy label load balancing. This feature enables the use of an entropy label at the stitching points in order to achieve end-to-end entropy label load balancing for BGP traffic. Junos OS allows the insertion of entropy labels at the BGP labeled unicast LSP ingress.

By default, routers that support entropy labels are configured with the `load-balance-label-capability` statement at the `[edit forwarding-options]` hierarchy level to signal the labels on a per-LSP basis. If the peer router is not equipped to handle load-balancing labels, you can prevent the signaling of entropy label capability by configuring the `no-load-balance-label-capability` at the `[edit forwarding-options]` hierarchy level.

```
[edit forwarding-options]
user@PE# no-load-balance-label-capability
```

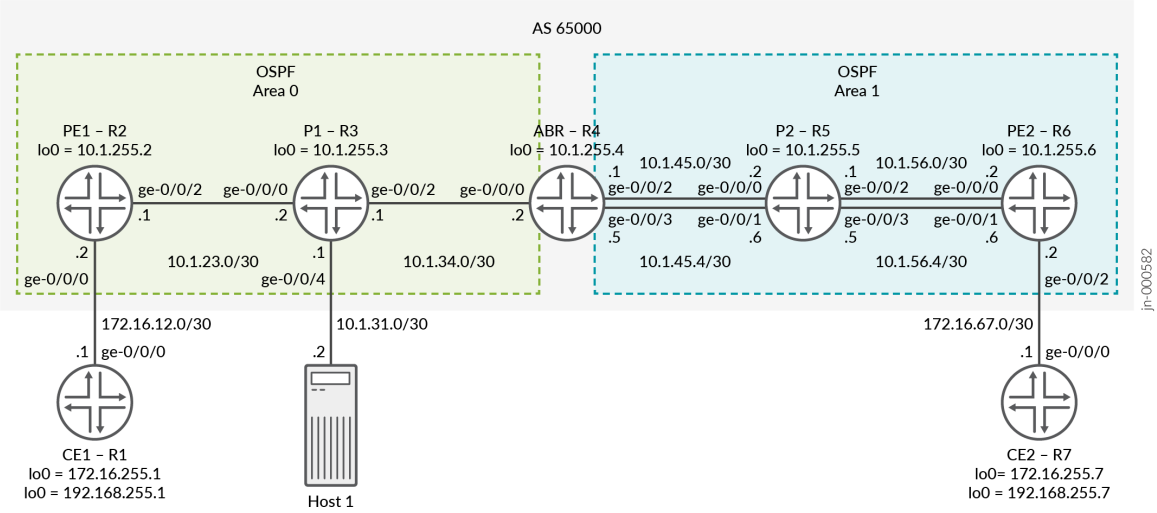
NOTE: You can explicitly disable advertising entropy label capability at egress for routes specified in the policy with the `no-entropy-label-capability` option at the `[edit policy-options policy-statement` *`policy name`* `then]` hierarchy level.

```
[edit policy-options policy-statement policy-name then]
user@PE# no-entropy-label-capability
```

Topology

In Figure 43 on page 648 , Router PE1 is the ingress router and Router PE2 is the egress router. Routers P1 and P2 are the transit routers. Router ABR is the area bridge router between Area 0 and Area 1. Two LSPs are configured on the ABR to PE2 for load balancing the traffic. Entropy label capability for BGP labeled unicast is enabled on the ingress Router PE1. Host 1 is connected to P1 for packet captures so that we can show the entropy label.

Figure 43: Configuring an Entropy Label for BGP Labeled Unicast



Configuration

IN THIS SECTION

- CLI Quick Configuration | **649**
- Configuring Router PE1 | **654**
- Configuring Router P1 | **657**
- Configuring Router ABR | **659**
- (Optional) Port-Mirroring Configuration | **661**

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter `commit` from configuration mode.

Router CE1

```
set interfaces ge-0/0/0 unit 0 family inet address 172.16.12.1/30
set interfaces lo0 unit 0 family inet address 172.16.255.1/32 primary
set interfaces lo0 unit 0 family inet address 192.168.255.1/32
set routing-options router-id 172.16.255.1
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

Router PE1

```
set interfaces ge-0/0/0 unit 0 family inet address 172.16.12.2/30
set interfaces ge-0/0/2 unit 0 family inet address 10.1.23.1/30
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.1.255.2/32 primary
set interfaces lo0 unit 1 family inet address 10.1.255.22/32
set policy-options policy-statement bgp-to-ospf from protocol bgp
set policy-options policy-statement bgp-to-ospf then accept
set policy-options policy-statement pplb then load-balance per-packet
set routing-instances VPN-l3vpn instance-type vrf
set routing-instances VPN-l3vpn protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set routing-instances VPN-l3vpn protocols ospf area 0.0.0.0 interface lo0.1 passive
set routing-instances VPN-l3vpn protocols ospf export bgp-to-ospf
set routing-instances VPN-l3vpn interface ge-0/0/0.0
set routing-instances VPN-l3vpn interface lo0.1
set routing-instances VPN-l3vpn route-distinguisher 10.1.255.2:1
set routing-instances VPN-l3vpn vrf-target target:65000:1
set routing-options router-id 10.1.255.2
set routing-options autonomous-system 65000
set routing-options forwarding-table export pplb
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.1.255.2
set protocols bgp group ibgp family inet labeled-unicast entropy-label
set protocols bgp group ibgp neighbor 10.1.255.4 family inet labeled-unicast rib inet.3
set protocols bgp group ibgp neighbor 10.1.255.6 family inet-vpn unicast
set protocols mpls icmp-tunneling
```

```
set protocols mpls label-switched-path pe1-abr to 10.1.255.4
set protocols mpls label-switched-path pe1-abr entropy-label
set protocols mpls interface ge-0/0/2.0
set protocols mpls interface lo0.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols rsvp interface ge-0/0/2.0
set protocols rsvp interface lo0.0
```

Router P1

```
set interfaces ge-0/0/0 unit 0 family inet address 10.1.23.2/30
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 10.1.34.1/30
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.1.255.3/32 primary
set routing-options router-id 10.1.255.3
set protocols mpls icmp-tunneling
set protocols mpls interface ge-0/0/0.0
set protocols mpls interface lo0.0
set protocols mpls interface ge-0/0/2.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols rsvp interface ge-0/0/0.0
set protocols rsvp interface lo0.0
set protocols rsvp interface ge-0/0/2.0
```

Router ABR

```
set interfaces ge-0/0/0 unit 0 family inet address 10.1.34.2/30
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 10.1.45.1/30
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 family inet address 10.1.45.5/30
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.1.255.4/32 primary
set forwarding-options hash-key family mpls label-1
set forwarding-options hash-key family mpls label-2
```

```
set forwarding-options hash-key family mpls label-3
set forwarding-options enhanced-hash-key family mpls no-payload
set policy-options policy-statement pplb then load-balance per-packet
set policy-options policy-statement send-inet3-pe1 from route-filter 10.1.255.2/32 exact
set policy-options policy-statement send-inet3-pe1 then accept
set policy-options policy-statement send-inet3-pe2 from route-filter 10.1.255.6/32 exact
set policy-options policy-statement send-inet3-pe2 then accept
set routing-options router-id 10.1.255.4
set routing-options autonomous-system 65000
set routing-options forwarding-table export pplb
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.1.255.4
set protocols bgp group ibgp family inet labeled-unicast rib inet.3
set protocols bgp group ibgp neighbor 10.1.255.2 export send-inet3-pe2
set protocols bgp group ibgp neighbor 10.1.255.6 export send-inet3-pe1
set protocols mpls icmp-tunneling
set protocols mpls label-switched-path abr-pe1 to 10.1.255.2
set protocols mpls label-switched-path abr-pe1 entropy-label
set protocols mpls label-switched-path abr-pe2 to 10.1.255.6
set protocols mpls label-switched-path abr-pe2 entropy-label
set protocols mpls label-switched-path abr-pe2 primary to-r6-1
set protocols mpls label-switched-path abr-pe2-2 to 10.1.255.6
set protocols mpls label-switched-path abr-pe2-2 entropy-label
set protocols mpls label-switched-path abr-pe2-2 primary to-r6-2
set protocols mpls path to-r6-1 10.1.45.2 strict
set protocols mpls path to-r6-1 10.1.56.2 strict
set protocols mpls path to-r6-2 10.1.45.6 strict
set protocols mpls path to-r6-2 10.1.56.6 strict
set protocols mpls interface lo0.0
set protocols mpls interface ge-0/0/0.0
set protocols mpls interface ge-0/0/2.0
set protocols mpls interface ge-0/0/3.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.1 interface ge-0/0/2.0
set protocols ospf area 0.0.0.1 interface ge-0/0/3.0
set protocols rsvp interface lo0.0
set protocols rsvp interface ge-0/0/0.0
set protocols rsvp interface ge-0/0/2.0
set protocols rsvp interface ge-0/0/3.0
```

Router P2

```
set interfaces ge-0/0/0 unit 0 family inet address 10.1.45.2/30
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 10.1.45.6/30
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 10.1.56.1/30
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 family inet address 10.1.56.5/30
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.1.255.5/32 primary
set forwarding-options hash-key family mpls label-1
set forwarding-options hash-key family mpls label-2
set forwarding-options hash-key family mpls label-3
set forwarding-options enhanced-hash-key family mpls no-payload
set policy-options policy-statement pplb then load-balance per-packet
set routing-options router-id 10.1.255.5
set routing-options forwarding-table export pplb
set protocols mpls icmp-tunneling
set protocols mpls interface ge-0/0/2.0
set protocols mpls interface lo0.0
set protocols mpls interface ge-0/0/0.0
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/3.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.1 interface lo0.0 passive
set protocols ospf area 0.0.0.1 interface ge-0/0/2.0
set protocols ospf area 0.0.0.1 interface ge-0/0/0.0
set protocols ospf area 0.0.0.1 interface ge-0/0/1.0
set protocols ospf area 0.0.0.1 interface ge-0/0/3.0
set protocols rsvp interface ge-0/0/2.0
set protocols rsvp interface lo0.0
set protocols rsvp interface ge-0/0/0.0
set protocols rsvp interface ge-0/0/1.0
set protocols rsvp interface ge-0/0/3.0
```

Router PE2

```
set interfaces ge-0/0/0 unit 0 family inet address 10.1.56.2/30
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 10.1.56.6/30
```

```
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 172.16.67.2/30
set interfaces lo0 unit 0 family inet address 10.1.255.6/32 primary
set interfaces lo0 unit 1 family inet address 10.1.255.66/32
set forwarding-options hash-key family mpls label-1
set forwarding-options hash-key family mpls label-2
set forwarding-options hash-key family mpls label-3
set forwarding-options enhanced-hash-key family mpls no-payload
set policy-options policy-statement bgp-to-ospf from protocol bgp
set policy-options policy-statement bgp-to-ospf then accept
set policy-options policy-statement pplb then load-balance per-packet
set routing-instances VPN-l3vpn instance-type vrf
set routing-instances VPN-l3vpn protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set routing-instances VPN-l3vpn protocols ospf area 0.0.0.0 interface lo0.1 passive
set routing-instances VPN-l3vpn protocols ospf export bgp-to-ospf
set routing-instances VPN-l3vpn interface ge-0/0/2.0
set routing-instances VPN-l3vpn interface lo0.1
set routing-instances VPN-l3vpn route-distinguisher 10.1.255.6:1
set routing-instances VPN-l3vpn vrf-target target:65000:1
set routing-options router-id 10.1.255.6
set routing-options autonomous-system 65000
set routing-options forwarding-table export pplb
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.1.255.6
set protocols bgp group ibgp family inet labeled-unicast entropy-label
set protocols bgp group ibgp neighbor 10.1.255.4 family inet labeled-unicast rib inet.3
set protocols bgp group ibgp neighbor 10.1.255.2 family inet-vpn unicast
set protocols mpls icmp-tunneling
set protocols mpls label-switched-path pe2-abr to 10.1.255.4
set protocols mpls label-switched-path pe2-abr entropy-label
set protocols mpls interface ge-0/0/0.0
set protocols mpls interface lo0.0
set protocols mpls interface ge-0/0/1.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.1 interface ge-0/0/0.0
set protocols ospf area 0.0.0.1 interface lo0.0 passive
set protocols ospf area 0.0.0.1 interface ge-0/0/1.0
set protocols rsvp interface ge-0/0/0.0
set protocols rsvp interface lo0.0
set protocols rsvp interface ge-0/0/1.0
```

Router CE2

```
set interfaces ge-0/0/0 unit 0 family inet address 172.16.67.1/30
set interfaces lo0 unit 0 family inet address 172.16.255.7/32 primary
set interfaces lo0 unit 0 family inet address 192.168.255.7/32
set routing-options router-id 172.16.255.7
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

Configuring Router PE1

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure Router PE1:

NOTE: Repeat this procedure for Router PE2 after modifying the appropriate interface names, addresses, and other parameters.

1. Configure the physical interfaces. Ensure to configure `family mpls` on the core facing interface.

```
[edit]
user@PE1# set interfaces ge-0/0/0 unit 0 family inet address 172.16.12.2/30
user@PE1# set interfaces ge-0/0/2 unit 0 family inet address 10.1.23.1/30
user@PE1# set interfaces ge-0/0/2 unit 0 family mpls
```

2. Configure the loopback interfaces. The secondary loopback is optional and is applied under the routing instance in a later step.

```
[edit]
user@PE1# set interfaces lo0 unit 0 family inet address 10.1.255.2/32 primary
user@PE1# set interfaces lo0 unit 1 family inet address 10.1.255.22/32
```

3. Configure the router ID and the autonomous system number.

```
[edit]
user@PE1# set routing-options router-id 10.1.255.2
user@PE1# set routing-options autonomous-system 65000
```

4. Configure the OSPF protocol.

```
[edit]
user@PE1# set protocols ospf traffic-engineering
user@PE1# set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
user@PE1# set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

5. Configure the RSVP protocol.

```
[edit]
user@PE1# set protocols rsvp interface ge-0/0/2.0
user@PE1# set protocols rsvp interface lo0.0
```

6. Configure the MPLS protocol and an LSP towards the ABR. Include the `entropy-label` option to add the entropy label to the MPLS label stack.

```
[edit protocols]
user@PE1# set protocols mpls icmp-tunneling
user@PE1# set protocols mpls label-switched-path pe1-abr to 10.1.255.4
user@PE1# set protocols mpls label-switched-path pe1-abr entropy-label
user@PE1# set protocols mpls interface ge-0/0/2.0
user@PE1# set protocols mpls interface lo0.0
```

7. Configure IBGP using `family inet labeled-unicast` for the ABR peering and `family inet-vpn` for the PE2 peering. Enable entropy label capability for BGP labeled unicast.

```
[edit]
user@PE1# set protocols bgp group ibgp type internal
user@PE1# set protocols bgp group ibgp local-address 10.1.255.2
user@PE1# set protocols bgp group ibgp family inet labeled-unicast entropy-label
user@PE1# set protocols bgp group ibgp neighbor 10.1.255.4 family inet labeled-unicast rib
```

```
inet.3
user@PE1# set protocols bgp group ibgp neighbor 10.1.255.6 family inet-vpn unicast
```

8. Define a policy to export BGP VPN routes into OSPF. The policy is applied under OSPF in the routing instance.

```
[edit]
user@PE1# set policy-options policy-statement bgp-to-ospf from protocol bgp
user@PE1# set policy-options policy-statement bgp-to-ospf then accept
```

9. Define a load balancing policy and apply it under the `routing-options forwarding-table`. PE1 only has one path in the example therefore this step is not needed, but for this example we are applying the same load balancing policy on all devices.

```
[edit]
user@PE1# set policy-options policy-statement pplb then load-balance per-packet
user@PE1# set routing-options forwarding-table export pplb
```

10. Configure the Layer 3 VPN routing instance.

```
[edit]
user@PE1# set routing-instances VPN-l3vpn instance-type vrf
```

11. Assign the interfaces to the routing instance.

```
[edit]
user@PE1# set routing-instances VPN-l3vpn interface ge-0/0/0.0
user@PE1# set routing-instances VPN-l3vpn interface lo0.1
```

12. Configure the route distinguisher for the routing instance.

```
[edit]
user@PE1# set routing-instances VPN-l3vpn route-distinguisher 10.1.255.2:1
```

13. Configure a VPN routing and forwarding (VRF) target for the routing instance.

```
[edit]
user@PE1# set routing-instances VPN-l3vpn vrf-target target:65000:1
```

14. Configure the protocol OSPF under the routing instance and apply the previously configured `bgp-to-ospf` policy.

```
[edit]
user@PE1# set routing-instances VPN-l3vpn protocols ospf area 0.0.0.0 interface ge-0/0/0.0
user@PE1# set routing-instances VPN-l3vpn protocols ospf area 0.0.0.0 interface lo0.1
passive
user@PE1# set routing-instances VPN-l3vpn protocols ospf export bgp-to-ospf
```

Configuring Router P1

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure Router P1:

NOTE: Repeat this procedure for Router P2 after modifying the appropriate interface names, addresses, and other parameters.

1. Configure the physical interfaces.

```
[edit]
user@P1# set interfaces ge-0/0/0 unit 0 family inet address 10.1.23.2/30
user@P1# set interfaces ge-0/0/0 unit 0 family mpls
user@P1# set interfaces ge-0/0/2 unit 0 family inet address 10.1.34.1/30
user@P1# set interfaces ge-0/0/2 unit 0 family mpls
```

2. Configure the loopback interface.

```
[edit]
user@P1# set interfaces lo0 unit 0 family inet address 10.1.255.3/32 primary
```

3. Configure the router ID.

```
[edit]
user@P1# set routing-options router-id 10.1.255.3
```

4. Configure the OSPF protocol.

```
[edit]
user@P1# set protocols ospf traffic-engineering
user@P1# set protocols ospf area 0.0.0.0 interface lo0.0 passive
user@P1# set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
user@P1# set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
```

5. Configure the RSVP protocol.

```
[edit]
user@P1# set protocols rsvp interface ge-0/0/0.0
user@P1# set protocols rsvp interface lo0.0
user@P1# set protocols rsvp interface ge-0/0/2.0
```

6. Configure the MPLS protocol.

```
[edit]
user@P1# set protocols mpls icmp-tunneling
user@P1# set protocols mpls interface ge-0/0/0.0
user@P1# set protocols mpls interface lo0.0
user@P1# set protocols mpls interface ge-0/0/2.0
```

Configuring Router ABR

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure Router ABR:

1. Configure the physical interfaces.

```
[edit]
user@ABR# set interfaces ge-0/0/0 unit 0 family inet address 10.1.34.2/30
user@ABR# set interfaces ge-0/0/0 unit 0 family mpls
user@ABR# set interfaces ge-0/0/2 unit 0 family inet address 10.1.45.1/30
user@ABR# set interfaces ge-0/0/2 unit 0 family mpls
user@ABR# set interfaces ge-0/0/3 unit 0 family inet address 10.1.45.5/30
user@ABR# set interfaces ge-0/0/3 unit 0 family mpls
```

2. Configure the loopback interface.

```
[edit]
user@ABR# set interfaces lo0 unit 0 family inet address 10.1.255.4/32 primary
```

3. Configure MPLS labels that the router uses for hashing the packets to its destination for load balancing.

```
[edit]
user@ABR# set forwarding-options hash-key family mpls label-1
user@ABR# set forwarding-options hash-key family mpls label-2
user@ABR# set forwarding-options hash-key family mpls label-3
user@ABR# set forwarding-options enhanced-hash-key family mpls no-payload
```

4. Configure the router ID and the autonomous system number.

```
[edit]
user@ABR# set routing-options router-id 10.1.255.4
user@ABR# set routing-options autonomous-system 65000
```

5. Configure the OSPF protocol.

```
[edit]
user@ABR# set protocols ospf traffic-engineering
user@ABR# set protocols ospf area 0.0.0.0 interface lo0.0 passive
user@ABR# set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
user@ABR# set protocols ospf area 0.0.0.1 interface ge-0/0/2.0
user@ABR# set protocols ospf area 0.0.0.1 interface ge-0/0/3.0
```

6. Configure the RSVP protocol.

```
[edit]
user@ABR# set protocols rsvp interface lo0.0
user@ABR# set protocols rsvp interface ge-0/0/0.0
user@ABR# set protocols rsvp interface ge-0/0/2.0
user@ABR# set protocols rsvp interface ge-0/0/3.0
```

7. Configure the MPLS protocol and specify the LSPs towards PE1 and PE2. Two LSPs are created towards PE2 for the purpose of load balancing traffic to show different LSPs and interfaces are used.

```
[edit]
user@ABR# set protocols mpls icmp-tunneling
user@ABR# set protocols mpls label-switched-path abr-pe1 to 10.1.255.2
user@ABR# set protocols mpls label-switched-path abr-pe1 entropy-label
user@ABR# set protocols mpls label-switched-path abr-pe2 to 10.1.255.6
user@ABR# set protocols mpls label-switched-path abr-pe2 entropy-label
user@ABR# set protocols mpls label-switched-path abr-pe2 primary to-r6-1
user@ABR# set protocols mpls label-switched-path abr-pe2-2 to 10.1.255.6
user@ABR# set protocols mpls label-switched-path abr-pe2-2 entropy-label
user@ABR# set protocols mpls label-switched-path abr-pe2-2 primary to-r6-2
user@ABR# set protocols mpls path to-r6-1 10.1.45.2 strict
user@ABR# set protocols mpls path to-r6-1 10.1.56.2 strict
user@ABR# set protocols mpls path to-r6-2 10.1.45.6 strict
user@ABR# set protocols mpls path to-r6-2 10.1.56.6 strict
user@ABR# set protocols mpls interface lo0.0
user@ABR# set protocols mpls interface ge-0/0/0.0
user@ABR# set protocols mpls interface ge-0/0/2.0
user@ABR# set protocols mpls interface ge-0/0/3.0
```

8. Configure IBGP to both PE1 and PE2 using `family inet labeled-unicast`. Apply the policy to advertise the inet.3 loopback route from both PE1 and PE2. We show the policy in the next step.

```
[edit]
user@ABR# set protocols bgp group ibgp type internal
user@ABR# set protocols bgp group ibgp local-address 10.1.255.4
user@ABR# set protocols bgp group ibgp family inet labeled-unicast rib inet.3
user@ABR# set protocols bgp group ibgp neighbor 10.1.255.2 export send-inet3-pe2
user@ABR# set protocols bgp group ibgp neighbor 10.1.255.6 export send-inet3-pe1
```

9. Define a policy to match on the loopback addresses for PE1 and PE2.

```
[edit]
user@ABR# set policy-options policy-statement send-inet3-pe1 from route-filter
10.1.255.2/32 exact
user@ABR# set policy-options policy-statement send-inet3-pe1 then accept
user@ABR# set policy-options policy-statement send-inet3-pe2 from route-filter
10.1.255.6/32 exact
user@ABR# set policy-options policy-statement send-inet3-pe2 then accept
```

10. Define a policy for load balancing and apply it under the `routing-options forwarding-table`.

```
[edit]
user@ABR# set policy-options policy-statement pplb then load-balance per-packet
user@ABR# set routing-options forwarding-table export pplb
```

(Optional) Port-Mirroring Configuration

To see the entropy label that is applied you can capture the traffic. In this example a filter is applied on the PE1 facing interface on P1 to capture the CE1 to CE2 traffic. The traffic is sent to Host 1 for viewing. There are different ways to capture traffic than what we use in this example. For more information see Port Mirroring and Analyzers.

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure Router P1:

1. Configure the interfaces. In this example we are putting the interface connected to Host1 in a bridge domain and creating an IRB interface for verifying connectivity to Host1.

```
[edit]
user@P1# set interfaces ge-0/0/4 unit 0 family bridge interface-mode access
user@P1# set interfaces ge-0/0/4 unit 0 family bridge vlan-id 100
user@P1# set interfaces irb unit 0 family inet address 10.1.31.1/30
```

2. Configure the bridge domain.

```
[edit]
user@P1# set bridge-domains v100 vlan-id 100
user@P1# set bridge-domains v100 routing-interface irb.0
```

3. Configure a filter to capture the traffic. For this example we are capturing all traffic.

```
[edit]
user@P1# set firewall family any filter test term 1 then count test
user@P1# set firewall family any filter test term 1 then port-mirror
user@P1# set firewall family any filter test term 1 then accept
```

4. Apply the filter to the PE1 facing interface.

```
[edit]
user@P1# set interfaces ge-0/0/0 unit 0 filter input test
```

5. Configure the port mirroring options. For this example we are mirroring all traffic and sending it to Host1 connected to interface ge-0/0/4.

```
[edit]
user@P1# set forwarding-options port-mirroring input rate 1
user@P1# set forwarding-options port-mirroring family any output interface ge-0/0/4.0
```

Verification

IN THIS SECTION

- Verifying That the Entropy Label Capability Is Being Advertised | **663**
- Verifying That Router PE1 Receives the Entropy Label Advertisement | **664**
- Verifying ECMP at the ABR to PE2 | **666**
- Show Routes to CE2 on PE1 | **667**
- Ping CE2 from CE1 | **669**
- Verify Load Balancing | **669**
- Verify the Entropy Label | **670**

Confirm that the configuration is working properly.

Verifying That the Entropy Label Capability Is Being Advertised

Purpose

Verify that the entropy label capability path attribute is being advertised from the ABR to PE1 for the route to PE2.

Action

From operational mode, run the **show route advertising-protocol bgp 10.1.255.2 detail** command on Router ABR.

```
user@ABR> show route advertising-protocol bgp 10.1.255.2 detail

inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
* 10.1.255.6/32 (1 entry, 1 announced)
 BGP group ibgp type Internal
     Route Label: 299952
     Nexthop: Self
     Flags: Nexthop Change
     MED: 2
     Localpref: 4294967294
     AS path: [65000] I
```

```
      Entropy label capable
```

Meaning

The output shows that the host PE2 with the IP address of 10.1.255.6 has the entropy label capability and the route label that is used. The host is advertising the entropy label capability to its BGP neighbors.

Verifying That Router PE1 Receives the Entropy Label Advertisement

Purpose

Verify that Router PE1 receives the entropy label advertisement for Router PE2.

Action

From operational mode, run the **show route protocol bgp 10.1.255.6 extensive** command on Router PE1.

```
user@PE1> show route protocol bgp 10.1.255.6 extensive

inet.0: 19 destinations, 19 routes (19 active, 0 holddown, 0 hidden)

inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
10.1.255.6/32 (1 entry, 1 announced)
        *BGP    Preference: 170/1
                Next hop type: Indirect, Next hop index: 0
                Address: 0x7b3ffd4
                Next-hop reference count: 2, key opaque handle: 0x0, non-key opaque handle: 0x0
                Source: 10.1.255.4
                Next hop type: Router, Next hop index: 0
                Next hop: 10.1.23.2 via ge-0/0/2.0, selected
                Label-switched-path pe1-abr
                Label operation: Push 299952, Push 299808(top)
                Label TTL action: prop-ttl, prop-ttl(top)
                Load balance label: Label 299952: Entropy label; Label 299808: None;
                Label element ptr: 0x93d6bf8
                Label parent element ptr: 0x93d6c20
                Label element references: 3
                Label element child references: 2
                Label element lsp id: 0
```

```
                Session Id: 0
                Protocol next hop: 10.1.255.4
                Label operation: Push 299952
                Label TTL action: prop-ttl
                Load balance label: Label 299952: Entropy label;
                Indirect next hop: 0x758c05c - INH Session ID: 0
                State: <Active Int Ext>
                Local AS: 65000 Peer AS: 65000
                Age: 1:33:11    Metric: 2       Metric2: 2
                Validation State: unverified
                Task: BGP_65000.10.1.255.4
                Announcement bits (2): 3-Resolve tree 1 4-Resolve_IGP_FRR task
                AS path: I
                Accepted
                Route Label: 299952
                Localpref: 4294967294
                Router ID: 10.1.255.4
                Session-IDs associated:
                Session-id: 324 Version: 3
                Thread: junos-main
                Indirect next hops: 1
                        Protocol next hop: 10.1.255.4 Metric: 2 ResolvState: Resolved
                        Label operation: Push 299952
                        Label TTL action: prop-ttl
                        Load balance label: Label 299952: Entropy label;
                        Indirect next hop: 0x758c05c - INH Session ID: 0
                        Indirect path forwarding next hops: 1
                                Next hop type: Router
                                Next hop: 10.1.23.2 via ge-0/0/2.0
                                Session Id: 0
                                10.1.255.4/32 Originating RIB: inet.3
                                  Metric: 2 Node path count: 1
                                  Forwarding nexthops: 1
                                        Next hop type: Router
                                        Next hop: 10.1.23.2 via ge-0/0/2.0
                                        Session Id: 0
```

Meaning

Router PE1 receives the entropy label capability advertisement from its BGP neighbor.

Verifying ECMP at the ABR to PE2

Purpose

Verify equal-cost multipath (ECMP) to PE2.

Action

From operational mode, run the **show route table mpls.0** and **show route forwarding-table label <label>**commands on Router ABR.

```
user@ABR> show route table mpls.0

mpls.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 2w1d 23:02:11, metric 1
                       Receive
1                  *[MPLS/0] 2w1d 23:02:11, metric 1
                       Receive
2                  *[MPLS/0] 2w1d 23:02:11, metric 1
                       Receive
13                 *[MPLS/0] 2w1d 23:02:11, metric 1
                       Receive
299936             *[VPN/170] 2d 21:47:02
                    >  to 10.1.34.1 via ge-0/0/0.0, label-switched-path abr-pe1
299952             *[VPN/170] 2d 21:47:02
                    >  to 10.1.45.2 via ge-0/0/2.0, label-switched-path abr-pe2
                       to 10.1.45.6 via ge-0/0/3.0, label-switched-path abr-pe2-2

ruser@ABR> show route forwarding-table label 299952
Routing table: default.mpls
MPLS:
Destination        Type RtRef Next hop           Type Index    NhRef Netif
299952             user     0                    ulst  1048575     2
                              10.1.45.2         Swap 299824      516     2 ge-0/0/2.0
                              10.1.45.6         Swap 299840      572     2 ge-0/0/3.0
...
```

Meaning

The output shows an ECMP for the label used for the BGP labeled unicast route.

Show Routes to CE2 on PE1

Purpose

Verify the routes to CE2.

Action

From operational mode, run the **show route table VPN-l3vpn.inet.0 172.16.255.7 extensive** and **show route table VPN-l3vpn.inet.0 192.168.255.7 extensive**commands on Router PE1.

```
user@PE1> show route table VPN-l3vpn.inet.0 172.16.255.7 extensive

VPN-l3vpn.inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
172.16.255.7/32 (1 entry, 1 announced)
TSI:
OSPF area : 0.0.0.0, LSA ID : 172.16.255.7, LSA type : Summary
KRT in-kernel 172.16.255.7/32 -> {indirect(1048574)}
        *BGP    Preference: 170/-101
                Route Distinguisher: 10.1.255.6:1
                Next hop type: Indirect, Next hop index: 0
                Address: 0x7b40434
                Next-hop reference count: 9, key opaque handle: 0x0, non-key opaque handle: 0x0
                Source: 10.1.255.6
                Next hop type: Router, Next hop index: 515
                Next hop: 10.1.23.2 via ge-0/0/2.0, selected
                Label-switched-path pe1-abr
                Label operation: Push 299824, Push 299952, Push 299808(top)
                Label TTL action: prop-ttl, prop-ttl, prop-ttl(top)
                Load balance label: Label 299824: None; Label 299952: Entropy label; Label
299808: None;
                Label element ptr: 0x93d6c98
                Label parent element ptr: 0x93d6bf8
                Label element references: 1
                Label element child references: 0
                Label element lsp id: 0
                Session Id: 140
                Protocol next hop: 10.1.255.6
```

```
                Label operation: Push 299824
                Label TTL action: prop-ttl
                Load balance label: Label 299824: None;
...

user@PE1> show route table VPN-l3vpn.inet.0 192.168.255.7 extensive
VPN-l3vpn.inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
192.168.255.7/32 (1 entry, 1 announced)
TSI:
OSPF area : 0.0.0.0, LSA ID : 192.168.255.7, LSA type : Summary
KRT in-kernel 192.168.255.7/32 -> {indirect(1048574)}
        *BGP    Preference: 170/-101
                Route Distinguisher: 10.1.255.6:1
                Next hop type: Indirect, Next hop index: 0
                Address: 0x7b40434
                Next-hop reference count: 9, key opaque handle: 0x0, non-key opaque handle: 0x0
                Source: 10.1.255.6
                Next hop type: Router, Next hop index: 515
                Next hop: 10.1.23.2 via ge-0/0/2.0, selected
                Label-switched-path pe1-abr
                Label operation: Push 299824, Push 299952, Push 299808(top)
                Label TTL action: prop-ttl, prop-ttl, prop-ttl(top)
                Load balance label: Label 299824: None; Label 299952: Entropy label; Label
299808: None;
                Label element ptr: 0x93d6c98
                Label parent element ptr: 0x93d6bf8
                Label element references: 1
                Label element child references: 0
                Label element lsp id: 0
                Session Id: 140
                Protocol next hop: 10.1.255.6
                Label operation: Push 299824
                Label TTL action: prop-ttl
                Load balance label: Label 299824: None;
...
```

Meaning

The output shows the same labels are used for both routes.

Ping CE2 from CE1

Purpose

Verify connectivity and to use for verifying load balancing.

Action

From operational mode, run the **ping 172.16.255.7 source 172.16.12.1 rapid count 100** and **ping 192.168.255.7 source 192.168.255.1 rapid count 200**commands on Router PE1.

```
user@CE1> ping 172.16.255.7 source 172.16.12.1 rapid count 100
PING 172.16.255.7 (172.16.255.7): 56 data bytes
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!
--- 172.16.255.7 ping statistics ---
100 packets transmitted, 100 packets received, 0% packet loss
round-trip min/avg/max/stddev = 5.369/6.070/8.828/0.612 ms

user@CE1> ping 192.168.255.7 source 192.168.255.1 rapid count 200
PING 192.168.255.7 (192.168.255.7): 56 data bytes
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!
--- 192.168.255.7 ping statistics ---
200 packets transmitted, 200 packets received, 0% packet loss
round-trip min/avg/max/stddev = 5.086/5.994/10.665/0.649 ms
```

Meaning

The output shows pings are successful.

Verify Load Balancing

Purpose

Verify load balancing.

Action

From operational mode, run the **show mpls lsp ingress statistics** command on the ABR.

```
user@ABR> show mpls lsp ingress statistics
Ingress LSP: 3 sessions
To              From            State     Packets            Bytes LSPname
10.1.255.2      10.1.255.4      Up            300            30000 abr-pe1
10.1.255.6      10.1.255.4      Up            200            20000 abr-pe2
10.1.255.6      10.1.255.4      Up            100            10000 abr-pe2-2
Total 3 displayed, Up 3, Down 0
```

Meaning

The output shows the first ping from the previous command used LSP **abr-pe2-2** and the second ping used LSP **abr-pe2**.

Verify the Entropy Label

Purpose

Verify the entropy label is different between the pings that were used.

Action

On Host 1, run the **tcpdump -i eth1 -n**.

```
user@Host1# tcpdump -i eth1 -n
...
13:42:31.993274 MPLS (label 299808, exp 0, ttl 63) (label 299952, exp 0, ttl 63) (label 7, exp
0, ttl 63) (label 1012776, exp 0, ttl 0)
(label 299824, exp 0, [S], ttl 63) IP 172.16.12.1 > 172.16.255.7: ICMP echo request, id 32813,
seq 9, length 64
...
13:43:19.570260 MPLS (label 299808, exp 0, ttl 63) (label 299952, exp 0, ttl 63) (label 7, exp
0, ttl 63) (label 691092, exp 0, ttl 0)
(label 299824, exp 0, [S], ttl 63) IP 192.168.255.1 > 192.168.255.7: ICMP echo request, id
46381, seq 9, length 64
```

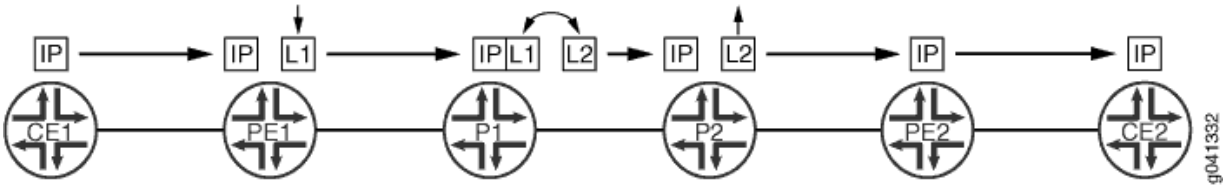
Meaning

The output shows the different value for the entropy label for the two different ping commands.

Configuring Ultimate-Hop Popping for RSVP-signaled MPLS LSPs

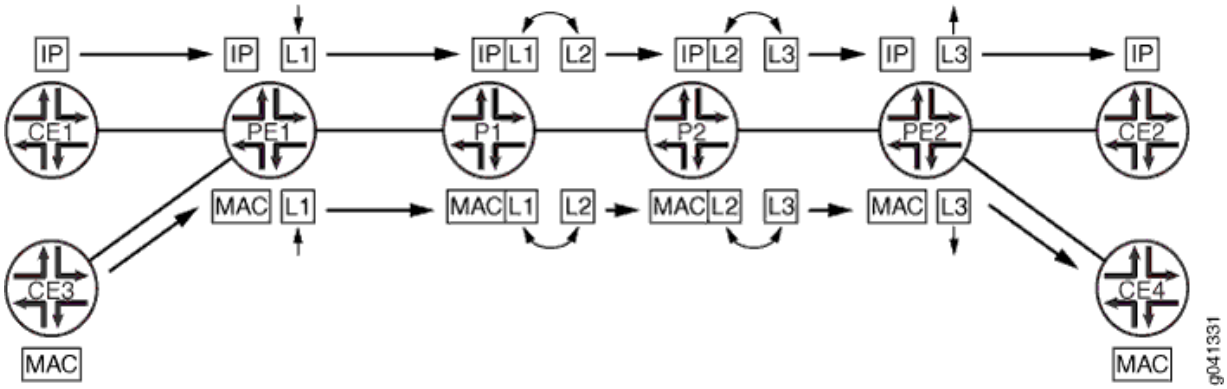
By default, RSVP-signaled LSPs use penultimate-hop popping (*PHP*). Figure 44 on page 671 illustrates a penultimate-hop popping LSP between Router PE1 and Router PE2. Router CE1 forwards a packet to its next hop (Router PE1), which is also the LSP ingress. Router PE1 pushes label 1 on the packet and forwards the labeled packet to Router P1. Router P1 completes the standard MPLS label swapping operation, swapping label 1 for label 2, and forwards the packet to Router P2. Since Router P2 is the penultimate-hop router for the LSP to Router PE2, it first pops the label and then forwards the packet to Router PE2. When Router PE2 receives it, the packet can have a service label, an explicit-null label, or just be a plain IP or VPLS packet. Router PE2 forwards the unlabeled packet to Router CE2.

Figure 44: Penultimate-Hop Popping for an LSP



You can also configure ultimate-hop popping (*UHP*) (as shown in Figure 45 on page 672) for RSVP-signaled LSPs. Some network applications can require that packets arrive at the egress router (Router PE2) with a non-null outer label. For an ultimate- hop popping LSP, the penultimate router (Router P2 in Figure 45 on page 672) performs the standard MPLS label swapping operation (in this example, label 2 for label 3) before forwarding the packet to egress Router PE2. Router PE2 pops the outer label and performs a second lookup of the packet address to determine the end destination. It then forwards the packet to the appropriate destination (either Router CE2 or Router CE4).

Figure 45: Ultimate-Hop Popping for an LSP



The following network applications require that you configure UHP LSPs:

- MPLS-TP for performance monitoring and in-band *OAM*
- Edge protection virtual circuits

The following features do not support the UHP behavior:

- LDP-signaled LSPs
- Static LSPs
- Point-to-multipoint LSPs
- CCC
- `traceroute` command

For more information about UHP behavior, see Internet draft draft-ietf-mpls-rsvp-te-no-php-oob-mapping-01.txt, *Non PHP behavior and Out-of-Band Mapping for RSVP-TE LSPs*.

For point-to-point RSVP-signaled LSPs, UHP behavior is signaled from the LSP ingress. Based on the ingress router configuration, RSVP can signal the UHP LSP with the non-PHP flag set. RSVP PATH messages carry the two flags in the LSP-ATTRIBUTES object. When the egress router receives the PATH message, it assigns a non-null label to the LSP. RSVP also creates and installs two routes in the mpls.0 routing table. S refers to the S bit of the MPLS label, which indicates whether or not the bottom of the label stack has been reached.

- Route S=0—Indicates that there are more labels in the stack. The next hop for this route points to the mpls.0 routing table, triggering a chained MPLS label lookup to discover the remaining MPLS labels in the stack.

- Route S=1—Indicates that there are no more labels. The next hop points to the inet.0 routing table if the platform supports chained and multi-family lookup. Alternatively, the label route can point to a *VT* interface to initiate IP forwarding.

If you enable UHP LSPs, MPLS applications such as Layer 3 VPNs, VPLS, Layer 2 VPNs, and Layer 2 circuits can use the UHP LSPs. The following explains how UHP LSPs affect the different types of MPLS applications:

- Layer 2 VPNs and Layer 2 circuits—A packet arrives at the PE router (egress of the UHP LSP) with two labels. The outer label (S=0) is the UHP label, and the inner label (S=1) is the *VC* label. A lookup based on the transport label results in a table handle for the mpls.0 routing table. There is an additional route in the mpls.0 routing table corresponding to the inner label. A lookup based on the inner label results in the CE router next hop.

- Layer 3 VPN—A packet arrives at the PE router (egress of the UHP LSP) with two labels. The outer label (S=0) is the UHP label, and the inner label is the VPN label (S=1). A lookup based on the transport label results in the table handle for the mpls.0 routing table. There are two cases in this scenario. By default, Layer 3 VPNs advertise the per-next hop label. A lookup based on the inner label results in the next hop toward the CE router. However, if you have configured the `vrf-table-label` statement for the Layer 3 VPN routing instance, the inner *LSI* label points to the VRF routing table. An IP lookup is also completed for the VRF routing table.

  NOTE: UHP for Layer 3 VPNs configured with the `vrf-table-label` statement is supported on MX Series 5G Universal Routing Platforms only.

- VPLS—A packet arrives at the PE router (egress of the UHP LSP) with two labels. The outer label is the transport label (S=0) and the inner label is the VPLS label (S=1). A lookup based on the transport label results in the table handle for the mpls.0 routing table. A lookup based on the inner label in mpls.0 routing table results in the LSI tunnel interface of the VPLS routing instance if tunnel-services is not configured (or a VT interface not available). MX 3D Series routers support chained lookup and multi-family lookup.

  NOTE: UHP for VPLS configured with the `no-tunnel-service` statement is supported on MX 3D Series routers only.

- IPv4 over MPLS—A packet arrives at the PE router (egress of the UHP LSP) with one label (S=1). A lookup based on this label returns a VT tunnel interface. Another IP lookup is completed on the VT interface to determine where to forward the packet. If the routing platform supports multi-family and chained lookups (for example, MX 3D routers and PTX Series Packet Transport Routers), lookup based on label route (S=1) points to the inet.0 routing table.

- IPv6 over MPLS—For IPv6 tunneling over MPLS, PE routers advertise IPv6 routes to each other with a label value of 2. This is the explicit null label for IPv6. As a result, the forwarding next hops for IPv6 routes that are learned from remote PE routers normally push two labels. The inner label is 2 (it could be different if the advertising PE router is from another vendor), and the router label is the LSP label. Packets arrive at the PE router (egress of the UHP LSP) with two labels. The outer label is the transport label (S=0), and the inner label is the IPv6 explicit-null label (label 2). Lookup based on the inner label in the mpls.0 routing table redirects back to the mpls.0 routing table. On MX 3D Series routers, the inner label (label 2) is stripped off and an IPv6 lookup is done using the inet6.0 routing table.

- Enabling both PHP and UHP LSPs—You can configure both PHP and UHP LSPs over the same network paths. You can separate PHP and UHP traffic by selecting forwarding LSP next hops using a regular expression with the `install-nexthop` statement. You can also separate traffic by simply naming the LSPs appropriately.

The following statements enable ultimate-hop popping for an LSP. You can enable this feature on a specific LSP or for all of the ingress LSPs configured on the router. Configure these statements on the router at the LSP ingress.

1. To enable ultimate-hop popping, include the `ultimate-hop-popping` statement:

```
ultimate-hop-popping;
```

   Include this statement at the `[edit protocols mpls label-switched-path` *`label-switched-path-name`*`]` hierarchy level to enable ultimate-hop popping on a specific LSP. Include this statement at the `[edit protocols mpls]` hierarchy level to enable ultimate-hop popping on all of the ingress LSPs configured on the router. You can also configure the `ultimate-hop-popping` statement under the equivalent `[edit logical-routers]` hierarchy levels.

   NOTE: When you enable ultimate-hop popping, RSVP attempts to resignal existing LSPs as ultimate-hop popping LSPs in a make-before-break fashion. If an egress router does not support ultimate-hop popping, the existing LSP is torn down (RSVP sends a *PathTear message* along an LSP's path, removing the path state and dependent reservation state and releasing the associated networking resources).

   If you disable ultimate-hop popping, RSVP resignals existing LSPs as penultimate-hop popping LSPs in a make-before-break fashion.

2. If you want to enable both ultimate-hop-popping and chained next hops on MX 3D Series routers only, you also need to configure the `enhanced-ip` option for the `network-services` statement:

```
network-services enhanced-ip;
```

You configure this statement at the [edit chassis] hierarchy level. Once you have configured the network-services statement, you need to reboot the router to enable UHP behavior.

Configuring Explicit-Path LSPs

If you disable constrained-path label-switched path (LSP) computation, as described in "Disabling Constrained-Path LSP Computation" on page 589, you can configure LSPs manually or allow the LSPs to follow the IGP path.

When explicit-path LSPs are configured, the LSP is established along the path you specified. If the path is topologically not feasible, either because the network is partitioned or insufficient resources are available along some parts of the path, the LSP will fail. No alternative paths can be used. If the setup succeeds, the LSP stays on the defined path indefinitely.

To configure an explicit-path LSP, follow these steps:

1. Configure the path information in a named path, as described in "Creating Named Paths" on page 594. To configure complete path information, specify every router hop between the ingress and egress routers, preferably using the strict attribute. To configure incomplete path information, specify only a subset of router hops, using the loose attribute in places where the path is incomplete.

   For incomplete paths, the MPLS routers complete the path by querying the local routing table. This query is done on a hop-by-hop basis, and each router can figure out only enough information to reach the next explicit hop. It might be necessary to traverse a number of routers to reach the next (loose) explicit hop.

   Configuring incomplete path information creates portions of the path that depend on the current routing table, and this portion of the path can reroute itself as the topology changes. Therefore, an explicit-path LSP that contains incomplete path information is not completely fixed. These types of LSPs have only a limited ability to repair themselves, and they tend to create loops or flaps depending on the contents of the local routing table.

2. To configure the LSP and point it to the named path, use either the primary or secondary statement, as described in "Configuring Primary and Secondary LSPs" on page 683.

3. Disable constrained-path LSP computation by including the no-cspf statement either as part of the LSP or as part of a primary or secondary statement. For more information, see "Disabling Constrained-Path LSP Computation" on page 589.

4. Configure any other LSP properties.

NOTE: When defining a constrained path LSP using more than one strict hop belonging to the egress node, the first strict hop must be set to match the IP address assigned to the egress node on the interface that receives the RSVP Path message. If the incoming

RSVP Path message arrives on an interface with a different IP address the LSP is rejected.

Prior to Junos OS 20.3X75-D20 or 22.2R1, any additional strict hop after the strict hop matching the IP address of the interface that receives the RSVP Path message *must* be set to match a loopback address assigned to the egress node. In later Junos releases this behavior is changed to permit an additional strict hop that matches an IP address assigned to *any* interface on the egress node

Using explicit-path LSPs has the following drawbacks:

- More configuration effort is required.
- Configured path information cannot take into account dynamic network bandwidth reservation, so the LSPs tend to fail when resources become depleted.
- When an explicit-path LSP fails, you might need to manually repair it.

Because of these limitations, we recommend that you use explicit-path LSPs only in controlled situations, such as to enforce an optimized LSP placement strategy resulting from computations with an offline simulation software package.

Example: Configuring an Explicit-Path LSP

On the ingress router, create an explicit-path LSP, and specify the transit routers between the ingress and egress routers. In this configuration, no constrained-path computation is performed. For the primary path, all intermediate hops are strictly specified so that its route cannot change. The secondary path must travel through router 14.1.1.1 first, then take whatever route is available to reach the destination. The remaining route taken by the secondary path is typically the shortest path computed by the IGP.

NOTE: When defining a constrained path LSP using more than one strict hop belonging to the egress node, the first strict hop must be set to match the IP address assigned to the egress node on the interface that receives the RSVP Path message. If the incoming RSVP Path message arrives on an interface with a different IP address the LSP is rejected.

Prior to Junos OS 20.3X75-D20 or 22.2R1, any additional strict hop after the strict hop matching the IP address of the interface that receives the RSVP Path message *must* be set to match a loopback address assigned to the egress node. In later Junos releases this behavior is changed to permit an additional strict hop that matches an IP address assigned to *any* interface on the egress node

```
[edit]
interfaces {
```

```
    so-0/0/0 {
        unit 0 {
            family mpls;
        }
    }
}
protocols {
    rsvp {
        interface so-0/0/0;
    }
    mpls {
        path to-hastings {
            14.1.1.1 strict;
            13.1.1.1 strict;
            12.1.1.1 strict;
            11.1.1.1 strict;
        }
        path alt-hastings {
            14.1.1.1 strict;
            11.1.1.1 loose; # Any IGP route is acceptable
        }
        label-switched-path hastings {
            to 11.1.1.1;
            hop-limit 32;
            bandwidth 10m; # Reserve 10 Mbps
            no-cspf;  # do not perform constrained-path computation
            primary to-hastings;
            secondary alt-hastings;
        }
        interface so-0/0/0;
    }
}
```

LSP Bandwidth Oversubscription Overview

LSPs are established with bandwidth reservations configured for the maximum amount of traffic you expect to traverse the LSP. Not all LSPs carry the maximum amount of traffic over their links at all times. For example, even if the bandwidth for link A has been completely reserved, actual bandwidth might still be available but not currently in use. This excess bandwidth can be used by allowing other LSPs to also use link A, oversubscribing the link. You can oversubscribe the bandwidth configured for individual class types or specify a single value for all of the class types using an interface.

You can use oversubscription to take advantage of the statistical nature of traffic patterns and to permit higher utilization of links.

The following examples describe how you might use bandwidth oversubscription and undersubscription:

- Use oversubscription on class types where peak periods of traffic do not coincide in time.
- Use oversubscription of class types carrying best-effort traffic. You take the risk of temporarily delaying or dropping traffic in exchange for making better utilization of network resources.
- Give different degrees of oversubscription or undersubscription of traffic for the different class types. For instance, you configure the subscription for classes of traffic as follows:
  - Best effort—`ct0 1000`
  - Voice—`ct3 1`

When you undersubscribe a class type for a multiclass LSP, the total demand of all RSVP sessions is always less than the actual capacity of the class type. You can use undersubscription to limit the utilization of a class type.

The bandwidth oversubscription calculation occurs on the local router only. Because no signaling or other interaction is required from other routers in the network, the feature can be enabled on individual routers without being enabled or available on other routers which might not support this feature. Neighboring routers do not need to know about the oversubscription calculation, they rely on the IGP.

The following sections describe the types of bandwidth oversubscription available in the Junos OS:

- "LSP Size Oversubscription" on page 678
- "LSP Link Size Oversubscription" on page 678
- "Class Type Oversubscription and Local Oversubscription Multipliers" on page 679

LSP Size Oversubscription

For LSP size oversubscription, you simply configure less bandwidth than the peak rate expected for the LSP. You also might need to adjust the configuration for automatic policers. Automatic policers manage the traffic assigned to an LSP, ensuring that it does not exceed the configured bandwidth values. LSP size oversubscription requires that the LSP can exceed its configured bandwidth allocation.

Policing is still possible. However, the policer must be manually configured to account for the maximum bandwidth planned for the LSP, rather than for the configured value.

LSP Link Size Oversubscription

You can increase the maximum reservable bandwidth on the link and use the inflated values for bandwidth accounting. Use the `subscription` statement to oversubscribe the link. The configured value is

applied to all class type bandwidth allocations on the link. For more information about link size oversubscription, see "Configuring the Bandwidth Subscription Percentage for LSPs" on page 679.

Class Type Oversubscription and Local Oversubscription Multipliers

Local oversubscription multipliers (LOMs) allow different oversubscription values for different class types. LOMs are useful for networks where the oversubscription ratio needs to be configured differently on different links and where oversubscription values are required for different classes. You might use this feature to oversubscribe class types handling best-effort traffic, but use no oversubscription for class types handling voice traffic. An LOM is calculated locally on the router. No information related to an LOM is signaled to other routers in the network.

An LOM is configurable on each link and for each class type. The per-class type LOM allows you to increase or decrease the oversubscription ratio. The per-class-type LOM is factored into all local bandwidth accounting for admission control and IGP advertisement of unreserved bandwidths.

The LOM calculation is tied to the bandwidth model (MAM, extended MAM, and Russian dolls) used, because the effect of oversubscription across class types must be accounted for accurately.

NOTE: All LOM calculations are performed by the Junos OS and require no user intervention.

The formulas related to the oversubscription of class types are described in the following sections:

- Class Type Bandwidth and the LOM
- LOM Calculation for the MAM and Extended MAM Bandwidth Models
- LOM Calculation for the Russian Dolls Bandwidth Model
- Example: LOM Calculation

Configuring the Bandwidth Subscription Percentage for LSPs

IN THIS SECTION

- Constraints on Configuring Bandwidth Subscription | 680

By default, RSVP allows all of a class type's bandwidth (100 percent) to be used for RSVP reservations. When you oversubscribe a class type for a multiclass LSP, the aggregate demand of all RSVP sessions is allowed to exceed the actual capacity of the class type.

If you want to oversubscribe or undersubscribe all of the class types on an interface using the same percentage bandwidth, configure the percentage using the `subscription` statement:

```
subscription percentage;
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section.

To undersubscribe or oversubscribe the bandwidth for each class type, configure a percentage for each class type (`ct0`, `ct1`, `ct2`, and `ct3`) option for the `subscription` statement. When you oversubscribe a class type, an LOM is applied to calculate the actual bandwidth reserved. See "Class Type Oversubscription and Local Oversubscription Multipliers" on page 679 for more information.

```
subscription {
    ct0 percentage;
    ct1 percentage;
    ct2 percentage;
    ct3 percentage;
}
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section.

percentage is the percentage of class type bandwidth that RSVP allows to be used for reservations. It can be a value from 0 through 65,000 percent. If you specify a value greater than 100, you are oversubscribing the interface or class type.

The value you configure when you oversubscribe a class type is a percentage of the class type bandwidth that can actually be used. The default subscription value is 100 percent.

You can use the `subscription` statement to disable new RSVP sessions for one or more class types. If you configure a percentage of 0, no new sessions (including those with zero bandwidth requirements) are permitted for the class type.

Existing RSVP sessions are not affected by changing the subscription factor. To clear an existing session, issue the `clear rsvp session` command. For more information on the `clear rsvp session` command, see the CLI Explorer.

Constraints on Configuring Bandwidth Subscription

Be aware of the following issues when configuring bandwidth subscription:

- If you configure bandwidth constraints at the [edit class-of-service interface *interface-name*] hierarchy level, they override any bandwidth configuration you specify at the [edit protocols rsvp interface *interface-name* bandwidth] hierarchy level for Diffserv-TE. Also note that either of the CoS or RSVP bandwidth constraints can override the interface hardware bandwidth constraints.

- If you configure a bandwidth subscription value for a specific interface that differs from the value configured for all interfaces (by including different values for the subscription statement at the [edit protocols rsvp interface *interface-name*] and [edit protocols rsvp interface all] hierarchy levels), the interface-specific value is used for that interface.

- You can configure subscription for each class type only if you also configure a bandwidth model. If no bandwidth model is configured, the commit operation fails with the following error message:

```
user@host# commit check
[edit protocols rsvp interface all]
  'subscription'
RSVP: Must have a diffserv-te bandwidth model configured when configuring subscription per
traffic class.
error: configuration check-out failed
```

- You cannot include the subscription statement both in the configuration for a specific class type and the configuration for the entire interface. The commit operation fails with the following error message:

```
user@host# commit check
[edit protocols rsvp interface all]
  'subscription'
    RSVP: Cannot configure both link subscription and per traffic class subscription.
error: configuration check-out failed
```

Detecting MPLS MTU Exceed Errors

Junos supports ICMP error message generation towards source for error conditions like TTL expiry, unreachable destination, unreachable destination (DF), redirect, etc. for IPv4, IPv6, and MPLS packets.

Starting from Junos OS Release 23.4R1, Junos supports ICMP error message generation for MTU exceed errors in an MPLS environment.

If a MPLS labeled packet failure occurs at the egress interface of the core or transit nodes due to MTU exceed errors, an ICMP error message is generated towards the peer PE device terminating the LSP. The peer PE device decapsulates the MPLS header and routes the ICMP error message to the source device. The return path could be either pure IP path or a different LSP based on the state of the device's routing

table. The source or customer edge device receives the ICMP error message and adjusts the packet size to avoid MTU errors.

RFC3032 defines ICMP tunnel mechanism to handle ICMP error message generation for MPLS packets for TTL expiry and MTU exceeded exceptions.

The following are some of the benefits of ICMP error message generation for MTU exceed errors in an MPLS environment:

- Understand if the cause of failure was due to MTU exceed errors.
- Know about MTU exceeded failures on transit nodes and ingress nodes in an MPLS setup.
- Supports the use case where an application on your network communicates to the endpoint over a Layer 3 VPN (unicast) or static LSP network.

To enable ICMP MTU exceed error message generation, you need to configure ICMP tunneling by enabling the `icmp-tunnelling` statement at the [`edit protocol mpls`] hierarchy level on the core and transit devices.

NOTE: For ICMP MTU exceed error message generation to work, you need to setup route tables on the peer CE device to route packet back to source CE device, else the ICMP MTU exceed error packets will be dropped.

When you configure the `chained-composite-next-hop transit <>` statement at the [`edit routing-options forwarding-table`] hierarchy level and MPLS MTU exception on transit router, there is no guarantee for the ICMP error message generation to work.

When you configure the `chained-composite-next-hop transit <>` statement at the [`edit routing-options forwarding-table`] hierarchy level on the ingress router, and ingress and egress interfaces are on different FPCs/PFEs, with ingress FPC/PFE performing more than 1 MPLS label addition, then, the ICMP error generation for MPLS MTU exception on ingress router will not be accurate.

ICMP error message generation is not supported for:

- Layer 2 VPN and Layer 2 Circuit configurations.
- Multicast configurations with traffic carried over MPLS. MTU exceptions packets will be counted and dropped.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.1R9	Starting in Junos OS Release 14.1R9, 15.1R7, 16.1R5, 16.1X2, 16.2R3, and 17.2R2, all zero value bandwidth samples are considered as underflow samples, except for the zero value samples that arrive after an LSP comes up for the first time, and the zero value samples that arrive first after a Routing Engine switchover.

RELATED DOCUMENTATION

MPLS Overview | 2

Primary, Secondary, and Static LSP Configuration

IN THIS SECTION

- Configuring Primary and Secondary LSPs | 683
- Configuring Hot Standby of Secondary Paths for LSPs | 689
- Configuring Static LSPs | 690
- Configuring Static Label Switched Paths for MPLS (CLI Procedure) | 703
- Configuring Static Label Switched Paths for MPLS | 706

Configuring Primary and Secondary LSPs

IN THIS SECTION

- Configuring Primary and Secondary Paths for an LSP | 684
- Configuring the Revert Timer for LSPs | 685
- Specifying the Conditions for Path Selection | 686
- Configure a Primary Path | 687

By default, an LSP routes itself hop-by-hop toward the egress router. The LSP tends to follow the shortest path as dictated by the local routing table, usually taking the same path as destination-based, best-effort traffic. These paths are “soft” in nature because they automatically re-route themselves whenever a change occurs in a routing table or in the status of a node or link.

To configure the path so that it follows a particular route, create a named path using the `path` statement, as described in "Creating Named Paths" on page 594. Then apply the named path by including the `primary` or `secondary` statement. A named path can be referenced by any number of LSPs.

To configure primary and secondary paths for an LSP, complete the steps in the following sections:

Configuring Primary and Secondary Paths for an LSP

The `primary` statement creates the primary path, which is the LSP’s preferred path. The `secondary` statement creates an alternative path. If the primary path can no longer reach the egress router, the alternative path is used.

To configure primary and secondary paths, include the `primary` and `secondary` statements:

```
            primary

            path-name {
    ...
}
secondary

            path-name {
    ...
}
```

You can include these statements at the following hierarchy levels:

- `[edit protocols mpls label-switched-path lsp-name]`
- `[edit logical-systems logical-system-name protocols mpls label-switched-path lsp-name]`

When the software switches from the primary to a secondary path, it continuously attempts to revert to the primary path, switching back to it when it is again reachable but no sooner than the time specified in the `revert-timer` statement. (For more information, see "Configuring the Connection Between Ingress and Egress Routers" on page 599.)

You can configure zero or one primary path. If you do not configure a primary path, the first secondary path that is established is selected as the path.

You can configure zero or more secondary paths. All secondary paths are equal. The software does not attempt to switch among secondary paths. If the current secondary path is not available, the next one is tried in no particular order. To create a set of equal paths, specify secondary paths without specifying a primary path.

If you do not specify any named paths, or if the path that you specify is empty, the software makes all routing decisions necessary to reach the egress router.

Configuring the Revert Timer for LSPs

For LSPs configured with both primary and secondary paths, it is possible to configure the revert timer. If a primary path goes down and traffic is switched to the secondary path, the revert timer specifies the amount of time (in seconds) that the LSP must wait before it can revert traffic back to a primary path. If during this time, the primary path experiences any connectivity problems or stability problems, the timer is restarted. You can configure the revert timer for both static and dynamic LSPs.

The Junos OS also makes a determination as to which path is the preferred path. The preferred path is the path that has not encountered any difficulty in the last revert timer period. If both the primary and secondary paths have encountered difficulty, neither path is considered preferred. However, if one of the paths is dynamic and the other static, the dynamic path is selected as the preferred path.

If you have configured BFD on the LSP, Junos OS waits until the BFD session comes up on the primary path before starting the revert timer counter.

The range of values you can configure for the revert timer is 0 through 65,535 seconds. The default value is 60 seconds.

If you configure a value of 0 seconds, the traffic on the LSP, once switched from the primary path to the secondary path, remains on the secondary path permanently (until the network operator intervenes or until the secondary path goes down).

You can configure the revert timer for all LSPs on the router at the `[edit protocols mpls]` hierarchy level or for a specific LSP at the `[edit protocols mpls label-switched-path` *`lsp-name`*`]` hierarchy level.

To configure the revert timer, include the `revert-timer` statement:

```
revert-timer
seconds;
```

For a list of hierarchy levels at which you can include this statement, see the summary section for this statement.

Specifying the Conditions for Path Selection

When you have configured both primary and secondary paths for an LSP, you may need to ensure that only a specific path is used.

The `select` statement is optional. If you do not include it, MPLS uses an automatic path selection algorithm.

The `manual` and `unconditional` options do the following:

- `manual`—The path is immediately selected for carrying traffic as long as it is up and stable. Traffic is sent to other working paths if the current path is down or degraded (receiving errors). This parameter overrides all other path attributes except the `select unconditional` statement.
- `unconditional`—The path is selected for carrying traffic unconditionally, regardless of whether the path is currently down or degraded (receiving errors). This parameter overrides all other path attributes.

  Because the `unconditional` option switches to a path without regard to its current status, be aware of the following potential consequences of specifying it:

  - If a path is not currently up when you enable the `unconditional` option, traffic can be disrupted. Ensure that the path is functional before specifying the `unconditional` option.
  - Once a path is selected because it has the `unconditional` option enabled, all other paths for the LSP are gradually cleared, including the primary and standby paths. No path can act as a standby to an unconditional path, so signaling those paths serves no purpose.

For a specific path, the `manual` and `unconditional` options are mutually exclusive. You can include the `select` statement with the `manual` option in the configuration of only one of an LSP's paths, and the `select` statement with the `unconditional` option in the configuration of only one other of its paths.

Enabling or disabling the `manual` and `unconditional` options for the `select` statement while LSPs and their paths are up does not disrupt traffic.

To specify that a path be selected for carrying traffic if it is up and stable for at least the revert timer window, include the `select` statement with the `manual` option:

```
select manual;
```

To specify that a path should always be selected for carrying traffic, even if it is currently down or degraded, include the `select` statement with the `unconditional` option:

```
select unconditional;
```

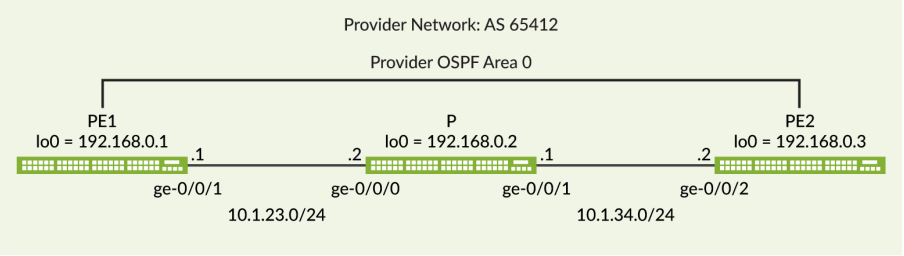
You can include the `select` statement at the following hierarchy levels:

- `[edit protocols mpls label-switched-path` *`lsp-name`* `(primary | secondary)` *`path-name`*`]`
- `[edit logical-systems` *`logical-system-name`* `protocols mpls label-switched-path` *`lsp-name`* `(primary | secondary)` *`path-name`*`]`

Configure a Primary Path

Follow these steps to configure a primary path with an ERO list, bandwidth, and priority. Refer to Figure 46 on page 687 to see how the sample configuration relates to a network topology.

Figure 46: Primary Path Topology



1. In configuration mode, position yourself at the `protocols mpls` hierarchy level:

```
[edit]
user@R1# edit protocols mpls
```

2. Configure the primary ERO list:

```
[edit protocols mpls]
user@R1# set path via-r2 10.1.23.2 strict
user@R1# set path via-r2 10.1.34.2 strict
```

3. Configure the LSP:

```
[edit protocols mpls]
user@R1# set label-switched-path pe1-pe2 to 192.168.0.3;
```

4. Configure the primary path:

```
[edit protocols mpls]
user@R1# set label-switched-path pe1-pe2 primary via-p1
```

5. Configure the bandwidth:

```
[edit protocols mpls]
user@R1# set label-switched-path pe1-pe2 primary via-p1 bandwidth 35m
```

6. Configure the priority value:

```
[edit protocols mpls]
user@R1# set label-switched-path pe1-pe2 primary via-p1 priority 6 6
```

7. Display the changes:

```
[edit protocols mpls]
user@R1# show
label-switched-path pe1-pe2 {
    to 192.168.0.3;
    primary via-p1 {
        bandwidth 35m;
        priority 6 6;
    }
}
```

```
path via-p1 {
    10.1.23.2 strict;
    10.1.34.2 strict;
}
```

Be sure to commit the changes when done. For a complete example of MPLS LSPs configured to support an MPLS-Based Layer 3 VPN see Example: Configure a Basic MPLS-Based Layer 3 VPN.

Configuring Hot Standby of Secondary Paths for LSPs

By default, secondary paths are set up only as needed. To have the system maintain a secondary path in a hot-standby state indefinitely, include the `standby` statement:

```
standby;
```

You can include this statement at the following hierarchy levels:

- `[edit protocols mpls label-switched-path` *`lsp-name`* `secondary]`
- `[edit logical-systems` *`logical-system-name`* `protocols mpls label-switched-path` *`lsp-name`* `secondary]`

The hot-standby state is meaningful only on secondary paths. Maintaining a path in a hot-standby state enables swift cutover to the secondary path when downstream routers on the current active path indicate connectivity problems. Although it is possible to configure the `standby` statement at the `[edit protocols mpls label-switched-path` *`lsp-name`* `primary` *`path-name`*`]` hierarchy level, it has no effect on router behavior.

If you configure the `standby` statement at the following hierarchy levels, the hot-standby state is activated on all secondary paths configured beneath that hierarchy level:

- `[edit protocols mpls]`
- `[edit protocols mpls label-switched-path` *`lsp-name`*`]`
- `[edit logical-systems` *`logical-system-name`* `protocols mpls]`
- `[edit logical-systems` *`logical-system-name`* `protocols mpls label-switched-path` *`lsp-name`*`]`

The hot-standby state has two advantages:

- It eliminates the call-setup delay during network topology changes. Call setup can suffer from significant delays when network failures trigger large numbers of LSP reroutes at the same time.
- A cutover to the secondary path can be made before RSVP learns that an LSP is down. There can be significant delays between the time the first failure is detected by protocol machinery (which can be an interface down, a neighbor becoming unreachable, a route becoming unreachable, or a transient

routing loop being detected) and the time an LSP actually fails (which requires a timeout of soft state information between adjacent RSVP routers). When topology failures occur, hot-standby secondary paths can usually achieve the smallest cutover delays with minimal disruptions to user traffic.

When the primary path is considered to be stable again, traffic is automatically switched from the standby secondary path back to the primary path. The switch is performed no faster than twice the retry-timer interval and only if the primary path exhibits stability throughout the entire switch interval.

The drawback of the hot-standby state is that more state information must be maintained by all the routers along the path, which requires overhead from each of the routers.

NOTE: When viewed with `inet.3`, the same LSP may appear to be shown twice as the active route (both primary and secondary), even though traffic actually is being forwarded over the primary path LSP only. This is normal output, and reflects only that the secondary standby path is available.

Configuring Static LSPs

IN THIS SECTION

- Configuring the Ingress Router for Static LSPs | **690**
- Configuring the Transit and Penultimate Routers for Static LSPs | **695**
- Configuring a Bypass LSP for the Static LSP | **700**
- Configuring the Protection Revert Timer for Static LSPs | **700**
- Configuring Static Unicast Routes for Point-to-Multipoint LSPs | **701**

To configure static LSPs, configure the ingress router and each router along the path up to and including the penultimate router.

To configure static MPLS, perform the following tasks:

Configuring the Ingress Router for Static LSPs

The ingress router checks the IP address in the incoming packet's destination address field and, if it finds a match in the routing table, applies the label associated with that address to the packets. The label has forwarding information associated with it, including the address of the next-hop router, and the route preference and CoS values.

To configure static LSPs on the ingress router, include the ingress statement:

```
ingress {
    bandwidth bps;
    class-of-service cos-value;
    description string;
    install {
        destination-prefix <active>;
    }
    link-protection bypass-name name;
    metric metric;
    next-hop (address | interface-name | address/interface-name);
    no-install-to-address;
    node-protection bypass-name name next-next-label label;
    policing {
        filter filter-name;
        no-auto-policing;
    }
    preference preference;
    push out-label;
    to address;
}
```

You can include these statements at the following hierarchy levels:

- [edit protocols mpls static-label-switched-path *static-lsp-name*]
- [edit logical-systems *logical-system-name* protocols mpls static-label-switched-path *static-lsp-name*]

When you configure a static LSP on the ingress router, the next-hop, push, and to statements are required; the other statements are optional.

The configuration for a static LSP on the ingress router includes the following:

- Criteria for analyzing an incoming packet:
  - The install statement creates an LSP that handles IPv4 packets. All static MPLS routes created using the install statement are installed in inet.3 routing table, and the creating protocol is identified as mpls. This process is no different from creating static IPv4 routes at the [edit routing-options static] hierarchy level.
  - In the to statement, you configure the IP destination address to check when incoming packets are analyzed. If the address matches, the specified outgoing label (push *out-label*) is assigned to the

packet, and the packet enters an LSP. Manually assigned outgoing labels can have values from 0 through 1,048,575. This IP address is installed into inet.3 table (by default) by the mpls protocol.

- The `next-hop` statement, which supplies the IP address of the next hop to the destination. You can specify this as the IP address of the next hop, the interface name (for point-to-point interfaces only), or as *address*| *interface-name* to specify an IP address on an operational interface. When the next hop is on a directly attached interface, the route is installed in the routing table. You cannot configure a LAN or nonbroadcast multiaccess (NBMA) interface as a next-hop interface.
- Properties to apply to the LSP (all are optional):
  - Bandwidth reserved for this LSP (`bandwidth` *bps*)
  - Link protection and node protection to apply to the LSP (`bypass` *bypass-name*, `link-protection` bypass-name *name*, `node-protection` bypass-name next-next-label *label*)
  - Metric value to apply to the LSP (`metric`)
  - Class-of-service value to apply to the LSP (`class-of-service`)
  - Preference value to apply to the LSP (`preference`)
  - Traffic policing to apply to the LSP (`policing`)
  - Text description to apply to the LSP (`description`)
  - Install or no-install policy (`install` or `no-install-to-address`)

To determine whether a static ingress route is installed, use the command `show route table inet.0 protocol static`. You can also see the route in table inet.3. The sample output uses the command `show route 10.1.45.2` to show both tables inet.0 and inet.3. The `Push` keyword denotes that a label is to be added in front of an IP packet.

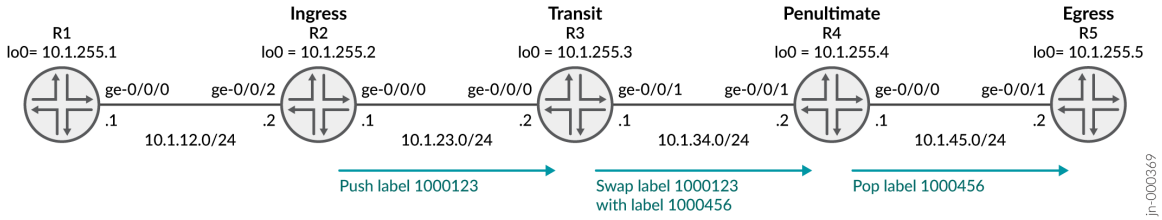
```
user@R2> show route 10.1.45.2

inet.0: 17 destinations, 17 routes (17 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.45.2/32       *[Static/5] 00:48:38
                    >  to 10.1.23.2 via ge-0/0/0.0, Push 1000123

inet.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
10.1.45.2/32       *[MPLS/6/1] 00:48:38, metric 0
                    >  to 10.1.23.2 via ge-0/0/0.0, Push 1000123
```

Example: Configuring the Ingress Router

Configure the ingress router for a static LSP that consists of four routers (see Figure 47 on page 693).

Figure 47: Static MPLS Configuration



NOTE: This example does not cover the R1 and R5 configurations. R1 and R5 have interface configuration and a static route to reach the other routers.

For packets addressed to `10.1.45.2`, assign label `1000123` and transmit them to the next-hop router at `10.1.23.2`:

```
[edit]
user@R2# show
interfaces {
    ge-0/0/0 {
        unit 0 {
            family inet {
                address 10.1.23.1/24;
            }
            family mpls;
        }
    }
    ge-0/0/2 {
        unit 0 {
            family inet {
                address 10.1.12.2/24;
            }
```

```
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.1.255.2/32;
            }
        }
    }
}
routing-options {
    router-id 10.1.255.2;
    static {
        route 10.1.45.2/32 {
            static-lsp-next-hop path1;
        }
    }
}
protocols {
    mpls {
        interface ge-0/0/0.0;
        static-label-switched-path path1 {
            ingress {
                next-hop 10.1.23.2;
                to 10.1.45.2;
                push 1000123;
            }
        }
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface ge-0/0/0.0;
            interface ge-0/0/2.0 {
                passive;
            }
            interface lo0.0;
        }
    }
}
```

To determine whether the static ingress route is installed, use the command `show route 10.1.45.2`.

The sample output shows the `Push 1000123` keyword identifies the route.

```
user@R2> show route 10.1.45.2

inet.0: 17 destinations, 17 routes (17 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.45.2/32       *[Static/5] 01:08:05
                    >  to 10.1.23.2 via ge-0/0/0.0, Push 1000123

inet.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.45.2/32       *[MPLS/6/1] 01:08:05, metric 0
                    >  to 10.1.23.2 via ge-0/0/0.0, Push 1000123
```

Configuring the Transit and Penultimate Routers for Static LSPs

The transit and penultimate routers perform similar functions—they modify the label that has been applied to a packet. An transit router can change the label. An penultimate router removes the label and continues forwarding the packet to its destination.

To configure static LSPs on transit and penultimate routers, include the `transit` statement:

```
static-label-switched-path lsp-name {
    transit incoming-label {
        bandwidth bps;
        description string;
        link-protection bypass-name name;
        next-hop (address | interface-name | address/interface-name);
        node-protection bypass-name name next-next-label label;
        pop;
        swap out-label;
    }
```

You can include these statements at the following hierarchy levels:

- `[edit protocols mpls static-label-switched-path static-lsp-name]`
- `[edit logical-systems logical-system-name protocols mpls static-label-switched-path static-lsp-name]`

For the `transit` statement configuration, the `next-hop` and `pop | swap` statements are required. The remaining statements are optional.

Each statement within the `transit` statement consists of the following parts:

- Packet label (specified in the `transit` statement)
- The `next-hop` statement, which supplies the IP address of the next hop to the destination. The address is specified as the IP address of the next hop, or the interface name (for point-to-point interfaces only), or *address* and *interface-name* to specify an IP address on an operational interface. When the specified next hop is on a directly attached interface, this route is installed in the routing table. You cannot configure a LAN or NBMA interface as a next-hop interface.
- Operation to perform on the labeled packet:
  - For penultimate routers, you generally just remove the packet's label altogether (`pop`) and continue forwarding the packet to the next hop. However, if the previous router removed the label, the egress router examines the packet's IP header and forwards the packet toward its IP destination.
  - For transit routers only, exchange the label for another label (`swap` *out-label*). Manually assigned incoming labels can have values from 1,000,000 through 1,048,575. Manually assigned outgoing labels can have values from 0 through 1,048,575.
- Label properties to apply to the packet (all are optional):
  - Bandwidth reserved for this route (`bandwidth` *bps*).
  - Link-protection and node-protection to apply to the LSP (`bypass` *bypass-name*, `link-protection` bypass-name *name*, `node-protection` bypass-name next-next-label *label*).
  - Text description to apply to the LSP (specified in the `description` statement).

The routes are installed in the default MPLS routing table, mpls.0, and the creating protocol is identified as MPLS. To verify that a route is properly installed, use the command `show route table mpls.0`. Sample output follows:

```
root@R3> show route table mpls.0
...
1000123            *[MPLS/6] 00:51:34, metric 1
                    >  to 10.1.34.2 via ge-0/0/1.0, Swap 1000456
```

You can configure a revert timer for a static LSP transiting a transit router. After traffic has been switched to a bypass static LSP, it is typically switched back to the primary static LSP when it comes back up. There is a configurable delay in the time (called the revert timer) between when the primary static LSP comes up and when traffic is reverted back to it from the bypass static LSP. This delay is needed because when the primary LSP comes back up, it is not certain whether all of the interfaces on

the downstream node of the primary path have come up yet. You can display the revert timer value for an interface using the `show mpls interface detail` command.

Example: Configuring a Transit Router

For packets labeled `1000123` arriving on interface `ge-0/0/0`, assign the label `1000456`, and transmit them to the next-hop router at `10.1.34.2`:

```
[edit]
user@R3# show
interfaces {
    ge-0/0/0 {
        unit 0 {
            family inet {
                address 10.1.23.2/24;
            }
            family mpls;
        }
    }
    ge-0/0/1 {
        unit 0 {
            family inet {
                address 10.1.34.1/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.1.255.3/32;
            }
        }
    }
}
routing-options {
    router-id 10.1.255.3;
}
protocols {
    mpls {
        interface ge-0/0/0.0;
        interface ge-0/0/1.0;
```

```
        static-label-switched-path path1 {
            transit 1000123 {
                next-hop 10.1.34.2;
                swap 1000456;
            }
        }
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface ge-0/0/0.0;
            interface ge-0/0/1.0;
            interface lo0.0;
        }
    }
}
```

To determine whether the route is installed, use the command `show route table mpls.0`.

Sample output follows. The `Swap 1000456` keyword identifies the route.

```
root@R3> show route table mpls.0
...
1000123            *[MPLS/6] 00:57:17, metric 1
                    >  to 10.1.34.2 via ge-0/0/1.0, Swap 1000456
```

Example: Configuring a Penultimate Router

For packets labeled `1000456` arriving on interface `ge-0/0/1`, remove the label and transmit the packets to the next-hop router at `10.1.45.2`:

```
[edit]
user@R4# show
interfaces {
    ge-0/0/0 {
        unit 0 {
            family inet {
                address 10.1.45.1/24;
            }
            family mpls;
        }
```

```
    }
    ge-0/0/1 {
        unit 0 {
            family inet {
                address 10.1.34.2/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.1.255.4/32;
            }
        }
    }
}
routing-options {
    router-id 10.1.255.4;
}
protocols {
    mpls {
        interface ge-0/0/1.0;
        interface ge-0/0/0.0;
        static-label-switched-path path1 {
            transit 1000456 {
                next-hop 10.1.45.2;
                pop;
            }
        }
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface ge-0/0/1.0;
            interface lo0.0;
            interface ge-0/0/0.0;
        }
    }
}
```

To determine whether the route is installed, use the command `show route table mpls.0`.

Sample output follows. The Pop keyword identifies the route.

```
user@R4> show route table mpls.0
...
1000456            *[MPLS/6] 00:50:55, metric 1
                    >  to 10.1.45.2 via ge-0/0/0.0, Pop
1000456(S=0)       *[MPLS/6] 00:50:55, metric 1
                    >  to 10.1.45.2 via ge-0/0/0.0, Pop
```

To verify end-to-end reachability and that the traffic is using the LSP, use the command traceroute 10.1.45.2 on R1.

```
user@R1> traceroute 10.1.45.2
traceroute to 10.1.45.2 (10.1.45.2), 30 hops max, 52 byte packets
 1  10.1.12.2 (10.1.12.2)  2.601 ms  2.261 ms  2.172 ms
 2  10.1.23.2 (10.1.23.2)  3.953 ms  3.425 ms  3.928 ms
     MPLS Label=1000123 CoS=0 TTL=1 S=1
 3  10.1.34.2 (10.1.34.2)  4.616 ms  4.300 ms  4.535 ms
     MPLS Label=1000456 CoS=0 TTL=1 S=1
 4  10.1.45.2 (10.1.45.2)  5.965 ms  5.232 ms  5.289 ms
```

Configuring a Bypass LSP for the Static LSP

To enable a bypass LSP for the static LSP, configure the bypass statement:

```
bypass bypass-name {
    bandwidth bps;
    description string;
    next-hop (address | interface-name | address/interface-name);
    next-table
    push out-label;
    to address;
}
```

Configuring the Protection Revert Timer for Static LSPs

For static LSPs configured with a bypass static LSP, it is possible to configure the protection revert timer. If a static LSP goes down and traffic is switched to the bypass LSP, the protection revert timer specifies

the amount of time (in seconds) that the LSP must wait before it can revert back to the original static LSP.

The range of values you can configure for the protection revert timer is 0 through 65,535 seconds. The default value is 5 seconds.

If you configure a value of 0 seconds, the traffic on the LSP, once switched from the original static LSP to the bypass static LSP, remains on the bypass LSP permanently (until the network operator intervenes or until the bypass LSP goes down).

You can configure the protection revert timer for all dynamic LSPs on the router at the `[edit protocols mpls]` hierarchy level or for a specific LSP at the `[edit protocols mpls label-switched-path lsp-name]` hierarchy level.

To configure the protection revert timer for static LSPs include the `protection-revert-time` statement:

```
protection-revert-time seconds;
```

For a list of hierarchy levels at which you can include this statement, see the summary section for this statement.

Configuring Static Unicast Routes for Point-to-Multipoint LSPs

You can configure a static unicast IP route with a point-to-multipoint LSP as the next hop. For more information about point-to-multipoint LSPs, see "Point-to-Multipoint LSPs Overview" on page 781, "Configuring Primary and Branch LSPs for Point-to-Multipoint LSPs" on page 811, and "Configuring CCC Switching for Point-to-Multipoint LSPs" on page 2094.

To configure a static unicast route for a point-to-multipoint LSP, complete the following steps:

1. On the ingress PE router, configure a static IP unicast route with the point-to-multipoint LSP name as the next hop by including the `p2mp-lsp-next-hop` statement:

   ```
   p2mp-lsp-next-hop point-to-multipoint-lsp-next-hop;
   ```

   You can include this statement at the following hierarchy levels:

   - `[edit routing-options static route route-name]`
   - `[edit logical-systems logical-system-name routing-options static route route-name]`

2. On the egress PE router, configure a static IP unicast route with the same destination address configured in Step "1" on page 701 (the address configured at the `[edit routing-options static route]` hierarchy level) by including the `next-hop` statement:

```
next-hop address;
```

   You can include this statement at the following hierarchy levels:

   - `[edit routing-options static route route-name]`
   - `[edit logical-systems logical-system-name routing-options static route route-name]`

> **NOTE**: CCC and static routes cannot use the same point-to-multipoint LSP.

For more information on static routes, see the Junos OS Routing Protocols Library for Routing Devices.

The following `show route` command output displays a unicast static route pointing to a point-to-multipoint LSP on the ingress PE router where the LSP has two branch next hops:

```
user@host> show route 5.5.5.5 detail
inet.0: 29 destinations, 30 routes (28 active, 0 holddown, 1 hidden)
5.5.5.5/32 (1 entry, 1 announced)
        *Static Preference: 5
                Next hop type: Flood
                Next hop: via so-0/3/2.0 weight 1
                Label operation: Push 100000
                Next hop: via t1-0/1/1.0 weight 1
                Label operation: Push 100064
                State: <Active Int Ext>
                Local AS: 10458
                Age: 2:41:15
                Task: RT
                Announcement bits (2): 0-KRT 3-BGP.0.0.0.0+179
                AS path: I
```

Configuring Static Label Switched Paths for MPLS (CLI Procedure)

IN THIS SECTION

- Configuring the Ingress PE Switch | 703
- Configuring the Provider and the Egress PE Switch | 705

Configuring static label-switched paths (LSPs) for MPLS is similar to configuring static routes on individual switches. As with static routes, there is no error reporting, liveliness detection, or statistics reporting.

To configure static LSPs, configure the ingress switch and each provider switch along the path up to and including the egress switch.

For the ingress switch, configure which packets to tag (based on the packet's destination IP address), configure the next switch in the LSP, and the tag to apply to the packet. Manually assigned labels can have values from 0 through 1,048,575. Optionally, you can apply preference, class-of-service (CoS) values, node protection, and link protection to the packets.

For the transit switches in the path, configure the next switch in the path and the tag to apply to the packet. Manually assigned labels can have values from 1,000,000 through 1,048,575. Optionally, you can apply node protection and link protection to the packets.

For the egress switch, you generally just remove the label and continue forwarding the packet to the IP destination. However, if the previous switch removed the label, the egress switch examines the packet's IP header and forwards the packet toward its IP destination.

Before you configure an LSP, you must configure the basic components for an MPLS network:

- Configure two PE switches. See "Configuring MPLS on Provider Edge EX8200 and EX4500 Switches Using Circuit Cross-Connect" on page 89.
- Configure one or more provider switches. See "Configuring MPLS on EX8200 and EX4500 Provider Switches" on page 93.

This topic describes how to configure an ingress PE switch, one or more provider switches, and an egress PE switch for static LSP:

Configuring the Ingress PE Switch

To configure the ingress PE switch:

1. Configure an IP address for the core interfaces:

```
[edit]
user@switch# set interfaces interface-name unit logical-unit-number family inet address
address
user@switch# set interfaces interface-name unit logical-unit-number family inet address
address
```

2. Configure the name and the traffic rate associated with the LSP:

```
[edit]
user@switch# set protocols mpls static-label-switched-path lsp-name ingress bandwidth rate
```

3. Configure the next hop switch for the LSP:

```
[edit]
user@switch# set protocols mpls static-label-switched-path lsp-name ingress next-hop address-
of-next-hop
```

4. Enable link protection on the specified static LSP:

```
[edit]
user@switch# set protocols mpls static-label-switched-path lsp-name ingress link-protection
bypass-name name
```

5. Specify the address of the egress switch for the LSP:

```
[edit]
user@switch# set protocols mpls static-label-switched-path path1 ingress to address-of-egress-
switch
```

6. Configure the new label that you want to add to the top of the label stack:

```
[edit]
user@switch# set protocols mpls static-label-switched-path path1 ingress push out-label
```

7. Optionally, configure the next hop address and the egress router address that you want to bypass, for the static LSP:

```
[edit]
user@switch# set protocols mpls static-label-switched-path lsp-name by bypass next-hop
address-of-next-hop
user@switch# set protocols mpls static-label-switched-path lsp-name by bypass to address-of-
the-egress-switch
user@switch# set protocols mpls static-label-switched-path lsp-name bypass push out-label
```

Configuring the Provider and the Egress PE Switch

To configure a static LSP for MPLS on the provider and egress provider edge switch:

1. Configure a transit static LSP:

```
[edit]
user@switch# set protocols mpls static-label-switched-path path1 transit incoming-label
```

2. Configure the next hop switch for the LSP:

```
[edit]
user@switch# set protocols mpls static-label-switched-path lsp-name transit incoming-label
next-hop address-of-next-hop
```

3. Only for provider switches, remove the label at the top of the label stack and replace it with the specified label:

```
[edit]
user@switch# set protocols mpls static-label-switched-path lsp-name transit incoming-label
swap out-label
```

4. Only for the egress provider edge switch, remove the label at the top of the label stack:

NOTE: If there is another label in the stack, that label becomes the label at the top of the label stack. Otherwise, the packet is forwarded as a native protocol packet (typically, as an IP packet).

```
[edit]
user@switch# set protocols mpls static-label-switched-path lsp-name transit incoming-label pop
```

Configuring Static Label Switched Paths for MPLS

IN THIS SECTION

- Configuring the Ingress PE Switch | **707**
- Configuring the Provider and the Egress PE Switch | **708**

Configuring static label-switched paths (LSPs) for MPLS is similar to configuring static routes on individual switches. As with static routes, there is no error reporting, liveliness detection, or statistics reporting.

To configure static LSPs, configure the ingress PE switch and each provider switch along the path up to and including the egress PE switch.

For the ingress PE switch, configure which packets to tag (based on the packet's destination IP address), configure the next switch in the LSP, and the tag to apply to the packet. Manually assigned labels can have values from 0 through 1,048,575.

For the transit switches in the path, configure the next switch in the path and the tag to apply to the packet. Manually assigned labels can have values from 1,000,000 through 1,048,575.

The egress PE switch removes the label and forwards the packet to the IP destination. However, if the previous switch removed the label, the egress switch examines the packet's IP header and forwards the packet toward its IP destination.

Before you configure a static LSP, you must configure the basic components for an MPLS network:

- Configure two PE switches. See "Configuring MPLS on Provider Edge Switches" on page 78.

NOTE: Do not configure LSPs at the `[edit protocols mpls label-switched-path]` hierarchy level on the PE switches.

- Configure one or more provider switches. See "Configuring MPLS on Provider Switches" on page 76.

This topic describes how to configure an ingress PE switch, one or more provider switches, and an egress PE switch for static LSP:

Configuring the Ingress PE Switch

To configure the ingress PE switch:

1. Configure an IP address for every core interface:

```
[edit interfaces]
user@switch# set interface-name unit logical-unit-number family inet address address
```

NOTE: You cannot use routed VLAN interfaces (RVIs) or Layer 3 subinterfaces as core interfaces.

2. Configure the name associated with the static LSP:

```
[edit protocols mpls]
user@switch# set static-label-switched-path lsp-name
```

3. Configure the next hop switch for the LSP:

```
[edit protocols mpls]
user@switch# set static-label-switched-path lsp-name ingress next-hop address-of-next-hop
```

4. Specify the address of the egress switch for the LSP:

```
[edit protocols mpls]
user@switch# set static-label-switched-path lsp-name ingress to address-of-egress-switch
```

5. Configure the new label that you want to add to the top of the label stack:

```
[edit protocols mpls]
user@switch# set static-label-switched-path lsp-name ingress push out-label
```

Configuring the Provider and the Egress PE Switch

To configure a static LSP for MPLS on the provider and egress PE switch:

1. Configure a transit static LSP:

```
[edit protocols mpls]
user@switch# set static-label-switched-path lsp-name transit incoming-label
```

2. Configure the next hop switch for the LSP:

```
[edit protocols mpls]
user@switch# set static-label-switched-path lsp-name transit incoming-label next-hop address-of-next-hop
```

3. Only for provider switches, remove the label at the top of the label stack and replace it with the specified label:

```
[edit protocols mpls]
user@switch# set static-label-switched-path lsp-name transit incoming-label swap out-label
```

4. Only for the egress PE switch, remove the label at the top of the label stack:

> **NOTE**: If there is another label in the stack, that label becomes the label at the top of the label stack. Otherwise, the packet is forwarded as a native protocol packet (typically, as an IP packet).

```
[edit protocols mpls]
user@switch# set static-label-switched-path lsp-name transit incoming-label pop
```

RELATED DOCUMENTATION

MPLS Overview | 2

Adaptive LSP Configuration

An LSP occasionally might need to reroute itself for these reasons:

- The continuous reoptimization process is configured with the `optimize-timer` statement.
- The current path has connectivity problems.
- The LSP is preempted by another LSP configured with the `priority` statement and is forced to reroute.
- The explicit-path information for an active LSP is modified, or the LSP's bandwidth is increased.

You can configure an LSP to be *adaptive* when it is attempting to reroute itself. When it is adaptive, the LSP holds onto existing resources until the new path is successfully established and traffic has been cut over to the new LSP. To retain its resources, an adaptive LSP does the following:

- Maintains existing paths and allocated bandwidths—This ensures that the existing path is not torn down prematurely and allows the current traffic to continue flowing while the new path is being set up.
- Avoids double-counting for links that share the new and old paths—Double-counting occurs when an intermediate router does not recognize that the new and old paths belong to the same LSP and counts them as two separate LSPs, requiring separate bandwidth allocations. If some links are close to saturation, double-counting might cause the setup of the new path to fail.

By default, adaptive behavior is disabled. You can include the `adaptive` statement in two different hierarchy levels.

If you specify the `adaptive` statement at the LSP hierarchy levels, the adaptive behavior is enabled on all primary/secondary paths of the LSP. This means both the primary and secondary paths share the same bandwidth on common links.

To configure adaptive behavior for all LSP paths, include the `adaptive` statement in the LSP configuration:

```
adaptive;
```

You can include this statement at the following hierarchy levels:

- `[edit protocols mpls label-switched-path lsp-name]`
- `[edit logical-systems logical-system-name protocols mpls label-switched-path lsp-name]`

If you specify the `adaptive` statement at the `[edit protocols mpls label-switched-path lsp-name (primary | secondary) path-name]` hierarchy level, adaptive behavior is enabled only on the path on which it is specified. Bandwidth double-counting occurs between different paths. However, if you also have the

adaptive statement configured at the [edit protocols mpls label-switched-path *lsp-name*] hierarchy level, it overrides the adaptive behavior of each individual path.

To configure adaptive behavior for either the primary or secondary level, include the adaptive statement:

```
adaptive;
```

You can include this statement at the following hierarchy levels:

- [edit protocols mpls label-switched-path *lsp-name* (primary | secondary) *path-name*]
- [edit logical-systems *logical-system-name* protocols mpls label-switched-path *lsp-name* (primary | secondary) *path-name*]

Container LSP Configuration

IN THIS SECTION

- Dynamic Bandwidth Management Using Container LSP Overview | **710**
- Example: Configuring Dynamic Bandwidth Management Using Container LSP | **742**
- Configuring Dynamic Bandwidth Management Using Container LSP | **774**

Dynamic Bandwidth Management Using Container LSP Overview

IN THIS SECTION

- Understanding RSVP Multipath Extensions | **711**
- Junos OS RSVP Multipath Implementation | **712**
- Current Traffic Engineering Challenges | **712**
- Using Container LSP as a Solution | **716**
- Junos OS Container LSP Implementation | **718**
- Configuration Statements Supported for Container LSPs | **735**
- Impact of Configuring Container LSPs on Network Performance | **740**

RSVP LSPs with the autobandwidth feature are increasingly deployed in networks to meet traffic engineering needs. However, the current traffic engineering solutions for point-to-point LSPs are inefficient in terms of network bandwidth utilization, mainly because the ingress routers originating the RSVP LSPs either try to fit the LSPs along a particular path without creating parallel LSPs, or do not interact with the other routers in the network and probe for additional available bandwidth.

This feature provides an ingress router with the capability of acquiring as much network bandwidth as possible by creating parallel LSPs dynamically.

Understanding RSVP Multipath Extensions

The RSVP multipath extensions proposed in the IETF [KOMPELLA-MLSP] allow the setup of traffic engineered multipath label-switched paths (container LSPs). The container LSPs, in addition to conforming to traffic engineering constraints, use multiple independent paths from a source to a destination, thereby facilitating load balancing of traffic. The multipath extensions require changes to the RSVP-TE protocol and allow for merging of labels at the downstream nodes (similar to LDP), which also helps in preserving forwarding resources.

The multipath extensions to RSVP provide the following benefits:

- Ease of configuration. Typically, multiple RSVP LSPs are configured for either load balancing or bin packing. With a container LSP, there is a single entity to provision, manage, and monitor LSPs. Changes in topology are handled easily and autonomously by the ingress LSP, by adding, changing, or removing member LSPs to rebalance traffic, while maintaining the same traffic engineering constraints.
- RSVP equal-cost multipath (ECMP) inherits the standard benefits of ECMP by absorbing traffic surges.
- Multipath traffic engineering allows for better and complete usage of network resources.
- Knowing the relationship among LSPs helps in computing diverse paths with constraint-based routing. It allows adjustment of member LSPs while other member LSPs continue to carry traffic.
- The intermediate routers have an opportunity to merge the labels of member LSPs. This reduces the number of labels that need to get added to the forwarding plane and in turn reduces the convergence time.

  If the number of independent ECMP paths is huge, label merging overcomes the platform limitations on maximum (ECMP) next hops. With point-to-point RSVP LSPs that require link or node protection,

the next hops are doubled as each LSP is programmed with both primary and backup next hops. RSVP multipath (or ECMP) obviates the need for backup next hops.

- When there is a link failure, the router upstream to the link failure can distribute traffic from the failed link to the remaining ECMP branches, obviating the need for bypass LSPs. The bypass LSP approach not only requires more state when signaling backup LSPs, but also suffers from scaling issues that result in merge-point timing out a protected path state block (PSB) before point of local repair (PLR) gets a chance to signal the backup LSP.

Junos OS RSVP Multipath Implementation

In order to deploy RSVP multipath (ECMP) in a network, all the nodes through which ECMP LSPs pass must understand RSVP ECMP protocol extensions. This can be a challenge, especially in a multivendor networks.

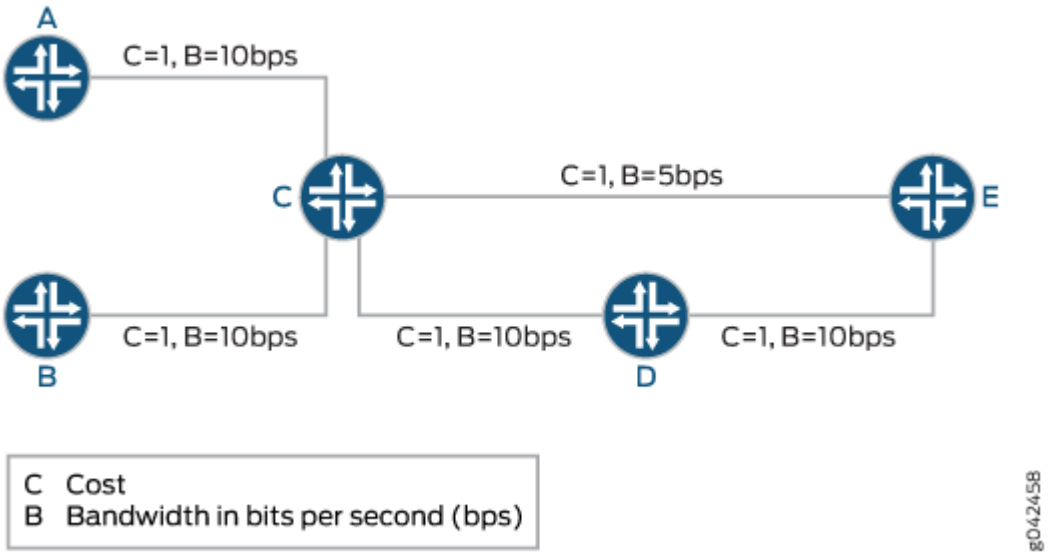
Junos OS implements the RSVP multipath extensions without the need for protocol extensions. A single container LSP, which has the characteristics of ECMP and RSVP TE, is provisioned. A container LSP consists of several member LSPs and is set up between the ingress and egress routing device. Each member LSP takes a different path to the same destination. The ingress routing device is configured with all the required parameters to compute the RSVP ECMP LSP. The parameters configured to compute a set of RSVP point-to-point LSPs can be used by the ingress routing device to compute the container LSP as well.

Current Traffic Engineering Challenges

The main challenge for traffic engineering is to cope with the dynamics of both topology and traffic demands. Mechanisms are needed that can handle traffic load dynamics in scenarios with sudden changes in traffic demand and dynamically distribute traffic to benefit form available resources.

Figure 48 on page 713 illustrates a sample network topology with all the LSPs having the same hold and setup priorities, and admission control restricted on the ingress router. All the links are annotated with a tuple (cost and capacity).

Figure 48: Sample Topology



Some of the traffic engineering problems seen in Figure 48 on page 713 are listed here:

- **Bin Packing**

  This problem arises because of a particular order in which LSPs are signaled. The ingress routers might not be able to signal some LSPs with required demands although bandwidth is available in the network, leading to under-utilization of link capacity.

  For example, the following LSPs arrive in the sequence mentioned in Table 15 on page 713.

  Table 15: LSP Sequence Order for Bin Packing

Time	Source	Destination	Demand	ERO
1	A	E	5	A-C-D-E
2	B	E	10	No ERO

  The LSP originating at Router B is not routable as constraint-based routing fails to find a feasible path. However, if Router B is signaled first, both the LSPs are routable. Bin packing happens because of lack of visibility of individual per-LSP, per-device bandwidth demands at the ingress routing device.

  Bin packing can also happen when there is no requirement for ordering of LSPs. For example, if there is an LSP with demand X and there are two different paths to the destination from the ingress router

with available bandwidths Y1 and Y2, such that Y1 is less than X, Y2 is less than X, and Y1 plus Y2 is greater than or equal to X.

In this case, even though there are enough network resources in terms of available bandwidth to satisfy the aggregate LSP demand X, the LSP might not be signaled or re-optimized with the new demand. In Figure 48 on page 713, with container LSP support, the ingress B creates two LSPs each of size 5 when demand 10 is posed. One LSP is routed along B-C-E and another one along B-C-D-E.

- `Deadlock`

  Considering Figure 48 on page 713, the LSPs follow the sequence mentioned in Table 16 on page 714.

  Table 16: LSP Sequence Order for Deadlock

Time	Source	Destination	Demand	ERO	Event
1	A	E	2	A-C-D-E	Constraint-based routing with RSVP signaling
2	B	E	2	B-C-D-E	Constraint-based routing with RSVP signaling
3	A	E	2 to 20	A-C-D-E	Constraint-based routing fails, no RSVP signaling

  At time 3, the demand on LSP from A to E increases from 2 to 20. If autobandwidth is configured, the change does not get detected until the adjustment timer expires. In the absence of admission control at A, the increased traffic demand might cause traffic to drop on other LSPs that share common links with the mis-behaving LSP.

  This happens due to the following reasons:

  - Lack of global state at all the ingress routers
  - Signaling of mis-behaving demands
  - Tearing down of mis-behaving demands

  With container LSP configured, ingress A has more chances of splitting the load (even incrementally if not fully) across multiple LSPs. So, LSP from A is less likely to see prolonged traffic loss.

- `Latency Inflation`

  Latency inflation is caused by the autobandwidth and other LSPs parameters. Some of the other factors that contribute to latency inflation include:

  - LSP priority

LSPs choose longer paths because shorter paths between data centers located in the same city can be congested. The bandwidth on the shorter paths can get exhausted by equal or higher priority LSPs. Due to periodic LSP optimization by autobandwidth, LSP can get rerouted to a higher delay path. When many LSPs undergo less than optimal path selection, they can potentially form a chain of dependencies. Modifing the LSP priorities dynamically is a workaround to the issue; however, dynamically adjusting LSP priorities to find shorter paths is a challenging task.

- All or Nothing policy

  When the demand on an LSP increases and at least one of the links along the shorter path is close to its reservation limit, LSP optimization can force the LSP to move to a longer latency path. LSP has to traverse a long path even though the short path is capable of carrying most of the traffic.

- Minimum and maximum bandwidth

  Minimum and maximum bandwidth specify the boundaries for LSP sizes. If minimum bandwidth is small, an LSP is more prone to autobandwidth adjustment because a small change in bandwidth is enough to cross the threshold limits. LSPs might reroute although bandwidth is available. On the other hand, if the minimum bandwidth is large, network bandwidth might be wasted. If the maximum bandwidth value is small, a large number of LSPs might be needed at the ingress router to accommodate the application demand. If the maximum bandwidth is large, the LSPs can grow larger in size. Such LSPs can suffer because of an all or nothing policy.

- Autobandwidth adjustment threshold

  Bandwidth threshold dictates if LSPs need to be re-optimized and resized. If the value is small, LSPs are frequently re-optimized and rerouted. That might cause CPU spike because applications or protocols, such as BGP resolving over the LSPs, might keep the Routing Engine busy doing next-hop resolution. A large value might make an LSP immobile. With container LSP configured, an LSP is less likely to get subjected to one or no policy. An ingress router originates multiple LSPs, although not all LSPs potentially traverse high latency paths.

- `Predictability`

  Service providers often want predictable behavior in terms of how LSPs get signaled and routed. Currently, without any global coordination, it is difficult to set up the same set of LSPs in a predictable way. Consider the two different orderings in Table 17 on page 715 and Table 18 on page 716. The ERO that an LSP uses depends on its signaling time.

  Table 17: LSP Sequence Order for Predictability

Time	Source	Destination	Demand	ERO
1	A	E	5	A-C-D-E

Table 17: LSP Sequence Order for Predictability *(Continued)*

Time	Source	Destination	Demand	ERO
2	B	E	5	B-C-E

Table 18: LSP Sequence Order for Predictability

Time	Source	Destination	Demand	ERO
1	B	E	5	B-C-E
2	A	E	5	A-C-D-E

Container LSP does not directly help LSPs find predictable EROs. If LSPs are getting rerouted because of an all or no policy without container LSP configured, such LSPs might see less churn if container LSPs are configured, because smaller LSPs have better chances of finding a shorter or same path.

Using Container LSP as a Solution

A container LSP can be used as a solution to the challenges faced by the current traffic engineering features. Considering Figure 48 on page 713, when the demand X on a container LSP increases with the network capacity (max-flow) being more than the demand, the following approaches come into effect with a container LSP:

Accommodating the New Demand X

In the current implementation, autobandwidth attempts to re-signal an LSP with the new demand X and follows the all or nothing policy as mentioned earlier.

The container LSP approach computes several small (smaller than demand X) bandwidth LSPs such that the aggregate bandwidth is not less than X, and the ingress router performs this adjustment periodically. One of the triggers to create new LSPs or to delete old LSPs can be changed in aggregate bandwidth. The ingress router then load-balances the incoming traffic across the newly created LSPs.

Creating New LSPs to Meet Demand X

Although the number of new LSPs created can be a maximum of the allowed configurable limit, there is not much benefit from these LSPs once the number of LSPs exceeds the number of possible diverse paths or equal-cost multipaths (ECMPs). The benefit of creating the smaller LSPs is seen when an ingress

router uses the newly created LSPs for load-balancing traffic. This, however, depends on the network topology and state.

Creating multiple parallel LSPs by all the ingress routers in the network can lead to scaling issues at the transit routers. Thus, the number of new LSPs to be created depends on the size of the individual LSPs and the given aggregate demand, X in this case.

Assigning Bandwidth to the New LSPs

In general, there can be a number of heuristics to allocate bandwidths to the newly created LSPs. An ingress router can solve an optimization problem in which it can maximize a given utility function. The output of an optimization problem is assigning optimal bandwidth values. However, to solve an optimization problem, the number of newly created LSPs has to be fixed. Therefore, it is complex to optimize the number and size of each LSP. Thus, to simplify the problem, the same amount of bandwidth is assumed for all the newly created LSPs, and then the number of required LSPs is computed.

Controlling the LSP Paths

The flexibility to control the LSP paths is expressed in terms of the configuration for point-to-point LSPs and container LSPs. Controlling the LSP paths using the configuration parameters can be applied under two different aspects:

- Topology—There are no topology constraints with this feature. Each member LSP is treated like a point-to-point LSP and is re-optimized individually. An ingress router does not try to compute equal IGP cost paths for all its LSPs, but instead it computes paths for all the LSPs using current traffic engineering database information. While computing a path, constraint-based routing adheres to any constraints specified through the configuration, although there is no change in the constraint-based routing method for path computation.

- When to create a new LSP—When to create a new LSP can be explicitly specified. By default, an ingress router periodically computes the aggregate traffic rate by adding up the traffic rate of all the individual LSPs. Looking at the aggregate bandwidth and configuration, the ingress router recomputes the number of LSPs and the bandwidths of the LSPs. The new LSPs are then signaled or the existing LSPs are re-signaled with the updated bandwidth. Instead of looking at the instantaneous aggregate rate, the ingress routers can compute an average (of aggregates) over some duration by removing outlier samples (of aggregates). Managing the LSPs that remain outstanding and active by considering aggregate bandwidth is more scalable than creating the new LSPs based on the usage of a particular LSP. The intervals and thresholds can be configured to track the aggregate traffic and trigger adjustment. These dynamic LSPs co-exist and interoperate with per-LSP autobandwidth configuration.

Junos OS Container LSP Implementation

A container LSP is an ECMP TE LSP that acts like a container LSP consisting of one or more member LSPs. A point-to-point TE LSP is equivalent to a container LSP with a single member LSP. Member LSPs are added to the container LSP through a process called splitting, and removed from the container LSP through a process called merging.

Container LSP Terminology

The following terms are defined in the context of a container LSP:

- `Normalization`—An event occurring periodically when an action is taken to adjust the member LSPs, either to adjust their bandwidths, their number, or both. A normalization process is associated with a sampling process and periodically estimates aggregate utilization of a container LSP.
- `Nominal LSP`—The instance of a container LSP that is always present.
- `Supplementary LSP`—The instances or sub-LSPs of a container LSP, which are dynamically created or removed.

  Autobandwidth is run over each of the member LSPs, and each LSP is resized according to the traffic it carries and the autobandwidth configuration parameters. The aggregate demand on a container LSP is tracked by adding up the bandwidth across all the member LSPs.
- `Minimum signaling-bandwidth`—The minimum bandwidth with which a member LSP is signaled at the time of normalization or initialization. This could be different from the minimum-bandwidth defined under autobandwidth.
- `Maximum signaling-bandwidth` —The maximum bandwidth with which a member LSP is signaled at the time of normalization or initialization. This could be different from the maximum-bandwidth defined under autobandwidth.
- `Merging-bandwidth` —Specifies the lower bandwidth threshold on the aggregate bandwidth usage, such that if the aggregate usage falls below this value, the ingress router merges the member LSPs at the time of normalization.
- `Splitting-bandwidth` —Specifies the upper bandwidth threshold on the aggregate bandwidth usage, such that if the aggregate usage exceeds this value, the ingress router splits the member LSPs at the time of normalization.
- `Aggregate minimum-bandwidth` —Sum of merging-bandwidth of the current active member LSPs. This minimum bandwidth is different from the autobandwidth minimum-bandwidth.
- `Aggregate maximum-bandwidth`—Sum of the splitting-bandwidth of the current active member LSPs. This maximum bandwidth is different from the autobandwidth maximum-bandwidth.

LSP Splitting

Operational Overview

The LSP splitting mechanism enables an ingress router to create new member LSPs or to re-signal existing LSPs with different bandwidths within a container LSP when a demand X is placed on the container LSP. With LSP splitting enabled, an ingress router periodically creates a number of LSPs (by signaling new ones or re-signaling existing ones) to accommodate a new aggregate demand X. In the current implementation, an ingress router tries to find an LSP path satisfying a demand X and other constraints. If no path is found, either the LSP is not signaled or it remains up, but with the old reserved bandwidth.

Between two normalization events (splitting or merging), individual LSPs might get re-signaled with different bandwidths due to the autobandwidth adjustments. If a container LSP is not configured with autobandwidth, then the member LSPs are signaled with the static bandwidth value, if configured. There is no dynamic splitting in this case, as there is no dynamic estimation of aggregate bandwidth. The splitting adjustments with a specific bandwidth value can be manually triggered.

NOTE: Be aware of the following considerations for LSP splitting:

- After LSP splitting, the ingress router continues to inject one forwarding adjacency. Forwarding adjacencies are not supported in IGP for this feature.
- Between two normalization events, two LSPs might have different bandwidths subjected to autobandwidth constraints.
- After LSPs are split (or merged), make-before-break uses the fixed filter (FF) style sharing unless the `adaptive` option is configured. However, two different LSPs do not do the shared explicit (SE) style sharing for this feature.
- When LSPs are re-signaled with modified bandwidths, some of the LSPs might not get signaled successfully, leading to failover options.

Operational Constraints

LSP splitting has the following operational constraints:

- LSP bandwidth—Although there are a number of ways to allocate bandwidth values to the LSPs, the Junos OS implementation supports only an equal-bandwidth allocation policy when normalization is done, wherein all the member LSPs are signaled or re-signaled with equal bandwidth.
- Number of LSPs—If an ingress router is configured to have a minimum number of LSPs, it maintains the minimum number of LSPs even if the demand can be satisfied with less than the minimum number of LSPs. In case the ingress router is unable to do constraint-based routing for computations

on the sufficient number of LSPs or signal sufficient number of LSPs, the ingress router resorts to a number of failback options.

By default, an incremental approach is supported as a fallback option (unless configured differently), where an ingress router makes attempts to bring up the sufficient number of LSPs, such that the new aggregate bandwidth exceeds the old aggregate bandwidth (and is as close to the desired demand as possible). The ingress router then load-balances traffic using the LSPs. The LSPs that could not be brought up are removed by the ingress router.

Supported Criteria

When a container LSP signals a member LSP, the member LSP gets signaled with minimum-signaling-bandwidth. Since each member LSP is configured with autobandwidth, between two normalization events, each LSP can undergo autobandwidth adjustment multiple times. As the traffic demand increases, the ingress router creates additional supplementary LSPs . All member LSPs are used for ECMP, so they should roughly have the same reserved bandwidth after normalization.

For example, if there are K LSPs signaled after normalization, each LSP is signaled with equal bandwidth B. The total aggregate bandwidth reserved is B.K, where B satisfies the following condition:

- Minimum signaling-bandwidth is less than or equal to B, which is turn is less than or equal to the maximum signaling-bandwidth

  (minimum-signaling-bandwidth ≤ B ≤ maximum-signaling-bandwidth)

Until the next normalization event, each member LSP undergoes several autobandwidth adjustments. After any autobandwidth adjustment, if there are N LSPs with reserved bandwidths bi, where i=1,2,.., N, each bi should satisfy the following the condition:

- Minimum bandwidth is less than or equal to bi, which in turn is less than or equal to the maximum bandwidth

  (minimum-bandwidth ≤ bi ≤ maximum-bandwidth)

Both the above-mentioned conditions are applicable for per member LSP (nominal and supplementary), and essentially have the reserved bandwidth to exist within a range.

Splitting Triggers

Every time the normalization timer expires, the ingress router decides if LSP splitting is required. The ingress router works with the aggregate bandwidth instead of the individual LSP bandwidths. The following two variables are defined for aggregate bandwidth:

- `Current-Aggr-Bw`—Sum of reserved bandwidths of all current member LSPs.
- `New-Aggr-Bw`—Sum of traffic rates on all current member LSPs based on sampling.

Taking for example, if there are N member LSPs in the network at the time of normalization, the two approaches to trigger LSP splitting are as follows:

- Absolute trigger—LSP splitting is performed when `New-Aggr-Bw` is greater than `Aggregate-maximum-bandwidth`.

  (`New-Aggr-Bw > Aggregate-maximum-bandwidth`)

- Relative trigger—Under relative triggering, a dynamic calculation is performed. The `Current-Aggr-Bw` is compared with `New-Aggr-Bw` at the ingress routing device. LSP splitting is performed when the difference in bandwidth is greater than or equal to a calculated threshold amount. The following equation describes the desired state:

  ([1-a] x `Current-Aggr-Bw` < `New-Aggr-Bw` < [1+a] x `Current-Aggr-Bw`, where 0 </= a </= 1)

  > **NOTE**: In the above condition, "a" is the adjustment threshold and its default value is 10 percent (that is, 0.10). You can configure the adjustment threshold using the `splitting-merging-threshold` statement at the `[edit protocols mpls container-label-switched-path` *`lsp-name`*`]` hierarchy level. The value is also displayed in the `show mpls container-lsp extensive` command output.

  When `New-Aggr-Bw` is greater than `Current-Aggr-Bw` multiplied by [1+a], thus exceeding the calculated threshold, the ingress routing device does not perform normalization. Instead because this is a relative triggering situation, LSP splitting is performed. However, when both LSP splitting and LSP merging are configured on the ingress router, LSP splitting is triggered on the ingress router when one of the two conditions is satisfied.

LSP Merging

Operational Overview

Junos OS supports two kinds of LSPs – CLI-configured LSPs and dynamically created LSPs. The CLI-configured LSPs are created manually and remain in the system until the configuration is modified. The dynamic LSPs are created dynamically by next generation MVPN, BGP virtual private LAN service (VPLS), or LDP, based on a template configuration, and are removed from the system when not used by any application for a certain duration. LSP merging follows a similar approach as dynamic LSPs.

LSP merging enables an ingress routing device to dynamically eliminate some member LSPs of the container LSP so less state information is maintained in the network. If an ingress router provisions several member LSPs between the ingress and egress routers, and there is an overall reduction in aggregate bandwidth (resulting in some LSPs being under-utilized), the ingress router distributes the new traffic load among fewer LSPs.

Although there are a number of ways to merge the member LSPs, Junos OS supports only overall-merge when normalization is being performed. An ingress router considers the aggregate demand and the minimum (or maximum) number of LSPs and revises the number of LSPs that should be active at an ingress routing device. As a result, the following can take place periodically as the normalization timer fires:

- Re-signaling some of the existing LSPs with updated bandwidth
- Creating new LSPs
- Removing some of the existing LSPs

Operational Constraints

If a container LSP is not configured with autobandwidth, then the member LSPs are signaled with the static bandwidth value, if configured. LSP merging does not happen because there is no dynamic estimation of aggregate bandwidth. However, a manual trigger for splitting and adjusting with a specific bandwidth value can be configured.

NOTE:

- Nominal LSPs are never deleted as part of LSP merging.
- Before deleting an LSP, the LSP is made inactive, so that traffic shifts to other LSPs before removing the LSP. This is because RSVP sends PathTear before deleting routes and next hops from the Packet Forwarding Engine.
- When member LSPs are re-signaled with modified bandwidth, it might happen that some LSPs do not get signaled successfully.

Merging Triggers

Every time the normalization timer expires, the ingress router decides if LSP merging is required. The ingress router works with the aggregate bandwidth instead of the individual LSP bandwidths. The following two variables are defined for aggregate bandwidth:

- `Current-Aggr-Bw`—Sum of reserved bandwidths of all current member LSPs.
- `New-Aggr-Bw`—Sum of traffic rates on all current member LSPs based on sampling.

For example, if there are N member LSPs in the network at the time of normalization, the two approaches to trigger LSP merging are as follows:

- Absolute trigger—LSP merging is performed when `New-Aggr-Bw` is less than `Aggregate-minimum-bandwidth`.

  (`New-Aggr-Bw < Aggregate-minimum-bandwidth`)

- Relative trigger—The `Current-Aggr-Bw` is compared with `New-Aggr-Bw` at the ingress routing device. LSP merging is performed when the difference in the bandwidth amount is off by a threshold.

  ([1-a] x `Current-Aggr-Bw` < `New-Aggr-Bw` < [1+a] x `Current-Aggr-Bw`, where 0 </= a </= 1)

  > **NOTE**: In the above condition, "a" is the adjustment threshold and its default value is 10 percent (that is, 0.10). You can configure the adjustment threshold using the `splitting-merging-threshold` statement at the `[edit protocols mpls container-label-switched-path` *`lsp-name`*`]` hierarchy level. The value is also displayed in the `show mpls container-lsp extensive` command output.

  When the `New-Aggr-Bw` value is less than or equal to [1+a] multiplied by the `Current-Aggr-Bw` value, the ingress routing device does not perform normalization, but instead LSP merging is done. However, when both LSP splitting and LSP merging are configured on the ingress router, LSP splitting is triggered on the ingress router when one of the two conditions is satisfied.

Node and Link Protection

Junos OS supports the following mechanisms for node and link protection:

- Fast-reroute
- Link protection
- Node-link protection

Only one of the above-mentioned modes of protection can be configured on an ingress routing device at any given time. All member LSPs (nominal and supplementary) use the same mode of protection that is configured.

Naming Convention

While configuring a container LSP, a name is assigned to the LSP. The name of a nominal and a supplementary LSP is formed by adding the configured-name suffix and an auto-generated suffix to the name of the container LSP. The name of the container LSP is unique and is checked for accuracy during the configuration parsing. The container LSP name should uniquely identify parameters, such as the ingress and egress router names.

> **NOTE**: A container LSP member LSP and a point-to-point LSP on an ingress routing device cannot have the same LSP name.

The container LSPs follow a number-based LSP naming convention. For example, if the nominal LSP's configured name is bob and the number of member LSPs is N, the member LSPs are named bob-*<configured-suffix>*-1, bob-*<configured-suffix>*-2, ..., and bob-*<configured-suffix>*-N.

After a normalization event, the number of member LSPs can change. For example, if the number of member LSPs increases from six to eight, then the ingress routing device keeps the first six LSPs named bob-*<configured-suffix>*-1, bob-*<configured-suffix>*-2, ..., and bob-*<configured-suffix>*-6. The two additional LSPs are named bob-7 and bob-8. The original LSPs might need to be re-optimized if their signaled bandwidth changes.

Similarly, if the number of member LSPs reduces from eight to six, the ingress routing device re-signals the member LSPs in such as a way that the remaining active LSPs in the system are named bob-*<configured-suffix>*-1, bob-*<configured-suffix>*-2, ..., and bob-*<configured-suffix>*-6.

In the process of creating new LSPs, an RSVP LSP named bob-*<configured-suffix>*-7 can be configured.

Normalization

Operational Overview

Normalization is an event that happens periodically. When it happens, a decision is made on the number of member LSPs that should remain active and their respective bandwidths in a container LSP. More specifically, the decision is made on whether new supplementary LSPs are to be created, or any existing LSPs are required to be re-signaled or deleted during the normalization event.

Between two normalization events, a member LSP can undergo several autobandwidth adjustments. A normalization timer, similar to re-optimization timer, is configured. The normalization timer interval should be no less than the adjustment interval or optimization timer.

NOTE: Normalization is not triggered based on network events, such as topology changes.

Operational Constraints

Normalization has the following operational constraints:

- Normalization happens only when none of the member LSPs are undergoing re-optimization or make-before-break. Normalization starts when all the member LSPs complete their ongoing make-before-break. If normalization is pending, new optimization should not be attempted until the normalization is complete.
- After normalization, an ingress routing device first computes a set of bandwidth-feasible paths using constraint-based routing computations. If enough constraint-based routing computed paths are not

brought up with an aggregate bandwidth value that exceeds the desired bandwidth, several failover actions are taken.

- After a set of bandwidth-feasible paths are available, the ingress routing device signals those paths while keeping the original set of paths up with the old bandwidth values. The make-before-break is done with shared explicit (SE) sharing style, and when some of the LSPs do not get successfully re-signaled, a bounded number of retries is attempted for a specified duration. Only when all the LSPs are successfully signaled does the ingress router switch from the old instance of the container LSP to the newer instance. If all LSPs could not be successfully signaled, the ingress router keeps those instances of members that are up with higher bandwidth values.

  For example, if the bandwidth of an old instance of a member LSP (LSP-1) is 1G, the LSP is split into LSP-1 with bandwidth 2G and LSP-2 with bandwidth 2G. If the signaling of LSP-1 with bandwidth 2G fails, the ingress router keeps LSP-1 with bandwidth 1G and LSP-2 with bandwidth 2G.

  When there is a signaling failure, the ingress routing device stays in the error state, where some LSPs have updated bandwidth values only if the aggregate bandwidth has increased. The ingress router makes an attempt to bring up those LSPs that could not be successfully signaled, resulting in minimum traffic loss.

- If an LSP goes down between two normalization events, it can increase the load on other LSPs that are up. In order to prevent overuse of other LSPs, premature normalization can be configured in case of LSP failure. LSPs can go down because of pre-emption or lack of node or link protection. It might not be necessary to bring up the LSPs that are down because the normalization process re-runs the constraint-based routing path computations.

Inter-Operation with Autobandwidth

Taking as an example, there is one nominal LSP named LSP-1 configured with the following parameters:

- Splitting-bandwidth and maximum-signaling-bandwidth of 1G
- Merging-bandwidth and minimum-signaling-bandwidth of 0.8G
- Autobandwidth

Normalization is performed differently in the following scenarios:

Changes in Per-LSP Autobandwidth Adjustments

Table 19 on page 726 illustrates how normalization splits and merges member LSPs as autobandwidth adjustments change per-LSP bandwidth with unconditional normalization.

Table 19: Normalization with Per-LSP Autobandwidth Adjustment Changes

Normalization Time	Current State	Events	Adjusted State
T0	No state.	Initialization	LSP-1 is signaled with bandwidth of 0.8G
T1	LSP-1 usage increases to 1.5G	• Multiple autobandwidth adjustments since T0 is possible. • The ingress router decides to split LSP-1 into two LSPs, and creates LSP-2.	LSP-1 = 0.8G LSP-2 = 0.8G
T2	LSP-1 usage increase to 2G LSP-2 usage increases to 0.9G (within limits)	• Aggregate bandwidth is 2.9G, which exceeds aggregate splitting maximum of 2G. • The ingress router decides to split LSP-1 into three LSPs, and creates LSP-3.	LSP-1 = 1G LSP-2 = 1G LSP-3 = 1G
T3	LSP-3 usage increases to 1.5G	• Aggregate bandwidth is 3.5G with a maximum aggregate splitting of 3G. • The ingress router decides to split LSP-1 into four LSPs, and creates LSP-4.	LSP-1 = 1G LSP-2 = 1G LSP-3 = 1G LSP-4 = 1G
T4	LSP-2 usage drops to 0.5G	• Aggregate bandwidth is 3G. • The ingress router decides to merge LSP-1 and removes LSP-4.	LSP-1 = 1G LSP-2 = 1G LSP-3 = 1G

Because autobandwidth is configured on a per-LSP basis, every time there is an autobandwidth adjustment, the ingress router re-signals each LSP with `Max Avg Bw`.

Another approach to handling the changes in per-LSP autobandwidth adjustments is to not allow individual LSPs to run autobandwidth on the ingress router, but to run autobandwidth in passive (monitor) mode. This way, sampling is done at every statistics interval for member LSPs only, and

normalization is performed for the container LSP alone instead of acting on individual LSPs adjustment timer expiry.

As a result, the number of re-signaling attempts and bandwidth fluctuations for a given member LSP is reduced. Only the computed bandwidth-values per-member LSP is used by the ingress router to find an aggregate bandwidth to be used during normalization. Configuring autobandwidth adjustment followed by normalization (adjustments and normalization intervals are comparable) can lead to considerable overhead because of re-signaling.

Taking the same example, and applying the second approach, LSP-1 goes from 0.8G to 1.5G and then back to 0.8G. If the normalization timer is of the same order as the adjustment interval, the ingress router leaves LSP-1 alone with its original 0.8G and only signals LSP-2 with 0.8G. This helps achieve the final result of normalization, thus avoiding the extra signaling attempt on LSP-1 with 1.5G at adjustment timer expiry.

Because member LSPs always use equal bandwidth, any adjustment done on member LSPs is undone. The member LSPs are re-signaled with reduced bandwidth when compared to the reserved capacity in adjustment trigger with normalization trigger. Therefore, avoiding adjustment trigger for member LSPs might be useful assuming that normalization and adjustment intervals are of the same order.

NOTE: We recommend that the normalization timer be higher than the autobandwidth adjustment interval and regular optimization duration, as the traffic trends are observed at a longer time scale and normalization is performed one-to-three times per day. An LSP can undergo optimization for the following reasons:

- Normal optimization
- Autobandwidth adjustment
- Normalization

Changes in Traffic Growth

Table 20 on page 727 illustrates how normalization is performed when traffic grows in large factor.

Table 20: Normalization with Traffic Growth

Normalization Time	Current State	Events	Adjusted State
T0	No state		LSP-1 is signaled with bandwidth of 0.8G

Table 20: Normalization with Traffic Growth *(Continued)*

Normalization Time	Current State	Events	Adjusted State
T1	LSP-1 usage increase to 3G	• Aggregate usage exceeds maximum splitting bandwidth • The ingress router decides to split LSP-1, and creates two more supplementary LSPs	LSP-1 = 1G LSP-2 = 1G LSP-3 = 1G

Having fewer LSPs is preferred over signaling four LSPs each with 0.8G bandwidth, unless there is a constraint on the minimum number of LSPs.

Computed Range and Configured Feasible Ranges

When an ingress router is configured with the minimum and maximum number of LSPs, and per LSP splitting-bandwidth and merging-bandwidth values, the bandwidth thresholds are used for splitting and merging. For this, the number of LSPs (N) should satisfy the following constraints:

```
minimum-member-lsps ≤ N ≤ maximum-member-lsps
```

At the time of normalization, based on the aggregate demand X:

```
[X/splitting-bandwidth]  ≤ N ≤ [X/merging-bandwidth]
```

The above-mentioned constraints provide two ranges for N to work from. If the two ranges for N are overlapping, N will be selected from the overlapping interval (lowest possible N) to keep the number of LSPs small in the network.

Otherwise, if maximum-member-lsps is less than [X/splitting-bandwidth], the ingress router keeps (at maximum) the maximum-member-lsps in the system, and the bandwidth of each LSP is [X/maximum-member-lsps] or the maximum-signaling-bandwidth, whichever is less. It is possible that some LSPs might not get signaled successfully.

Similarly, if minimum-member-lsps is greater than [X/merging-bandwidth], the ingress router keeps (at minimum) the minimum-member-lsps in the system, and the bandwidth of each LSP is [X/minimum-member-lsps] or the minimum-signaling-bandwidth, whichever is less.

Taking as an example, normalization is performed as following in these cases:

- Case 1

- minimum-member-lsps = 2
- maximum-member-lsps = 10
- aggregate demand = 10G
- merging-bandwidth = 1G
- splitting-bandwidth = 2.5G

In this case, the ingress routing device signals four member LSPs each with a bandwidth of 2G.

- Case 2
  - minimum-member-lsps = 5
  - maximum-member-lsps = 10
  - aggregate demand = 10G
  - merging-bandwidth = 2.5G
  - splitting-bandwidth = 10G

  In this case, the ingress routing device signals five member LSPs each with a bandwidth of 2G. Here, the static configuration on the number of member LSPs takes precedence.

- Case 3
  - minimum-signaling-bandwidth = 5G
  - maximum-signaling-bandwidth = 40G
  - merging-bandwidth = 10G
  - splitting-bandwidth = 50G

  When a container LSP comes up, the nominal LSP is signaled with minimum-signaling-bandwidth. At the time of normalization, the new-aggregate-bandwidth is 100G. To find N and the bandwidth of each LSP, N should satisfy the following constraint:

  ```
  100/50 ≤ N ≤ 100/10, which gives 2 ≤ N ≤ 10
  ```

  Therefore, N is equal to:

  - N = 2, bandwidth = min {100/2G, 40G} = 40G

    This option does not satisfy the new aggregate of 100G.

- N = 3, bandwidth = min {100/3G, 40G} = 33.3G

  This option makes the aggregate bandwidth equal to 100G.

In this case, the ingress routing device signals three LSPs each with a bandwidth of 33.3G.

NOTE: The ingress router does not signal an LSP smaller than the minimum-signaling-bandwidth.

Constraint-Based Routing Path Computation

Although there are no changes in the general constraint-based routing path computation, with a container LSP, there is a separate module that oversees the normalization process, schedules constraint-based routing events, and schedules switchover from an old instance to a new instance, when appropriate. An ingress routing device has to handle the constraint-based routing path computation periodically. When normalization occurs, an ingress router has to compute constraint-based routing paths, if the number of LSPs or the bandwidth of the LSPs needs to be changed.

For example, there are K LSPs at the ingress router with bandwidth values X-1, X-2, ..., and X-K. The current aggregate bandwidth value is Y, which is the sum of X-1 plus X-2 plus X-K. If there is a new demand of W, the ingress router first computes how many LSPs are required. If the ingress router only needs N LSPs (LSP-1, LSP-2, .., and LSP-N) each with bandwidth value B, the task of the constraint-based routing module is to provide a set of bandwidth-feasible LSPs that can accommodate the new aggregate demand which is not less than Y.

The ingress router then tries to see if the constraint-based routing paths can be computed successfully for all N LSPs. If the paths for all the LSPs are found successfully, the constraint-based routing module returns the set to the normalization module.

It is possible that the constraint-based routing computation is not successful for some LSPs. In this case, the ingress routing device takes the following action:

- If the configuration allows for incremental-normalization, implying if the ingress router has enough LSPs whose aggregate exceeds Y, the constraint-based routing module returns that set of paths.
- Whether increment-normalization is configured or not, if constraint-based routing paths could not be computed for a sufficient number of LSPs, the ingress router has to repeat the process of finding a new set of LSPs. Initially, the ingress router starts with the lowest value of N from the feasible region. Every time, the ingress router has to revise the number, it linearly increases it by 1. As a result, per LSP bandwidth becomes less and therefore, there is a greater chance of successful signaling. The process is repeated for all feasible values of N (or some bounded number of times or duration as configured).

The ingress router signals the LSPs after successful computations of the constraint-based routing path computation. It might happen that when the LSPs are signaled, signaling of many LSPs fail. In addition to the constraint-based routing path computations to be successful, the RSVP signaling should also succeed, such that the new aggregate is not less than the old aggregate bandwidth.

Sampling

Sampling is important for normalization to function. With sampling configured, an ingress routing device is able to make a statistical estimate of the aggregate traffic demands. Every time the sampling timer fires, the ingress routing device can consider traffic rates on different LSPs and compute an aggregate bandwidth sample. This sampling timer is different from the statistics sampling done periodically by RSVP on all LSPs. The aggregate bandwidth is a sample to be used at the time of normalization. An ingress routing device can save past samples to compute an average (or some other statistical measure) and use it the next time normalization happens.

To remove any outlier samples, a sampling token is configured. In other words, from all the aggregate samples collected during the configured time, the bottom and top outliers are ignored before computing a statistical measure from the remaining samples.

The following two methods of computing an aggregate bandwidth value are supported:

- Average—All the aggregate bandwidth samples are considered by the ingress routing device, and then all the outlier samples are removed. The average bandwidth value is computed from the remaining samples to be used during normalization.
- Max—All the aggregate bandwidth samples are considered by the ingress routing device, and then all the outlier samples are removed. The maximum bandwidth value is picked from the remaining samples to be used during normalization.

The time duration, the number of past aggregate samples to store, the percentile value to determine, and the ignore outliers are user-configurable parameters.

Support for NSR, IPG-FA, and Static Routes

Starting with Junos OS Release 15.1, container label-switched paths (LSPs) provide support for nonstop active routing (NSR), IGP forwarding adjacency (FA), and static routes to address the requirements of wider business cases.

NSR Support

A container LSP has the characteristics of ECMP and RSVP traffic engineering. Because a container LSP consists of several member LSPs between an ingress and an egress router, with each member LSP taking a different path to the same destination, the ingress router is configured with all the parameters necessary to compute an RSVP ECMP LSP. These parameters along with the forwarding state

information have to be synchronized between the primary and backup Routing Engines to enable the support for nonstop active routing (NSR) for container LSPs. While some of the forwarding state information on the backup Routing Engine is locally built based on the configuration, most of it is built based on periodic updates from the primary Routing Engine. The container LSPs are created dynamically using the replicated states on the backup Routing Engine.

By default, normalization occurs once in every 6 hours and during this time, a number of autobandwidth adjustments happen over each member LSP. A member LSP is resized according to the traffic it carries and the configured autobandwidth configuration parameters. The aggregate demand on a container LSP is tracked by summing up the bandwidth across all the member LSPs.

For RSVP point-to-point LSPs, a Routing Engine switchover can be under any one of the following:

- **Steady state**

  In the steady state, the LSP state is up and forwards traffic; however, no other event, such as the make-before-break (MBB), occurs on the LSP. At this stage, the RPD runs on both the Routing Engines, and the switchover event toggles between the primary and backup Routing Engine. The backup Routing Engine has the LSP information replicated already. After the switchover, the new primary uses the information of the replicated structure to construct the container LSP and en-queues the path (ERO) of LSP in the retrace mode. RSVP signals and checks if the path mentioned in the ERO is reachable. If the RSVP checks fail, then the LSP is restarted. If the RSVP checks succeed, the LSP state remains up.

- **Action leading to make-before-break (MBB)**

  A container LSP can be optimized with updated bandwidth, and this change is done in a MBB fashion. During an MBB process, there are two path instances for a given LSP, and the LSP switches from one instance to another. For every Routing Engine switchover, the path is checked to find out where in the MBB process the path is. If the path is in the middle of the MBB process, with the main instance being down and the re-optimized path being up, then MBB can switch over to the new instance. The `show mpls lsp extensive` command output, in this case, is as follows:

```
13 Dec  3 01:33:38.941 Make-before-break: Switched to new instance
   12 Dec  3 01:33:37.943 Record Route:  10.1.1.1
   11 Dec  3 01:33:37.942 Up
   10 Dec  3 01:33:37.942 Automatic Autobw adjustment succeeded: BW changes from 100 bps to
281669 bps
    9 Dec  3 01:33:37.932 Originate make-before-break call
    8 Dec  3 01:33:37.931 CSPF: computation result accepted  10.1.1.1
    7 Dec  3 01:28:44.228 CSPF: ERO retrace was successful  10.1.1.1
    6 Dec  3 01:19:39.931 10.1.1.2 Down: mbb/reopt
    5 Dec  3 01:18:29.286 Up: mbb/reopt
    4 Dec  3 01:14:47.119 10.1.1.2 Down: mbb/reopt
```

```
    3 Dec  3 01:13:29.285 Up: mbb/reopt
    2 Dec  3 01:10:59.755 Selected as active path: selected by master RE
```

A similar behavior is retained for member LSPs during bandwidth optimization.

A Routing Engine switchover under the steady state (when normalization is not in progress), keeps the container LSPs up and running without any traffic loss. Events, such as an MBB due to autobandwidth adjustments, link status being down, or double failure, in the steady state are similar to a normal RSVP point-to-point LSP.

If the container LSP is in the process of normalization, and the normalization event is triggered either manually or periodically, it goes through the computation and execution phase. In either of the cases, zero percent traffic loss is not guaranteed.

- Normalization in the computation phase

  During the computation phase, the primary Routing Engine calculates the targeted member LSP count and bandwidth with which each member LSP should be re-signaled. The backup Routing Engine has limited information about the container LSP, such as the LSP name, LSP ID, current bandwidth of its member LSP, member LSP count, and the normalization retry count. If the switchover happens during the computation phase, then the backup Routing Engine is not aware of the targeted member LSP count and the bandwidth to be signaled. Since traffic statistics are not copied to the backup Routing Engine, it cannot compute the targeted member count and bandwidth. In this case, the new primary Routing Engine uses the old data stored in the targeted member LSP count and the targeted bandwidth to signal the LSPs.

- Normalization in the execution phase

  During the execution phase, RSVP of the primary Routing Engine tries to signal the LSPs with the newly calculated bandwidth. If the switchover occurs during the signaling of LSPs with greater bandwidth or during LSP splitting or merging, then the new primary Routing Engine uses the information of the targeted member count and bandwidth value to be signaled with, to bring up the LSPs.

IPG-FA Support

A forwarding adjacency (FA) is a traffic engineering label-switched path (LSP) that is configured between two nodes and used by an interior gateway protocol (IGP) to forward traffic. By default, an IGP does not consider MPLS traffic-engineering tunnels between sites, for traffic forwarding. Forwarding adjacency treats a traffic engineering LSP tunnel as a link in an IGP topology, thus allowing the nodes in the network also to forward the IP traffic to reach the destination over this FA LSP. A forwarding adjacency can be created between routing devices regardless of their location in the network.

To advertise a container LSP as an IGP-FA, the LSP name needs to be configured either under IS-IS or OSPF. For example:

IS-IS

```
[edit]
protocols {
    isis {
        label-switched-path container-lsp-name;
    }
}
```

OSPF

```
[edit]
protocols {
    ospf {
        area 0.0.0.0 {
            label-switched-path container-lsp-name;
        }
    }
}
```

NOTE: The IGP-FA is applied to both container LSPs and regular point-to-point LSPs. If a container LSP and a point-to-point LSP share the same name, the point-to-point LSP is given preference for FA.

Static Route Support

Static routes often include only one or very few paths to a destination and generally do not change. These routes are used for stitching services when policies and other protocols are not configured.

To advertise a container LSP as a static route, the LSP name needs to be configured under the static route configuration. For example:

Static Route

```
[edit]
routing-options {
    static {
        route destination {
            lsp-next-hop container-lsp-name;
        }
```

```
    }
}
```

NOTE: The static route support is applied to both container LSPs and regular point-to-point LSPs. If a container LSP and a point-to-point LSP share the same name, the point-to-point LSP is given preference for static routing.

Configuration Statements Supported for Container LSPs

Table 21 on page 735 lists the MPLS LSP configuration statements that apply to RSVP LSP and a container LSP (nominal and supplementary).

The configuration support is defined using the following terms:

- Yes—The configuration statement is supported for this type of LSP.
- No—The configuration statement is not supported for this type of LSP.
- N/A—The configuration statement is not applicable for this type of LSP.

Table 21: Applicability of RSVP LSPs Configuration to a Container LSP

Configuration Statement	RSVP LSP (Ingress)	Member LSP (Ingress)
adaptive (Default: non-adaptive)	Yes	Yes
admin-down	Yes	Yes
admin-group	Yes	Yes
admin-groups-except	Yes	Yes
apply-groups	Yes	Yes
apply-groups-except	Yes	Yes

Table 21: Applicability of RSVP LSPs Configuration to a Container LSP *(Continued)*

Configuration Statement	RSVP LSP (Ingress)	Member LSP (Ingress)
associate-backup-pe-groups	Yes	No
associate-lsp (No bidirectional support)	Yes	No
auto-bandwidth	Yes	Yes
backup	Yes	No
bandwidth	Yes	Yes
class-of-service	Yes	Yes
corouted-bidirectional (No bidirectional support)	Yes	No
corouted-bidirectional-passive (No bidirectional support)	Yes	No
description	Yes	Yes
disable	Yes	Yes
egress-protection	Yes	No
exclude-srlg	Yes	Yes
fast-reroute (Same fast reroute for all member LSPs)	Yes	Yes

Table 21: Applicability of RSVP LSPs Configuration to a Container LSP *(Continued)*

Configuration Statement	RSVP LSP (Ingress)	Member LSP (Ingress)
from	Yes	Yes
hop-limit	Yes	Yes
install	Yes	Yes
inter-domain (Same termination router)	Yes	Yes
secondary (All LSPs are primary)	Yes	No
ldp-tunneling (All LSPs do tunneling)	Yes	Yes
least-fill	Yes	Yes
link-protection (All LSPs share same link protection mechansim)	Yes	Yes
lsp-attributes	Yes	Yes
lsp-external-controller	Yes	No
metric (All LSPs are same)	Yes	Yes
most-fill	Yes	Yes

Table 21: Applicability of RSVP LSPs Configuration to a Container LSP *(Continued)*

Configuration Statement	RSVP LSP (Ingress)	Member LSP (Ingress)
no-cspf (LSPs use IGP)	Yes	Yes
no-decrement-ttl (All LSPs share same TTL behavior)	Yes	Yes
no-install-to-address	Yes	Yes
no-record	Yes	Yes
node-link-protection (Al LSPs share same node-link protection mechanism)	Yes	Yes
oam	Yes	Yes
optimize-hold-dead-delay (All LSPs have same value)	Yes	Yes
optimize-switchover-delay (All LSPs have same value)	Yes	Yes
optimize-timer (All LSPs have same value)	Yes	Yes
p2mp	Yes	N/A
policing (Variable traffic)	Yes	No

Table 21: Applicability of RSVP LSPs Configuration to a Container LSP *(Continued)*

Configuration Statement	RSVP LSP (Ingress)	Member LSP (Ingress)
preference	Yes	Yes
primary (All paths are primary)	Yes	No
random	Yes	Yes
record	Yes	Yes
retry-limit (Applicable to members)	Yes	Yes
retry-timer (Applicable to members)	Yes	Yes
revert-timer (No secondary LSP)	Yes	No
secondary (All LSPs are primary)	Yes	No
soft-preemption	Yes	Yes
standby (All LSPs are standby)	Yes	No
template	Yes	No

Table 21: Applicability of RSVP LSPs Configuration to a Container LSP *(Continued)*

Configuration Statement	RSVP LSP (Ingress)	Member LSP (Ingress)
to	Yes	Yes
traceoptions	Yes	Yes
ultimate-hop-popping	Yes	Yes

Impact of Configuring Container LSPs on Network Performance

A container LSP is a container LSP that allows multiple member LSPs to co-exist and be managed as a bundle. The member LSPs are similar to independent point-to-point RSVP LSPs. As a result, resource consumption is similar to the sum of resources consumed by each point-to-point RSVP LSP. However, provisioning a container LSP is more efficient, as under-utilized member LSPs are dynamically removed, thus saving memory and CPU resources.

The container LSP features are dependent on the presence of a functional base MPLS RSVP implementation. As a result, a container LSP does not introduce any security considerations beyond the existing considerations for the base MPLS RSVP functionality. The categories of possible attacks and countermeasures are as follows:

- Interaction with processes and router configuration

  No new communication mechanisms with external hosts are required for a container LSP. Data arrives at the RSVP module through local software processes and router configuration, other than RSVP neighbor adjacency. Junos OS provides security controls on access to the router and router configuration.

- Communication with external RSVP neighbors

  RSVP signaled MPLS LSPs depend on the services of RSVP and IGP to communicate RSVP messages among neighboring routers across the network . Because the RSVP sessions involve communication outside of the local router, they are subject to many forms of attack, such as spoofing of peers, injection of falsified RSVP messages and route updates, and attacks on the underlying TCP/UDP transport for sessions. Junos OS provides countermeasures for such attack vectors.

- Resource limits and denial of service

  Junos OS provides several mechanisms through policers and filters to protect against denial-of-service attacks based on injecting higher than the expected traffic demands. At the MPLS LSP level,

Junos OS allows operators to configure limits on the LSP bandwidth and the number of LSPs. However, like point-to-point RSVP LSPs, container LSPs do not enforce limits on the volume of traffic forwarded over these LSPs.

Supported and Unsupported Features

Junos OS supports the following container LSP features:

- Equal-bandwidth-based LSP splitting mechanism
- Aggregate-bandwidth-based LSP splitting and merging in a make-before-break way
- LSP-number-based naming mechanism for dynamically created member LSPs
- Periodic sampling mechanisms to estimate aggregate bandwidth
- Interoperability with auto-bandwidth feature
- ECMP using the dynamically created LSPs
- LDP-tunneling on the dynamically created LSP
- Configuring container LSP using IGP shortcuts
- Aggregated Ethernet links
- Logical systems

Junos OS does **not** support the following container LSP functionality:

- Node and link disjoint paths for different LSPs between an ingress and an egress routing device
- Bandwidth allocation policy different from equal bandwidth policy at the normalization event
- Constraint-based routing path computation to find equal IGP cost paths for different LSPs
- RSVP objects, such as `MLSP_TUNNEL Sender Template`, and `MLSP_TUNNEL Filter Specification` defined in [KOMPELLA-MLSP]
- Change in topology as a trigger for LSP splitting and merging
- Change in topology and link failure as a trigger for normalization, unless member LSPs go down
- Egress protection on container LSP
- Container LSP as a backup LSP for IGP interface
- Container LSP as provider tunnel for multicast VPNs
- Dynamic LSPs for normalization

- CCC using container LSP
- Secondary paths for container LSP
- Bidirectional container LSP
- Policing
- Static routes using container LSPs as next hops on a best-effort basis
- External path computing entity, such as PCE
- Multichassis
- IPv6

Example: Configuring Dynamic Bandwidth Management Using Container LSP

IN THIS SECTION

- Requirements | 742
- Overview | 743
- Configuration | 744
- Verification | 756

This example shows how to enable dynamic bandwidth management by configuring container label-switched paths (LSPs) that enable load balancing across multiple point-to-point member LSPs.

Requirements

This example uses the following hardware and software components:

- Five routers that can be a combination of M Series, MX Series, or T Series routers, out of which two routers are provider edge (PE) routers and three routers are provider (P) routers
- Junos OS Release 14.2 or later running on all the routers

Before you begin:

1. Configure the device interfaces.
2. Configure the autonomous system numbers and router IDs for the devices.

3. Configure the following protocols:

   - RSVP
   - MPLS
   - BGP
   - OSPF

Overview

IN THIS SECTION

- Topology | **743**

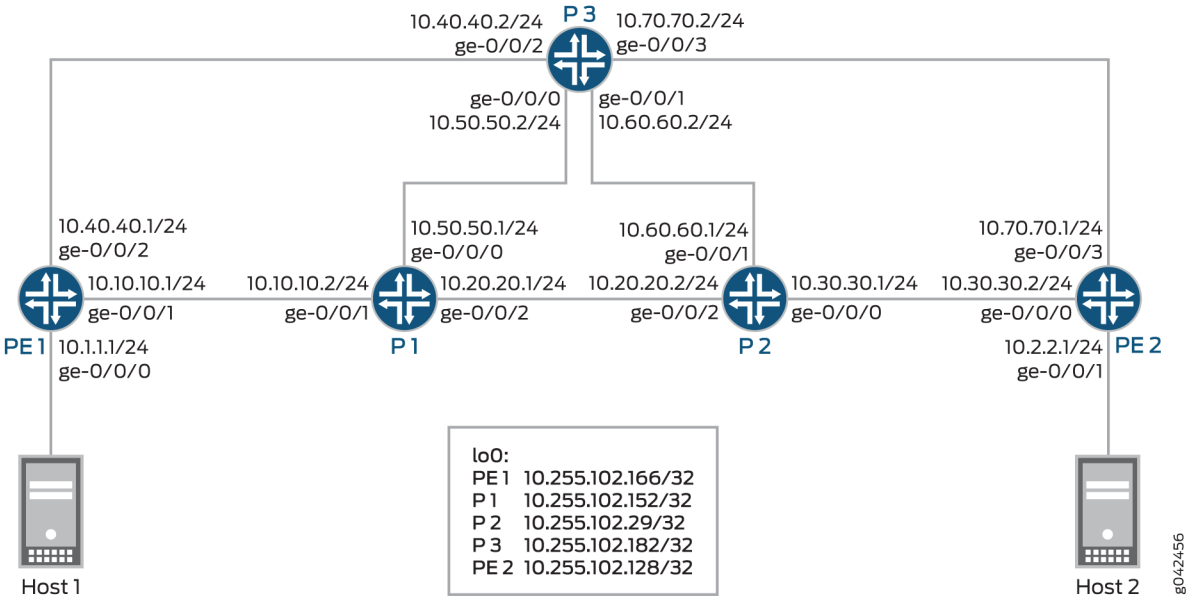
Starting with Junos OS Release 14.2, a new type of LSP, called a container LSP, is introduced to enable load balancing across multiple point-to-point LSPs. A container LSP includes one or more member LSPs between the same ingress and egress routing devices. The member LSPs are similar to independent point-to-point LSPs, and each member LSP takes a different path to the same destination and can be routed along a different IGP cost path.

A container LSP provides support for dynamic bandwidth management by enabling the ingress router to dynamically add and remove member LSPs through a process called LSP splitting and LSP merging, respectively, based on configuration and aggregate traffic. Besides addition and deletion, member LSPs can also be re-optimized with different bandwidth values in a make-before-break way.

Topology

Figure 49 on page 744 is a sample topology configured with container LSPs.

Figure 49: Dynamic Bandwidth Management Using Container LSP



In this example, Routers PE1 and PE2 are the PE routers connected to hosts Host1 and Host2, respectively. The core routers, Routers P1, and P2, and P3 connect to the PE routers.

Configuration

IN THIS SECTION

- CLI Quick Configuration | 744
- Procedure | 749

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

PE1

```
set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
set interfaces ge-0/0/1 unit 0 family inet address 10.10.10.1/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 10.40.40.1/24
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.102.166/32
set interfaces lo0 unit 0 family mpls
set routing-options router-id 10.255.102.166
set routing-options autonomous-system 65034
set routing-options forwarding-table export pplb
set protocols rsvp preemption aggressive
set protocols rsvp interface all aggregate
set protocols rsvp interface fxp0.0 disable
set protocols rsvp interface ge-0/0/1.0
set protocols rsvp interface ge-0/0/2.0
set protocols mpls statistics file auto-bw
set protocols mpls statistics file size 10m
set protocols mpls statistics interval 10
set protocols mpls statistics auto-bandwidth
set protocols mpls label-switched-path PE1-to-PE2-template1 template
set protocols mpls label-switched-path PE1-to-PE2-template1 optimize-timer 30
set protocols mpls label-switched-path PE1-to-PE2-template1 link-protection
set protocols mpls label-switched-path PE1-to-PE2-template1 adaptive
set protocols mpls label-switched-path PE1-to-PE2-template1 auto-bandwidth adjust-interval 300
set protocols mpls label-switched-path PE1-to-PE2-template1 auto-bandwidth adjust-threshold 5
set protocols mpls label-switched-path PE1-to-PE2-template1 auto-bandwidth minimum-bandwidth 10m
set protocols mpls label-switched-path PE1-to-PE2-template1 auto-bandwidth maximum-bandwidth 10m
set protocols mpls container-label-switched-path PE1-PE2-container-100 label-switched-path-
template PE1-to-PE2-template1
set protocols mpls container-label-switched-path PE1-PE2-container-100 to 10.255.102.128
set protocols mpls container-label-switched-path PE1-PE2-container-100 splitting-merging maximum-
member-lsps 20
set protocols mpls container-label-switched-path PE1-PE2-container-100 splitting-merging minimum-
member-lsps 2
set protocols mpls container-label-switched-path PE1-PE2-container-100 splitting-merging
splitting-bandwidth 40m
set protocols mpls container-label-switched-path PE1-PE2-container-100 splitting-merging merging-
bandwidth 6m
set protocols mpls container-label-switched-path PE1-PE2-container-100 splitting-merging maximum-
signaling-bandwidth 10m
```

```
set protocols mpls container-label-switched-path PE1-PE2-container-100 splitting-merging minimum-
signaling-bandwidth 10m
set protocols mpls container-label-switched-path PE1-PE2-container-100 splitting-merging
normalization normalize-interval 400
set protocols mpls container-label-switched-path PE1-PE2-container-100 splitting-merging
normalization failover-normalization
set protocols mpls container-label-switched-path PE1-PE2-container-100 splitting-merging
normalization normalization-retry-duration 20
set protocols mpls container-label-switched-path PE1-PE2-container-100 splitting-merging
normalization normalization-retry-limits 3
set protocols mpls container-label-switched-path PE1-PE2-container-100 splitting-merging
sampling cut-off-threshold 1
set protocols mpls container-label-switched-path PE1-PE2-container-100 splitting-merging
sampling use-percentile 90
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp group to-PE2 type internal
set protocols bgp group to-PE2 local-address 10.255.102.166
set protocols bgp group to-PE2 family inet-vpn unicast
set protocols bgp group to-PE2 export direct
set protocols bgp group to-PE2 neighbor 10.255.102.128
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0 metric 100
set policy-options policy-statement direct term 1 from protocol direct
set policy-options policy-statement direct term 1 then accept
set policy-options policy-statement pplb then load-balance per-packet
set routing-instances vpn1 instance-type vrf
set routing-instances vpn1 interface ge-0/0/0.0
set routing-instances vpn1 route-distinguisher 10.255.102.166:1
set routing-instances vpn1 vrf-target target:1:1
set routing-instances vpn1 vrf-table-label
```

P1

```
set interfaces ge-0/0/0 unit 0 family inet address 10.50.50.1/24
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 10.10.10.2/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 10.20.20.1/24
set interfaces ge-0/0/2 unit 0 family mpls
```

```
set interfaces lo0 unit 0 family inet address 10.255.102.152/32
set protocols rsvp interface all aggregate
set protocols rsvp interface fxp0.0 disable
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0 metric 100
```

P2

```
set interfaces ge-0/0/0 unit 0 family inet address 10.30.30.1/24
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 10.60.60.1/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 10.20.20.2/24
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.102.29/32
set protocols rsvp interface all aggregate
set protocols rsvp interface fxp0.0 disable
set protocols mpls statistics file auto_bw
set protocols mpls statistics file size 10m
set protocols mpls statistics interval 5
set protocols mpls statistics auto-bandwidth
set protocols mpls icmp-tunneling
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0 metric 100
```

P3

```
set interfaces ge-0/0/0 unit 0 family inet address 10.50.50.2/24
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 10.60.60.2/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 10.40.40.2/24
set interfaces ge-0/0/2 unit 0 family mpls
```

```
set interfaces ge-0/0/3 unit 0 family inet address 10.70.70.2/24
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.102.182/32
set protocols rsvp interface all aggregate
set protocols rsvp interface fxp0.0 disable
set protocols mpls icmp-tunneling
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
```

PE2

```
set interfaces ge-0/0/0 unit 0 family inet address 10.30.30.2/24
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 10.2.2.1/24
set interfaces ge-0/0/3 unit 0 family inet address 10.70.70.1/24
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.102.128/32
set interfaces lo0 unit 0 family mpls
set routing-options router-id 10.255.102.128
set routing-options autonomous-system 65034
set protocols rsvp interface all aggregate
set protocols rsvp interface fxp0.0 disable
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp group to-PE1 type internal
set protocols bgp group to-PE1 local-address 10.255.102.128
set protocols bgp group to-PE1 family inet-vpn unicast
set protocols bgp group to-PE1 neighbor 10.255.102.166
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set policy-options policy-statement direct from protocol direct
set policy-options policy-statement direct then accept
set routing-instances vpn1 instance-type vrf
set routing-instances vpn1 interface ge-0/0/1.0
set routing-instances vpn1 route-distinguisher 10.255.102.128:1
set routing-instances vpn1 vrf-target target:1:1
set routing-instances vpn1 vrf-table-label
```

Procedure

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure Router PE1:

1. Configure the Router PE1 interfaces.

```
[edit interfaces]
user@PE1# set ge-0/0/0 unit 0 family inet address 10.1.1.1/24
user@PE1# set ge-0/0/1 unit 0 family inet address 10.10.10.1/24
user@PE1# set ge-0/0/1 unit 0 family mpls
user@PE1# set ge-0/0/2 unit 0 family inet address 10.40.40.1/24
user@PE1# set ge-0/0/2 unit 0 family mpls
user@PE1# set lo0 unit 0 family inet address 10.255.102.166/32
user@PE1# set lo0 unit 0 family mpls
```

2. Configure the router ID and autonomous system number for Router PE1.

```
[edit routing-options]
user@PE1# set router-id 10.255.102.166
user@PE1# set autonomous-system 65034
```

3. Enable the policy to load-balance traffic.

```
[edit routing-options]
user@PE1# set forwarding-table export pplb
```

4. Enable RSVP on all Router PE1 interfaces (excluding the management interface).

```
[edit protocols]
user@PE1# set rsvp preemption aggressive
user@PE1# set rsvp interface all aggregate
user@PE1# set rsvp interface fxp0.0 disable
```

```
user@PE1# set rsvp interface ge-0/0/1.0
user@PE1# set rsvp interface ge-0/0/2.0
```

5. Enable MPLS on all the interfaces of Router PE1 (excluding the management interface).

```
[edit protocols]
user@PE1# set mpls interface all
user@PE1# set mpls interface fxp0.0 disable
```

6. Configure the MPLS statistics parameters.

```
[edit protocols]
user@PE1# set mpls statistics file auto-bw
user@PE1# set mpls statistics file size 10m
user@PE1# set mpls statistics interval 10
user@PE1# set mpls statistics auto-bandwidth
```

7. Configure the label-switched path (LSP) template parameters.

```
[edit protocols]
user@PE1# set mpls label-switched-path PE1-to-PE2-template1 template
user@PE1# set mpls label-switched-path PE1-to-PE2-template1 optimize-timer 30
user@PE1# set mpls label-switched-path PE1-to-PE2-template1 link-protection
user@PE1# set mpls label-switched-path PE1-to-PE2-template1 adaptive
user@PE1# set mpls label-switched-path PE1-to-PE2-template1 auto-bandwidth adjust-interval
300
user@PE1# set mpls label-switched-path PE1-to-PE2-template1 auto-bandwidth adjust-threshold
5
user@PE1# set mpls label-switched-path PE1-to-PE2-template1 auto-bandwidth minimum-
bandwidth 10m
user@PE1# set mpls label-switched-path PE1-to-PE2-template1 auto-bandwidth maximum-
bandwidth 10m
```

8. Configure a container LSP between Router PE1 and Router PE2, and assign the PE1-to-PE2-template1 LSP template.

```
[edit protocols]
user@PE1# set mpls container-label-switched-path PE1-PE2-container-100 to 10.255.102.128
```

```
user@PE1# set mpls container-label-switched-path PE1-PE2-container-100 label-switched-path-
template PE1-to-PE2-template1
```

9. Configure the container LSP parameters.

```
[edit protocols]
user@PE1# set mpls container-label-switched-path PE1-PE2-container-100 splitting-merging
maximum-member-lsps 20
user@PE1# set mpls container-label-switched-path PE1-PE2-container-100 splitting-merging
minimum-member-lsps 2
user@PE1# set mpls container-label-switched-path PE1-PE2-container-100 splitting-merging
splitting-bandwidth 40m
user@PE1# set mpls container-label-switched-path PE1-PE2-container-100 splitting-merging
merging-bandwidth 6m
user@PE1# set mpls container-label-switched-path PE1-PE2-container-100 splitting-merging
maximum-signaling-bandwidth 10m
user@PE1# set mpls container-label-switched-path PE1-PE2-container-100 splitting-merging
minimum-signaling-bandwidth 10m
user@PE1# set mpls container-label-switched-path PE1-PE2-container-100 splitting-merging
normalization normalize-interval 400
user@PE1# set mpls container-label-switched-path PE1-PE2-container-100 splitting-merging
normalization failover-normalization
user@PE1# set mpls container-label-switched-path PE1-PE2-container-100 splitting-merging
normalization normalization-retry-duration 20
user@PE1# set mpls container-label-switched-path PE1-PE2-container-100 splitting-merging
normalization normalization-retry-limits 3
user@PE1# set mpls container-label-switched-path PE1-PE2-container-100 splitting-merging
sampling cut-off-threshold 1
user@PE1# set mpls container-label-switched-path PE1-PE2-container-100 splitting-merging
sampling use-percentile 90
```

10. Configure the BGP group, and assign the local and neighbor IP addresses.

```
[edit protocols]
user@PE1# set bgp group to-PE2 type internal
user@PE1# set bgp group to-PE2 local-address 10.255.102.166
user@PE1# set bgp group to-PE2 neighbor 10.255.102.128
user@PE1# set bgp group to-PE2 family inet-vpn unicast
user@PE1# set bgp group to-PE2 export direct
```

11. Enable OSPF on all the interfaces of Router PE1 (excluding the management interface) along with traffic engineering capabilities.

```
[edit protocols]
user@PE1# set ospf traffic-engineering
user@PE1# set ospf area 0.0.0.0 interface all
user@PE1# set ospf area 0.0.0.0 interface fxp0.0 disable
user@PE1# set ospf area 0.0.0.0 interface ge-0/0/2.0 metric 100
```

12. Configure the policy statement to load-balance traffic.

```
[edit policy-options]
user@PE1# set policy-statement direct term 1 from protocol direct
user@PE1# set policy-statement direct term 1 then accept
user@PE1# set policy-statement pplb then load-balance per-packet
```

13. Configure a routing instance on Router PE1, and assign the routing instance interface.

```
[edit routing-instances]
user@PE1# set vpn1 instance-type vrf
user@PE1# set vpn1 interface ge-0/0/0.0
```

14. Configure the route distinguisher, vrf target, and vrf-table label values for the VRF routing instance.

```
[edit routing-instances]
user@PE1# set vpn1 route-distinguisher 10.255.102.166:1
user@PE1# set vpn1 vrf-target target:1:1
user@PE1# set vpn1 vrf-table-label
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, `show protocols`, `show policy-options`, and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@PE1# show interfaces
ge-0/0/0 {
```

```
    unit 0 {
        family inet {
            address 10.1.1.1/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.10.10.1/24;
        }
        family mpls;
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 10.40.40.1/24;
        }
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.255.102.166/32;
        }
        family mpls;
    }
}
```

```
user@PE1# show routing-options
router-id 10.255.102.166;
autonomous-system 65034;
forwarding-table {
    export pplb;
}
```

```
user@PE1# show protocols
rsvp {
```

```
    preemption aggressive;
    interface all {
        aggregate;
    }
    interface fxp0.0 {
        disable;
    }
    interface ge-0/0/1.0;
    interface ge-0/0/2.0;
}
mpls {
    statistics {
        file auto-bw size 10m;
        interval 10;
        auto-bandwidth;
    }
    label-switched-path PE1-to-PE2-template1 {
        template;
        optimize-timer 30;
        link-protection;
        adaptive;
        auto-bandwidth {
            adjust-interval 300;
            adjust-threshold 5;
            minimum-bandwidth 10m;
            maximum-bandwidth 10m;
        }
    }
    container-label-switched-path PE1-PE2-container-100 {
        label-switched-path-template {
            PE1-to-PE2-template1;
        }
        to 10.255.102.128;
        splitting-merging {
            maximum-member-lsps 20;
            minimum-member-lsps 2;
            splitting-bandwidth 40m;
            merging-bandwidth 6m;
            maximum-signaling-bandwidth 10m;
            minimum-signaling-bandwidth 10m;
            normalization {
                normalize-interval 400;
                failover-normalization;
```

```
                normalization-retry-duration 20;
                normalization-retry-limits 3;
            }
            sampling {
                cut-off-threshold 1;
                use-percentile 90;
            }
        }
    }
    interface all;
    interface fxp0.0 {
        disable;
    }
}
bgp {
    group to-PE2 {
        type internal;
        local-address 10.255.102.166;
        family inet-vpn {
            unicast;
        }
        export direct;
        neighbor 10.255.102.128;
    }
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface all;
        interface fxp0.0 {
            disable;
        }
        interface ge-0/0/2.0 {
            metric 100;
        }
    }
}
```

```
user@PE1# show policy-options
policy-statement direct {
    term 1 {
```

```
        from protocol direct;
        then accept;
    }
}
policy-statement pplb {
    then load-balance {
        per-packet;
    }
}
```

```
user@PE1# show routing-instances
vpn1 {
    instance-type vrf;
    interface ge-0/0/0.0;
    route-distinguisher 10.255.102.166:1;
    vrf-target target:1:1;
    vrf-table-label;
}
```

Verification

IN THIS SECTION

- Verifying the Container LSP Status Without Bandwidth | **757**
- Verifying the Container LSP Status with Increased Bandwidth (Before Normalization) | **760**
- Verifying the Container LSP Status with Increased Bandwidth (After Normalization) | **762**
- Verifying the Container LSP Splitting Process | **766**
- Verifying the Container LSP Statistics | **767**
- Verifying the Container LSP Status with Decreased Bandwidth (Before Normalization) | **768**
- Verifying the Container LSP Status with Decreased Bandwidth (After Normalization) | **768**
- Verifying the Container LSP Merging Process | **769**
- Verifying Failover Normalization | **770**
- Verifying Incremental Normalization | **772**

Confirm that the configuration is working properly.

Verifying the Container LSP Status Without Bandwidth

Purpose

Verify the status of the container LSP.

Action

From operational mode, run the **show mpls container-lsp extensive** command.

```
user@PE1> show mpls container-lsp extensive
Ingress LSP: 1 sessions
Container LSP name: PE1-PE2-container-100, State: Up, Member count: 2
 Normalization
  Min LSPs: 2, Max LSPs: 20
  Aggregate bandwidth: 20Mbps, Sampled Aggregate bandwidth: 0bps
  NormalizeTimer: 400 secs, NormalizeThreshold: 10%
  Max Signaling BW: 10Mbps, Min Signaling BW: 10Mbps, Splitting BW: 40Mbps, Merging BW: 6Mbps
  Mode: incremental-normalization, failover-normalization
  Sampling: Outlier cut-off 1, Percentile 90 of Aggregate
  Normalization in 143 second(s)
   36 Jun  5 04:12:17.497 Clear history and statistics: on container (PE1-PE2-container-100)
   35 Jun  5 04:12:17.497 Avoid normalization: not needed with bandwidth 0 bps
   34 Jun  5 04:05:37.484 Clear history and statistics: on container (PE1-PE2-container-100)
   33 Jun  5 04:05:37.484 Avoid normalization: not needed with bandwidth 0 bps
   32 Jun  5 03:58:57.470 Clear history and statistics: on container (PE1-PE2-container-100)
   31 Jun  5 03:58:57.470 Avoid normalization: not needed with bandwidth 0 bps
   30 Jun  5 03:52:17.455 Clear history and statistics: on container (PE1-PE2-container-100)
   29 Jun  5 03:52:17.455 Avoid normalization: not needed with bandwidth 0 bps
   28 Jun  5 03:45:37.440 Clear history and statistics: on container (PE1-PE2-container-100)
   27 Jun  5 03:45:37.440 Avoid normalization: not needed with bandwidth 0 bps
   26 Jun  5 03:38:59.013 Normalization complete: container (PE1-PE2-container-100) with 2
members
   25 Jun  5 03:38:57.423 Delete member LSPs: PE1-PE2-container-100-3 through PE1-PE2-
container-100-7
   24 Jun  5 03:38:57.423 Normalize: container (PE1-PE2-container-100) create 2 LSPs, min bw
10000000bps, member count 7
   23 Jun  5 03:38:57.423 Normalize: normalization with aggregate bandwidth 0 bps
   22 Jun  5 03:32:19.019 Normalization complete: container (PE1-PE2-container-100) with 7
members
   21 Jun  5 03:32:17.404 Clear history and statistics: on container (PE1-PE2-container-100)
   20 Jun  5 03:32:17.403 Normalize: container (PE1-PE2-container-100) into 7 members - each
```

```
with bandwidth 10000000 bps
   19 Jun  5 03:32:17.403 Normalize: normalization with aggregate bandwidth 62914560 bps
   18 Jun  5 03:32:17.403 Normalize: normalizaton with 62914560 bps
   17 Jun  5 03:32:09.219 Normalization complete: container (PE1-PE2-container-100) with 7
members
   16 Jun  5 03:32:07.600 Clear history and statistics: on container (PE1-PE2-container-100)
   15 Jun  5 03:32:07.600 Normalize: container (PE1-PE2-container-100) into 7 members - each
with bandwidth 10000000 bps
   14 Jun  5 03:32:07.599 Normalize: normalization with aggregate bandwidth 62914560 bps
   13 Jun  5 03:32:07.599 Normalize: normalizaton with 62914560 bps
   12 Jun  5 03:26:57.295 Clear history and statistics: on container (PE1-PE2-container-100)
   11 Jun  5 03:26:57.295 Avoid normalization: not needed with bandwidth 0 bps
   10 Jun  5 03:20:18.297 Normalization complete: container (PE1-PE2-container-100) with 2
members
    9 Jun  5 03:20:17.281 Normalize: container (PE1-PE2-container-100) create 2 LSPs, min bw
10000000bps, member count 0
    8 Jun  5 03:20:17.281 Normalize: normalization with aggregate bandwidth 0 bps
    7 Jun  5 03:17:43.218 Clear history and statistics: on container (PE1-PE2-container-100)
    6 Jun  5 03:17:43.218 Avoid normalization: not needed with bandwidth 0 bps
    5 Jun  5 03:17:43.212 Normalize: container (PE1-PE2-container-100) received PathErr on
member PE1-PE2-container-100-2
    4 Jun  5 03:17:43.212 Normalize: container (PE1-PE2-container-100) received PathErr on
member PE1-PE2-container-100-1
    3 Jun  5 03:12:47.323 Normalization complete: container (PE1-PE2-container-100) with 2
members
    2 Jun  5 03:12:16.555 Normalize: container (PE1-PE2-container-100) create 2 LSPs, min bw
10000000bps, member count 0
    1 Jun  5 03:12:16.555 Normalize: normalization with aggregate bandwidth 0 bps

10.255.102.128
  From: 10.255.102.166, State: Up, ActiveRoute: 0, LSPname: PE1-PE2-container-100-1
  ActivePath:  (primary)
  LSPtype: Dynamic Configured, Penultimate hop popping
  LoadBalance: Random
  Autobandwidth
  MinBW: 10Mbps, MaxBW: 10Mbps
  AdjustTimer: 300 secs
  Max AvgBW util: 0bps, Bandwidth Adjustment in 12 second(s).
  Overflow limit: 0, Overflow sample count: 0
  Underflow limit: 0, Underflow sample count: 0, Underflow Max AvgBW: 0bps
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                   State: Up
    Priorities: 7 0
```

```
    Bandwidth: 10Mbps
    SmartOptimizeTimer: 180
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 3)
 10.10.10.2 S 10.20.20.2 S 10.30.30.2 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
          10.10.10.2 10.20.20.2 10.30.30.2
   17 Jun  5 03:38:59.013 Make-before-break: Switched to new instance
   16 Jun  5 03:38:58.003 Record Route:  10.10.10.2 10.20.20.2 3=10.30.30.2
   15 Jun  5 03:38:58.003 Up
   14 Jun  5 03:38:57.423 Originate make-before-break call
   13 Jun  5 03:38:57.423 CSPF: computation result accepted  10.10.10.2 10.20.20.2 10.30.30.2
   12 Jun  5 03:33:30.400 CSPF: computation result ignored, new path no benefit
   11 Jun  5 03:32:17.403 Pending old path instance deletion
   10 Jun  5 03:32:09.218 Make-before-break: Switched to new instance
    9 Jun  5 03:32:08.202 Record Route:  10.10.10.2 10.20.20.2 10.30.30.2
    8 Jun  5 03:32:08.202 Up
    7 Jun  5 03:32:07.603 Originate make-before-break call
    6 Jun  5 03:32:07.603 CSPF: computation result accepted  10.10.10.2 10.20.20.2 10.30.30.2
    5 Jun  5 03:20:18.278 Selected as active path
    4 Jun  5 03:20:18.277 Record Route:  10.10.10.2 10.20.20.2 10.30.30.2
    3 Jun  5 03:20:18.277 Up
    2 Jun  5 03:20:17.281 Originate Call
    1 Jun  5 03:20:17.281 CSPF: computation result accepted  10.10.10.2 10.20.20.2 10.30.30.2
  Created: Thu Jun  5 03:20:16 2014

10.255.102.128
  From: 10.255.102.166, State: Up, ActiveRoute: 0, LSPname: PE1-PE2-container-100-2
  ActivePath:  (primary)
  LSPtype: Dynamic Configured, Penultimate hop popping
  LoadBalance: Random
  Autobandwidth
  MinBW: 10Mbps, MaxBW: 10Mbps
  AdjustTimer: 300 secs
  Max AvgBW util: 0bps, Bandwidth Adjustment in 76 second(s).
  Overflow limit: 0, Overflow sample count: 0
  Underflow limit: 0, Underflow sample count: 0, Underflow Max AvgBW: 0bps
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                   State: Up
    Priorities: 7 0
    Bandwidth: 10Mbps
    SmartOptimizeTimer: 180
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 3)
  10.10.10.2 S 10.20.20.2 S 10.30.30.2 S
```

```
     Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
           10.10.10.2 10.20.20.2 10.30.30.2
    17 Jun  5 03:38:59.013 Make-before-break: Switched to new instance
    16 Jun  5 03:38:58.002 Record Route:  10.10.10.2 10.20.20.2 10.30.30.2
    15 Jun  5 03:38:58.002 Up
    14 Jun  5 03:38:57.423 Originate make-before-break call
    13 Jun  5 03:38:57.423 CSPF: computation result accepted  10.10.10.2 10.20.20.2 10.30.30.2
    12 Jun  5 03:33:26.189 CSPF: computation result ignored, new path no benefit
    11 Jun  5 03:32:17.403 Pending old path instance deletion
    10 Jun  5 03:32:09.219 Make-before-break: Switched to new instance
     9 Jun  5 03:32:08.204 Record Route:  10.10.10.2 10.20.20.2 10.30.30.2
     8 Jun  5 03:32:08.204 Up
     7 Jun  5 03:32:07.603 Originate make-before-break call
     6 Jun  5 03:32:07.603 CSPF: computation result accepted  10.10.10.2 10.20.20.2 10.30.30.2
     5 Jun  5 03:20:18.297 Selected as active path
     4 Jun  5 03:20:18.295 Record Route:  10.10.10.2 10.20.20.2 10.30.30.2
     3 Jun  5 03:20:18.295 Up
     2 Jun  5 03:20:17.281 Originate Call
     1 Jun  5 03:20:17.281 CSPF: computation result accepted  10.10.10.2 10.20.20.2 10.30.30.2
  Created: Thu Jun  5 03:20:16 2014
Total 2 displayed, Up 2, Down 0

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Meaning

The container LSP is established between Routers PE1 and PE2.

Verifying the Container LSP Status with Increased Bandwidth (Before Normalization)

Purpose

Verify the status of the container LSP with increased bandwidth before normalization happens.

Action

From operational mode, run the **show mpls container-lsp extensive** command.

```
user@PE1> show mpls container-lsp extensive
Ingress LSP: 1 sessions
Container LSP name: PE1-PE2-container-100, State: Up, Member count: 2
 Normalization
  Min LSPs: 2, Max LSPs: 20
  Aggregate bandwidth: 20Mbps, Sampled Aggregate bandwidth: 42.6984Mbps
  NormalizeTimer: 400 secs, NormalizeThreshold: 10%
  Max Signaling BW: 10Mbps, Min Signaling BW: 10Mbps, Splitting BW: 40Mbps, Merging BW: 6Mbps
  Mode: incremental-normalization, failover-normalization
  Sampling: Outlier cut-off 1, Percentile 90 of Aggregate
  Normalization in 321 second(s)
    3 Jun  5 21:22:34.731 Normalization complete: container (PE1-PE2-container-100) with 2
members
    2 Jun  5 21:22:15.503 Normalize: container (PE1-PE2-container-100) create 2 LSPs, min bw
10000000bps, member count 0
    1 Jun  5 21:22:15.503 Normalize: normalization with aggregate bandwidth 0 bps

10.255.102.128
  From: 10.255.102.166, State: Up, ActiveRoute: 0, LSPname: PE1-PE2-container-100-1
  ActivePath:  (primary)
  Link protection desired
  LSPtype: Dynamic Configured, Penultimate hop popping
  LoadBalance: Random
  Autobandwidth
  MinBW: 10Mbps, MaxBW: 10Mbps
  AdjustTimer: 300 secs AdjustThreshold: 5%
  Max AvgBW util: 23.9893Mbps, Bandwidth Adjustment in 221 second(s).
  Overflow limit: 0, Overflow sample count: 6
  Underflow limit: 0, Underflow sample count: 0, Underflow Max AvgBW: 0bps
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                   State: Up
    Priorities: 7 0
    Bandwidth: 10Mbps
    OptimizeTimer: 30
    SmartOptimizeTimer: 180
    Reoptimization in 9 second(s).
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 3)
 10.10.10.2 S 10.20.20.2 S 10.30.30.2 S
```

```
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
          10.255.102.166(flag=0x20) 10.10.10.2(Label=303440) 10.255.102.29(flag=0x20)
10.20.20.2(Label=302144) 10.255.102.128(flag=0x20) 10.30.30.2(Label=3)

10.255.102.128
  From: 10.255.102.166, State: Up, ActiveRoute: 0, LSPname: PE1-PE2-container-100-2
  ActivePath:  (primary)
  Link protection desired
  LSPtype: Dynamic Configured, Penultimate hop popping
  LoadBalance: Random
  Autobandwidth
  MinBW: 10Mbps, MaxBW: 10Mbps
  AdjustTimer: 300 secs AdjustThreshold: 5%
  Max AvgBW util: 22.1438Mbps, Bandwidth Adjustment in 221 second(s).
  Overflow limit: 0, Overflow sample count: 6
  Underflow limit: 0, Underflow sample count: 0, Underflow Max AvgBW: 0bps
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                   State: Up
    Priorities: 7 0
    Bandwidth: 10Mbps
    OptimizeTimer: 30
    SmartOptimizeTimer: 180
    Reoptimization in 9 second(s).
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 3)
 10.10.10.2 S 10.20.20.2 S 10.30.30.2 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
          10.255.102.166(flag=0x20) 10.10.10.2(Label=303456) 10.255.102.29(flag=0x20)
10.20.20.2(Label=302160) 10.255.102.128(flag=0x20) 10.30.30.2(Label=3)

Total 2 displayed, Up 2, Down 0
```

Meaning

Because normalization has not happened, the member LSP count remains at 2.

Verifying the Container LSP Status with Increased Bandwidth (After Normalization)

Purpose

Verify the status of the container LSP with increased bandwidth after normalization happens.

Action

From operational mode, run the **show mpls container-lsp extensive** command.

```
user@PE1> show mpls container-lsp extensive
Ingress LSP: 1 sessions
Container LSP name: PE1-PE2-container-100, State: Up, Member count: 5
 Normalization
  Min LSPs: 2, Max LSPs: 20
  Aggregate bandwidth: 50Mbps, Sampled Aggregate bandwidth: 45.8873Mbps
  NormalizeTimer: 400 secs, NormalizeThreshold: 10%
  Max Signaling BW: 10Mbps, Min Signaling BW: 10Mbps, Splitting BW: 40Mbps, Merging BW: 6Mbps
  Mode: incremental-normalization, failover-normalization
  Sampling: Outlier cut-off 1, Percentile 90 of Aggregate
  Normalization in 169 second(s)
    7 Jun  5 21:29:02.921 Normalization complete: container (PE1-PE2-container-100) with 5
members
    6 Jun  5 21:28:55.505 Clear history and statistics: on container (PE1-PE2-container-100)
    5 Jun  5 21:28:55.505 Normalize: container (PE1-PE2-container-100) into 5 members - each
with bandwidth 10000000 bps
    4 Jun  5 21:28:55.504 Normalize: normalization with aggregate bandwidth 45281580 bps
    3 Jun  5 21:22:34.731 Normalization complete: container (PE1-PE2-container-100) with 2
members
    2 Jun  5 21:22:15.503 Normalize: container (PE1-PE2-container-100) create 2 LSPs, min bw
10000000bps, member count 0
    1 Jun  5 21:22:15.503 Normalize: normalization with aggregate bandwidth 0 bps

10.255.102.128
  From: 10.255.102.166, State: Up, ActiveRoute: 0, LSPname: PE1-PE2-container-100-1
  ActivePath:  (primary)
  Link protection desired
  LSPtype: Dynamic Configured, Penultimate hop popping
  LoadBalance: Random
  Autobandwidth
  MinBW: 10Mbps, MaxBW: 10Mbps
  AdjustTimer: 300 secs AdjustThreshold: 5%
  Max AvgBW util: 11.0724Mbps, Bandwidth Adjustment in 129 second(s).
  Overflow limit: 0, Overflow sample count: 1
  Underflow limit: 0, Underflow sample count: 0, Underflow Max AvgBW: 0bps
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                   State: Up
    Priorities: 7 0
```

```
    Bandwidth: 10Mbps
    OptimizeTimer: 30
    SmartOptimizeTimer: 180
    Reoptimization in 12 second(s).
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 3)
 10.10.10.2 S 10.20.20.2 S 10.30.30.2 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
          10.255.102.166(flag=0x20) 10.10.10.2(Label=303488) 10.255.102.29(flag=0x20)
10.20.20.2(Label=302224) 10.255.102.128(flag=0x20) 10.30.30.2(Label=3)

10.255.102.128
  From: 10.255.102.166, State: Up, ActiveRoute: 0, LSPname: PE1-PE2-container-100-2
  ActivePath:  (primary)
  Link protection desired
  LSPtype: Dynamic Configured, Penultimate hop popping
  LoadBalance: Random
  Autobandwidth
  MinBW: 10Mbps, MaxBW: 10Mbps
  AdjustTimer: 300 secs AdjustThreshold: 5%
  Max AvgBW util: 8.50751Mbps, Bandwidth Adjustment in 189 second(s).
  Overflow limit: 0, Overflow sample count: 0
  Underflow limit: 0, Underflow sample count: 11, Underflow Max AvgBW: 8.50751Mbps
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                   State: Up
    Priorities: 7 0
    Bandwidth: 10Mbps
    OptimizeTimer: 30
    SmartOptimizeTimer: 180
    Reoptimization in 6 second(s).
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 3)
 10.10.10.2 S 10.20.20.2 S 10.30.30.2 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
          10.255.102.166(flag=0x20) 10.10.10.2(Label=303504) 10.255.102.29(flag=0x20)
10.20.20.2(Label=302240) 10.255.102.128(flag=0x20) 10.30.30.2(Label=3)

10.255.102.128
  From: 10.255.102.166, State: Up, ActiveRoute: 0, LSPname: PE1-PE2-container-100-3
  ActivePath:  (primary)
  Link protection desired
  LSPtype: Dynamic Configured, Penultimate hop popping
  LoadBalance: Random
  Autobandwidth
  MinBW: 10Mbps, MaxBW: 10Mbps
```

```
  AdjustTimer: 300 secs AdjustThreshold: 5%
  Max AvgBW util: 9.59422Mbps, Bandwidth Adjustment in 249 second(s).
  Overflow limit: 0, Overflow sample count: 0
  Underflow limit: 0, Underflow sample count: 5, Underflow Max AvgBW: 9.59422Mbps
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                   State: Up
    Priorities: 7 0
    Bandwidth: 10Mbps
    OptimizeTimer: 30
    SmartOptimizeTimer: 180
    Reoptimization in 25 second(s).
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 3)
 10.10.10.2 S 10.20.20.2 S 10.30.30.2 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
          10.255.102.166(flag=0x20) 10.10.10.2(Label=303472) 10.255.102.29(flag=0x20)
10.20.20.2(Label=302176) 10.255.102.128(flag=0x20) 10.30.30.2(Label=3)

10.255.102.128
  From: 10.255.102.166, State: Up, ActiveRoute: 0, LSPname: PE1-PE2-container-100-4
  ActivePath:  (primary)
  Link protection desired
  LSPtype: Dynamic Configured, Penultimate hop popping
  LoadBalance: Random
  Autobandwidth
  MinBW: 10Mbps, MaxBW: 10Mbps
  AdjustTimer: 300 secs AdjustThreshold: 5%
  Max AvgBW util: 9.16169Mbps, Bandwidth Adjustment in 9 second(s).
  Overflow limit: 0, Overflow sample count: 0
  Underflow limit: 0, Underflow sample count: 29, Underflow Max AvgBW: 9.16169Mbps
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                   State: Up
    Priorities: 7 0
    Bandwidth: 10Mbps
    OptimizeTimer: 30
    SmartOptimizeTimer: 180
    Reoptimization in 1 second(s).
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 3)
 10.10.10.2 S 10.20.20.2 S 10.30.30.2 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
          10.255.102.166(flag=0x20) 10.10.10.2(Label=303520) 10.255.102.29(flag=0x20)
10.20.20.2(Label=302192) 10.255.102.128(flag=0x20) 10.30.30.2(Label=3)

10.255.102.128
```

```
  From: 10.255.102.166, State: Up, ActiveRoute: 0, LSPname: PE1-PE2-container-100-5
  ActivePath:  (primary)
  Link protection desired
  LSPtype: Dynamic Configured, Penultimate hop popping
  LoadBalance: Random
  Autobandwidth
  MinBW: 10Mbps, MaxBW: 10Mbps
  AdjustTimer: 300 secs AdjustThreshold: 5%
  Max AvgBW util: 8.39908Mbps, Bandwidth Adjustment in 69 second(s).
  Overflow limit: 0, Overflow sample count: 0
  Underflow limit: 0, Underflow sample count: 23, Underflow Max AvgBW: 8.39908Mbps
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                   State: Up
    Priorities: 7 0
    Bandwidth: 10Mbps
    OptimizeTimer: 30
    SmartOptimizeTimer: 180
    Reoptimization in 17 second(s).
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 3)
 10.10.10.2 S 20.20.20.2 S 30.30.30.2 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
          10.255.102.166(flag=0x20) 10.10.10.2(Label=303536) 10.255.102.29(flag=0x20)
10.20.20.2(Label=302208) 10.255.102.128(flag=0x20) 10.30.30.2(Label=3)
Total 5 displayed, Up 5, Down 0

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Meaning

At the expiry of the normalization timer, the container LSP is split into five member LSPs, each with 10 Mbps (minimum and maximum signaling bandwidth). As a result, the aggregate bandwidth is 50 Mbps.

Verifying the Container LSP Splitting Process

Purpose

Verify the container LSP splitting process after normalization happens.

Action

From operational mode, run the **show route 10.2.2.0** command.

```
user@PE1> show route 10.2.2.0
vpn1.inet.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.2.2.0/24         *[BGP/170] 00:12:14, localpref 100, from 10.255.102.128
                      AS path: I, validation-state: unverified
>to 10.10.10.2 via ge-0/0/1.0,label-switched-path PE1-PE2-container100-1
to 10.10.10.2 via ge-0/0/1.0, label-switched-path PE1-PE2-container100-2
to 10.10.10.2 via ge-0/0/1.0, label-switched-path PE1-PE2-container100-3
to 10.10.10.2 via ge-0/0/1.0, label-switched-path PE1-PE2-container100-4
to 10.10.10.2 via ge-0/0/1.0, label-switched-path PE1-PE2-container100-5
```

Meaning

After LSP splitting, Router PE1 has injected the forwarding adjacency.

Verifying the Container LSP Statistics

Purpose

Verify the container LSP statistics after normalization happens.

Action

From operational mode, run the **show mpls container-lsp statistics** command.

```
user@PE1> show mpls container-lsp statistics
Ingress LSP: 1 sessions
Container LSP name                              State     Member LSP count
PE1-PE2-container-100                           Up                5
To              From            State     Packets            Bytes LSPname
10.255.102.128   10.255.102.166  Up   15166271       2062612856 PE1-PE2-container-100-1
10.255.102.128   10.255.102.166  Up   12289912       1671428032 PE1-PE2-container-100-2
10.255.102.128   10.255.102.166  Up   13866911       1885899896 PE1-PE2-container-100-3
```

```
10.255.102.128   10.255.102.166  Up   12558707       1707984152 PE1-PE2-container-100-4
10.255.102.128   10.255.102.166  Up   11512151       1565652536 PE1-PE2-container-100-5
```

Meaning

Traffic is load-balanced across the newly created member LSPs.

Verifying the Container LSP Status with Decreased Bandwidth (Before Normalization)

Purpose

Verify the status of the container LSP with decreased bandwidth before normalization happens.

Action

From operational mode, run the **show mpls container-lsp detail** command.

```
user@PE1> show mpls container-lsp detail
Ingress LSP: 1 sessions
Container LSP name: PE1-PE2-container-100, State: Up, Member count: 5
 Normalization
  Min LSPs: 2, Max LSPs: 20
  Aggregate bandwidth: 50Mbps, Sampled Aggregate bandwidth: 2.0215Mbps
  NormalizeTimer: 400 secs, NormalizeThreshold: 10%
  Max Signaling BW: 10Mbps, Min Signaling BW: 10Mbps, Splitting BW: 40Mbps, Merging BW: 6Mbps
  Mode: incremental-normalization, failover-normalization
  Sampling: Outlier cut-off 1, Percentile 90 of Aggregate
  Normalization in 384 second(s)
---Output truncated---
```

Meaning

Because normalization has not happened, the member LSP count remains at 5.

Verifying the Container LSP Status with Decreased Bandwidth (After Normalization)

Purpose

Verify the status of the container LSP with decreased bandwidth after normalization happens.

Action

From operational mode, run the **show mpls container-lsp detail** command.

```
user@PE1> show mpls container-lsp detail
Ingress LSP: 1 sessions
Container LSP name: PE1-PE2-container-100, State: Up, Member count: 2
 Normalization
  Min LSPs: 2, Max LSPs: 20
  Aggregate bandwidth: 20Mbps, Sampled Aggregate bandwidth: 0bps
  NormalizeTimer: 400 secs, NormalizeThreshold: 10%
  Max Signaling BW: 10Mbps, Min Signaling BW: 10Mbps, Splitting BW: 40Mbps, Merging BW: 6Mbps
  Mode: incremental-normalization, failover-normalization
  Sampling: Outlier cut-off 1, Percentile 90 of Aggregate
  Normalization in 397 second(s)
   22 Jun  5 22:30:37.094 Clear history and statistics: on container (PE1-PE2-container-100)
   21 Jun  5 22:30:37.094 Delete member LSPs: PE1-PE2-container-100-3 through PE1-PE2-
container-100-5
   20 Jun  5 22:30:37.090 Normalize: container (PE1-PE2-container-100) into 2 members - each
with bandwidth 10000000 bps
   19 Jun  5 22:30:37.090 Normalize: normalization with aggregate bandwidth 2037595 bps
   18 Jun  5 22:30:37.090 Normalize: normalizaton with 2037595 bps
---Output truncated---
```

Meaning

At the expiry of the normalization timer, the container LSP merging takes place because there is an overall reduction in bandwidth. The member LSPs are merged, and the member LSP count is 2 after normalization.

Verifying the Container LSP Merging Process

Purpose

Verify the container LSP splitting process after normalization happens.

Action

From operational mode, run the **show route 10.2.2.0** command.

```
user@PE1> show route 10.2.2.0
vpn1.inet.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.2.2.0/24         *[BGP/170] 01:09:45, localpref 100, from 10.255.102.128
                      AS path: I, validation-state: unverified
                    > to 10.10.10.2 via ge-0/0/1.0, label-switched-path PE1-PE2-container-100-1
                      to 10.10.10.2 via ge-0/0/1.0, label-switched-path PE1-PE2-container-100-2
```

Meaning

After LSP merging, Router PE1 has deleted the merged member LSPs.

Verifying Failover Normalization

Purpose

Verify load redistribution when traffic is sent at 35 Mbps and the link between Routers P1 and P2 is disabled. Arrival of PathErr on link failure triggers immediate normalization.

To enable failover normalization, include the `failover-normalization` configuration statement at the `[edit protocols mpls container-label-switched-path container-lsp-name splitting-merging normalization]` hierarchy level.

Action

From operational mode, run the **show mpls container-lsp** command.

```
user@PE1> show mpls container-lsp
Ingress LSP: 1 sessions
Container LSP name                              State     Member LSP count
PE1-PE2-container-100                            Up               2
To              From            State Rt P     ActivePath       LSPname
10.255.102.128   10.255.102.166  Up     0 *                      PE1-PE2-container-100-1
10.255.102.128   10.255.102.166  Up     0 *                      PE1-PE2-container-100-2
Total 2 displayed, Up 2, Down 0
```

After the ge-0/0/2 link between Routers P1 and P2 goes down, normalization is immediately triggered.

From operational mode, run the **show mpls container-lsp detail** command.

```
user@PE1> show mpls container-lsp detail
Ingress LSP: 1 sessions
Container LSP name: PE1-PE2-container-100, State: Up, Member count: 4
 Normalization
  Min LSPs: 2, Max LSPs: 20
  Aggregate bandwidth: 40Mbps, Sampled Aggregate bandwidth: 34.5538Mbps
  NormalizeTimer: 3000 secs, NormalizeThreshold: 10%
  Max Signaling BW: 10Mbps, Min Signaling BW: 10Mbps, Splitting BW: 40Mbps, Merging BW: 6Mbps
  Mode: incremental-normalization, failover-normalization
  Sampling: Outlier cut-off 1, Percentile 90 of Aggregate
  Normalization in 2970 second(s)
   11 Jun 5 19:28:27.564 Normalization complete: container (PE1-PE2-container-100) with 4 members
   10 Jun 5 19:28:20.315 Normalize: container (PE1-PE2-container-100) received PathErr on member
PE1-PE2-container-100-2[2 times]
    9 Jun 5 19:28:20.315 Normalize: container (PE1-PE2-container-100) received PathErr on member
PE1-PE2-container-100-1[2 times]
    8 Jun 5 19:28:20.311 Clear history and statistics: on container (PE1-PE2-container-100)
    7 Jun 5 19:28:20.311 Normalize: container (PE1-PE2-container-100) into 4 members - each with
bandwidth 10000000 bps
    6 Jun 5 19:28:20.311 Normalize: normalization with aggregate bandwidth 33665020 bps
    5 Jun 5 19:28:20.308 Normalize: container (PE1-PE2-container-100) received PathErr on member
PE1-PE2-container-100-2
    4 Jun 5 19:28:20.308 Normalize: container (PE1-PE2-container-100) received PathErr on member
PE1-PE2-container-100-1
    3 Jun 5 19:27:48.574 Normalization complete: container (PE1-PE2-container-100) with 2 members
    2 Jun 5 19:27:28.644 Normalize: container (PE1-PE2-container-100) create 2 LSPs, min bw
10000000bps, member count 0
    1 Jun 5 19:27:28.644 Normalize: normalization with aggregate bandwidth 0 bps
----Output truncated----
```

Meaning

Arrival of PathErr message on link failure triggers immediate normalization.

Verifying Incremental Normalization

Purpose

Verify incremental normalization when enough bandwidth is not available.

On Router PE1, the RSVP interfaces static bandwidth is restricted to 22 Mbps each.

Action

From operational mode, run the **show rsvp interface** command.

```
user@PE1> show rsvp interface
RSVP interface: 4 active
                  Active Subscr- Static      Available   Reserved    Highwater
Interface   State resv   iption  BW          BW          BW          mark
ge-0/0/2.0  Up         0   100%  22Mbps      22Mbps      0bps        21.4031Mbps
ge-0/0/1.0  Up         2   100%  22Mbps      12Mbps      10Mbps      21.7011Mbps
```

Before normalization happens:

From operational mode, run the **show mpls container-lsp** command.

```
user@PE1> show mpls container-lsp
Ingress LSP: 1 sessions
Container LSP name                              State     Member LSP count
PE1-PE2-container-100                           Up                2
To              From            State Rt P     ActivePath       LSPname
10.255.102.128   10.255.102.166  Up     0 *                      PE1-PE2-container-100-1
10.255.102.128   10.255.102.166  Up     0 *                      PE1-PE2-container-100-2
```

After normalization happens:

From operational mode, run the **show mpls container-lsp** command.

```
user@PE1> show mpls container-lsp
Ingress LSP: 1 sessions
Container LSP name                              State     Member LSP count
PE1-PE2-container-100                           Up                7
To              From            State Rt P  ActivePath LSPname
10.255.102.128   10.255.102.166  Up     0 *       PE1-PE2-container-100-1
```

```
10.255.102.128   10.255.102.166  Up     0 *       PE1-PE2-container-100-2
10.255.102.128   10.255.102.166  Up     0 *       PE1-PE2-container-100-3
10.255.102.128   10.255.102.166  Up     0 *       PE1-PE2-container-100-4
10.255.102.128   10.255.102.166  Up     0 *       PE1-PE2-container-100-5
10.255.102.128   10.255.102.166  Up     0 *       PE1-PE2-container-100-6
10.255.102.128   0.0.0.0         Dn     0    -    PE1-PE2-container-100-7
Total 7 displayed, Up 6, Down 1
```

From operational mode, run the **show mpls container-lsp detail** command.

```
user@PE1> show mpls container-lsp detail
Ingress LSP: 1 sessions
Container LSP name: PE1-PE2-container-100, State: Up, Member count: 7
 Normalization
  Min LSPs: 2, Max LSPs: 10
  Aggregate bandwidth: 40.8326Mbps, Sampled Aggregate bandwidth: 50.129Mbps
  NormalizeTimer: 9000 secs, NormalizeThreshold: 10%
  Max Signaling BW: 10Mbps, Min Signaling BW: 5Mbps, Splitting BW: 40Mbps, Merging BW: 5Mbps
  Mode: incremental-normalization, failover-normalization
  Sampling: Outlier cut-off 1, Percentile 90 of Aggregate
  Normalization in 8072 second(s)
   10 Jun 5 18:40:17.812 Normalization complete: container (PE1-PE2-container-100) with 7
members, retry-limit reached
    9 Jun 5 18:40:08.028 Normalize: container (PE1-PE2-container-100) for target member count 7,
member bandwidth 6805439 bps
    8 Jun 5 18:39:58.301 Normalize: container (PE1-PE2-container-100) for target member count 6,
member bandwidth 7939679 bps
    7 Jun 5 18:39:48.470 Clear history and statistics: on container (PE1-PE2-container-100)
    6 Jun 5 18:39:48.470 Normalize: container (PE1-PE2-container-100) into 5 members - each with
bandwidth 9527615 bps
    5 Jun 5 18:39:48.469 Normalize: normalization with aggregate bandwidth 47638076 bps
    4 Jun 5 18:39:48.469 Normalize: normalizaton with 47638076 bps
    3 Jun 5 18:39:09.471 Normalization complete: container (PE1-PE2-container-100) with 2 members
    2 Jun 5 18:38:59.822 Normalize: container (PE1-PE2-container-100) create 2 LSPs, min bw
5000000bps, member count 0
    1 Jun 5 18:38:59.822 Normalize: normalization with aggregate bandwidth 0 bps
```

Meaning

After normalization, the aggregate bandwidth after three retries is 40.8326 Mbps.

Configuring Dynamic Bandwidth Management Using Container LSP

You can configure a container LSP to enable load balancing across multiple point-to-point LSPs dynamically. A container LSP includes one or more member LSPs between the same ingress and egress routing devices. The member LSPs are similar to independent point-to-point LSPs, and each member LSP takes a different path to the same destination and can be routed along a different IGP cost path.

A container LSP provides support for dynamic bandwidth management by enabling the ingress router to dynamically add and remove member LSPs through a process called LSP splitting and LSP merging, respectively, based on configuration and aggregate traffic. Besides addition and deletion, member LSPs can also be re-optimized with different bandwidth values in a make-before-break way.

Before you begin:

1. Configure the device interfaces.
2. Configure the device router ID and autonomous system number.
3. Configure the following protocols:
   - RSVP
   - BGP

     Configure a BGP group to peer device with remote provider edge (PE) device.
   - OSPF

     Enable traffic engineering capabilities.
4. Configure a VRF routing instance.

To configure the PE device:

1. Enable MPLS on all the interfaces (excluding the management interface).

```
[edit protocols]
user@PE1# set mpls interface all
user@PE1# set mpls interface fxp0.0 disable
```

2. Configure the MPLS statistics parameters.

```
[edit protocols]
user@PE1# set mpls statistics file file-name
user@PE1# set mpls statistics file size size
```

```
user@PE1# set mpls statistics interval seconds
user@PE1# set mpls statistics auto-bandwidth
```

3. Configure the label-switched path (LSP) template parameters.

```
[edit protocols]
user@PE1# set mpls label-switched-path template-name template
user@PE1# set mpls label-switched-path template-name optimize-timer seconds
user@PE1# set mpls label-switched-path template-name link-protection
user@PE1# set mpls label-switched-path template-name adaptive
user@PE1# set mpls label-switched-path template-name auto-bandwidth adjust-interval seconds
user@PE1# set mpls label-switched-path template-name auto-bandwidth adjust-threshold seconds
user@PE1# set mpls label-switched-path template-name auto-bandwidth minimum-bandwidth mbps
user@PE1# set mpls label-switched-path template-name auto-bandwidth maximum-bandwidth mbps
```

4. Configure a container LSP between the two PE routers, and assign the LSP template.

```
[edit protocols]
user@PE1# set mpls container-label-switched-path container-lsp-name to remote-PE-ip-address
user@PE1# set mpls container-label-switched-path container-lsp-name label-switched-path-
template template-name
```

5. Configure the container LSP parameters.

```
[edit protocols]
user@PE1# set mpls container-label-switched-path container-lsp-name splitting-merging maximum-
member-lsps number
user@PE1# set mpls container-label-switched-path container-lsp-name splitting-merging minimum-
member-lsps number
user@PE1# set mpls container-label-switched-path container-lsp-name splitting-merging
splitting-bandwidth mbps
user@PE1# set mpls container-label-switched-path container-lsp-name splitting-merging merging-
bandwidth mbps
user@PE1# set mpls container-label-switched-path container-lsp-name splitting-merging maximum-
signaling-bandwidth mbps
user@PE1# set mpls container-label-switched-path container-lsp-name splitting-merging minimum-
signaling-bandwidth mbps
user@PE1# set mpls container-label-switched-path container-lsp-name splitting-merging
normalization normalize-interval seconds
user@PE1# set mpls container-label-switched-path container-lsp-name splitting-merging
normalization failover-normalization
```

```
user@PE1# set mpls container-label-switched-path container-lsp-name splitting-merging
normalization normalization-retry-duration seconds
user@PE1# set mpls container-label-switched-path container-lsp-name splitting-merging
normalization normalization-retry-limits number
user@PE1# set mpls container-label-switched-path container-lsp-name splitting-merging
sampling cut-off-threshold number
user@PE1# set mpls container-label-switched-path container-lsp-name splitting-merging
sampling use-percentile number
```

6. Configure the policy statement to load-balance traffic.

```
[edit policy-options]
user@PE1# set policy-statement first-policy-name term 1 from protocol direct
user@PE1# set policy-statement first-policy-name term 1 then accept
user@PE1# set policy-statement second-policy-name then load-balance per-packet
```

NOTE: The policy to load-balance traffic should be assigned to the forwarding table configuration under the [edit routing-options] hierarchy level.

```
user@PE1# set forwarding-table export pplb
```

7. Verify and commit the configuration.

   For example:

```
[edit protocols]
user@PE1# set rsvp preemption aggressive
user@PE1# set rsvp interface all aggregate
user@PE1# set rsvp interface fxp0.0 disable
user@PE1# set rsvp interface ge-0/0/1.0
user@PE1# set rsvp interface ge-0/0/2.0
user@PE1# set mpls statistics file auto-bw
user@PE1# set mpls statistics file size 10m
user@PE1# set mpls statistics interval 10
user@PE1# set mpls statistics auto-bandwidth
user@PE1# set mpls label-switched-path PE1-to-PE2-template1 template
user@PE1# set mpls label-switched-path PE1-to-PE2-template1 optimize-timer 30
user@PE1# set mpls label-switched-path PE1-to-PE2-template1 link-protection
user@PE1# set mpls label-switched-path PE1-to-PE2-template1 adaptive
user@PE1# set mpls label-switched-path PE1-to-PE2-template1 auto-bandwidth adjust-interval 300
user@PE1# set mpls label-switched-path PE1-to-PE2-template1 auto-bandwidth adjust-threshold 5
```

```
user@PE1# set mpls label-switched-path PE1-to-PE2-template1 auto-bandwidth minimum-bandwidth
10m
user@PE1# set mpls label-switched-path PE1-to-PE2-template1 auto-bandwidth maximum-bandwidth
10m
user@PE1# set mpls label-switched-path PE1-PE2-template-1 template
user@PE1# set mpls label-switched-path PE1-PE2-template-1 auto-bandwidth adjust-interval 8000
user@PE1# set mpls label-switched-path PE1-PE2-template-1 auto-bandwidth minimum-bandwidth 5m
user@PE1# set mpls label-switched-path PE1-PE2-template-1 auto-bandwidth maximum-bandwidth 10m
user@PE1# set mpls container-label-switched-path PE1-PE2-container-100 label-switched-path-
template PE1-to-PE2-template1
user@PE1# set mpls container-label-switched-path PE1-PE2-container-100 to 10.255.102.128
user@PE1# set mpls container-label-switched-path PE1-PE2-container-100 splitting-merging
maximum-member-lsps 20
user@PE1# set mpls container-label-switched-path PE1-PE2-container-100 splitting-merging
minimum-member-lsps 2
user@PE1# set mpls container-label-switched-path PE1-PE2-container-100 splitting-merging
splitting-bandwidth 40m
user@PE1# set mpls container-label-switched-path PE1-PE2-container-100 splitting-merging
merging-bandwidth 6m
user@PE1# set mpls container-label-switched-path PE1-PE2-container-100 splitting-merging
maximum-signaling-bandwidth 10m
user@PE1# set mpls container-label-switched-path PE1-PE2-container-100 splitting-merging
minimum-signaling-bandwidth 10m
user@PE1# set mpls container-label-switched-path PE1-PE2-container-100 splitting-merging
normalization normalize-interval 400
user@PE1# set mpls container-label-switched-path PE1-PE2-container-100 splitting-merging
normalization failover-normalization
user@PE1# set mpls container-label-switched-path PE1-PE2-container-100 splitting-merging
normalization normalization-retry-duration 20
user@PE1# set mpls container-label-switched-path PE1-PE2-container-100 splitting-merging
normalization normalization-retry-limits 3
user@PE1# set mpls container-label-switched-path PE1-PE2-container-100 splitting-merging
sampling cut-off-threshold 1
user@PE1# set mpls container-label-switched-path PE1-PE2-container-100 splitting-merging
sampling use-percentile 90
user@PE1# set mpls interface all
user@PE1# set mpls interface fxp0.0 disable
user@PE1# set bgp group to-PE2 type internal
user@PE1# set bgp group to-PE2 local-address 10.255.102.166
user@PE1# set bgp group to-PE2 family inet-vpn unicast
user@PE1# set bgp group to-PE2 export direct
user@PE1# set bgp group to-PE2 neighbor 10.255.102.128
user@PE1# set ospf traffic-engineering
```

```
user@PE1# set ospf area 0.0.0.0 interface all
user@PE1# set ospf area 0.0.0.0 interface fxp0.0 disable
user@PE1# set ospf area 0.0.0.0 interface ge-0/0/2.0 metric 100
```

```
[edit policy-options]
user@PE1# set policy-statement direct term 1 from protocol direct
user@PE1# set policy-statement direct term 1 then accept
user@PE1# set policy-statement pplb then load-balance per-packet
```

```
[edit]
user@PE1# commit
commit complete
```

RELATED DOCUMENTATION

MPLS Overview | 2

Multiclass LSP Configuration

IN THIS SECTION

- Multiclass LSP Overview | 778
- Multiclass LSPs | 779
- Establishing a Multiclass LSP on the Differentiated Services Domain | 779

Multiclass LSP Overview

A multiclass LSP is an LSP that can carry several class types. One multiclass LSP can be used to support up to four class types. On the packets, the class type is specified by the EXP bits (also known as the class-of-service bits) and the per-hop behavior (PHB) associated with the EXP bits. The mapping between the EXP bits and the PHB is static, rather than being signaled in RSVP.

Once a multiclass LSP is configured, traffic from all of the class types can:

- Follow the same path
- Be rerouted along the same path
- Be taken down at the same time

Class types must be configured consistently across the Differentiated Services domain, meaning the class type configuration must be consistent from router to router in the network.

You can unambiguously map a class type to a queue. On each node router, the CoS queue configuration for an interface translates to the available bandwidth for a particular class type on that link.

The combination of a class type and a priority level forms a traffic engineering class. The IGPs can advertise up to eight traffic engineering classes for each link.

For more information about the EXP bits, see "MPLS Label Allocation" on page 524.

For more information about forwarding classes, see the Junos OS Class of Service User Guide for Routing Devices.

Multiclass LSPs

Multiclass LSPs function like standard LSPs, but they also allow you to configure multiple class types with guaranteed bandwidth. The EXP bits of the MPLS header are used to distinguish between class types. Multiclass LSPs can be configured for a variety of purposes. For example, you can configure a multiclass LSP to emulate the behavior of an ATM circuit. An ATM circuit can provide service-level guarantees to a class type. A multiclass LSP can provide a similar guaranteed level of service.

The following sections discuss multiclass LSPs:

- "Multiclass LSP Overview" on page 778
- "Establishing a Multiclass LSP on the Differentiated Services Domain" on page 779

Establishing a Multiclass LSP on the Differentiated Services Domain

The following occurs when a multiclass LSP is established on the differentiated services domain:

1. The IGPs advertise how much unreserved bandwidth is available for the traffic engineering classes.
2. When calculating the path for a multiclass LSP, CSPF is used to ensure that the constraints are met for all the class types carried by the multiclass LSP (a set of constraints instead of a single constraint).
3. Once a path is found, RSVP signals the LSP using an RSVP object in the path message. At each node in the path, the available bandwidth for the class types is adjusted as the path is set up. The RSVP object is a hop-by-hop object. Multiclass LSPs cannot be established through routers that do not understand this object. Preventing routers that do not understand the RSVP object from carrying

traffic helps to ensure consistency throughout the differentiated services domain by preventing the multiclass LSP from using a router that is incapable of supporting differentiated services.

By default, multiclass LSPs are signaled with setup priority 7 and holding priority 0. A multiclass LSP configured with these values cannot preempt another LSP at setup time and cannot be preempted.

It is possible to have both multiclass LSPs and regular LSPs configured at the same time on the same physical interfaces. For this type of heterogeneous environment, regular LSPs carry best-effort traffic by default. Traffic carried in the regular LSPs must have the correct EXP settings.

RELATED DOCUMENTATION

Basic MPLS Configuration | 46

Point-to-Multipoint LSP Configuration

IN THIS SECTION

- Point-to-Multipoint LSPs Overview | 781
- Understanding Point-to-Multipoint LSPs | 782
- Point-to-Multipoint LSP Configuration Overview | 784
- Example: Configuring a Collection of Paths to Create an RSVP-Signaled Point-to-Multipoint LSP | 784
- Configuring Primary and Branch LSPs for Point-to-Multipoint LSPs | 811
- Configuring Inter-Domain Point-to-Multipoint LSPs | 813
- Configuring Link Protection for Point-to-Multipoint LSPs | 815
- Configuring Graceful Restart for Point-to-Multipoint LSPs | 815
- Configuring a Multicast RPF Check Policy for Point-to-Multipoint LSPs | 816
- Configuring Ingress PE Router Redundancy for Point-to-Multipoint LSPs | 817
- Configuring a Service to Correlate Point-to-Multipoint sub-LSPs with FPCs | 818
- Enabling Point-to-Point LSPs to Monitor Egress PE Routers | 822
- Preserving Point-to-Multipoint LSP Functioning with Different Junos OS Releases | 823
- Re-merge Behavior on Point-to-Multipoint LSP Overview | 823

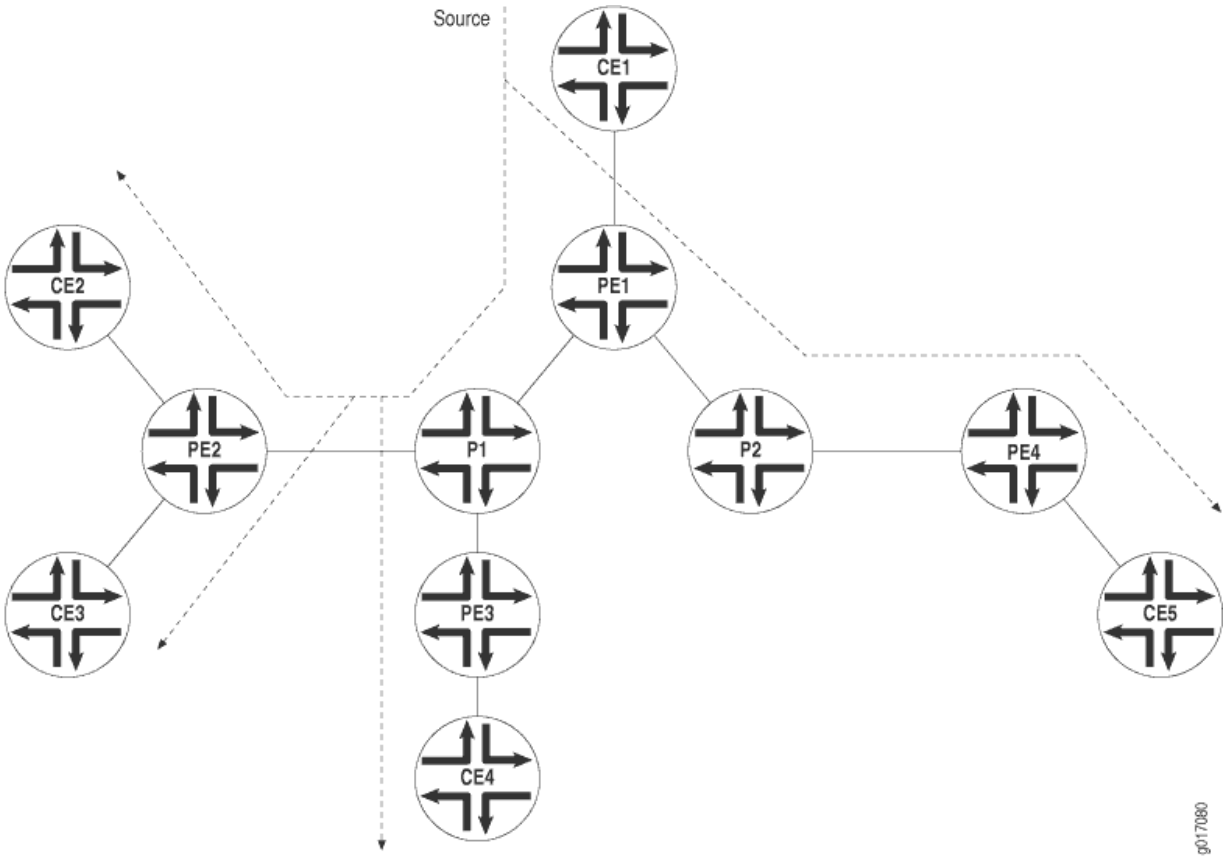
Point-to-Multipoint LSPs Overview

A point-to-multipoint MPLS LSP is an LSP with a single source and multiple destinations. By taking advantage of the MPLS packet replication capability of the network, point-to-multipoint LSPs avoid unnecessary packet replication at the ingress router. Packet replication takes place only when packets are forwarded to two or more different destinations requiring different network paths.

This process is illustrated in Figure 50 on page 781. Router PE1 is configured with a point-to-multipoint LSP to Routers PE2, PE3, and PE4. When Router PE1 sends a packet on the point-to-multipoint LSP to Routers P1 and P2, Router P1 replicates the packet and forwards it to Routers PE2 and PE3. Router P2 sends the packet to Router PE4.

This feature is described in detail in the Internet drafts draft-raggarwa-mpls-p2mp-te-02.txt (expired February 2004), *Establishing Point to Multipoint MPLS TE LSPs*, draft-ietf-mpls-rsvp-te-p2mp-02.txt, *Extensions to Resource Reservation Protocol-Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label-Switched Paths (LSPs)*, and RFC 6388, *Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths* (only point-to-multipoint LSPs are supported).

Figure 50: Point-to-Multipoint LSPs



The following are some of the properties of point-to-multipoint LSPs:

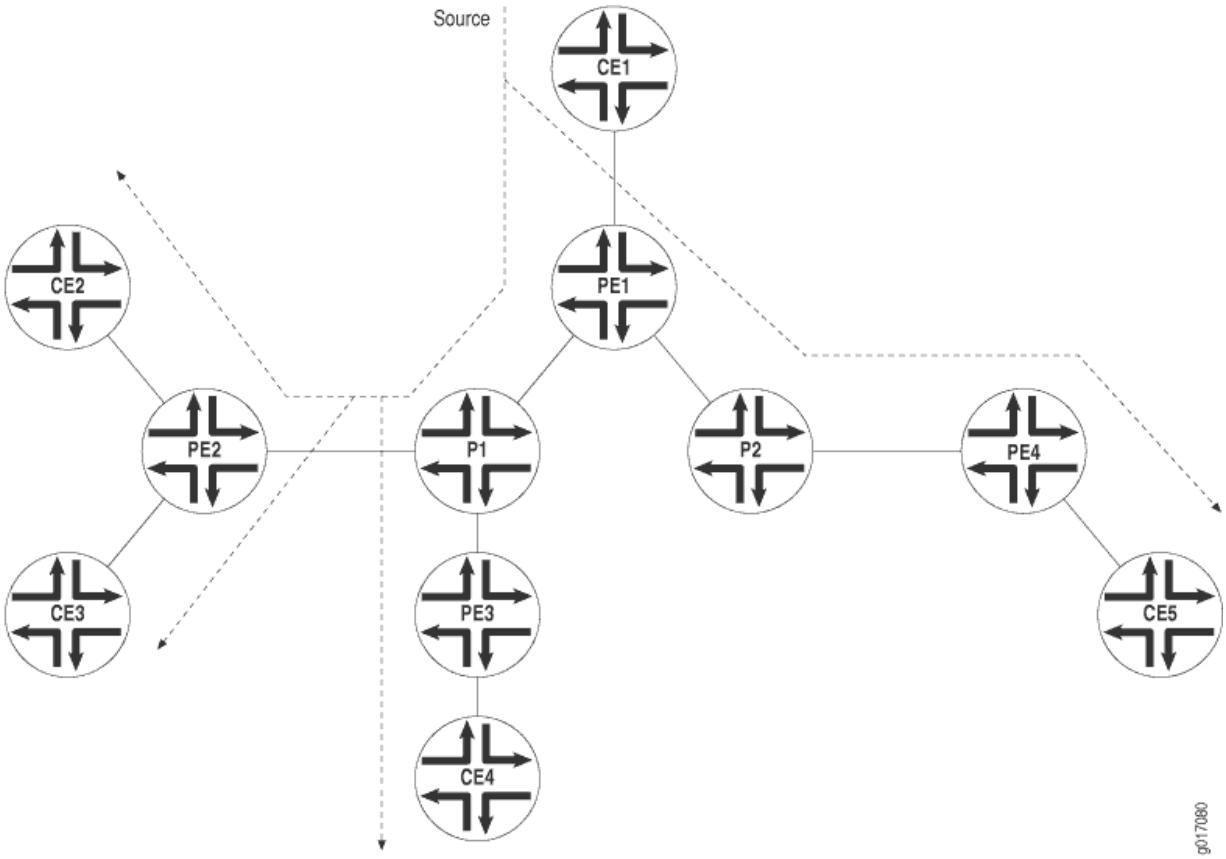
- A point-to-multipoint LSP enables you to use MPLS for point-to-multipoint data distribution. This functionality is similar to that provided by IP multicast.
- You can add and remove branch LSPs from a main point-to-multipoint LSP without disrupting traffic. The unaffected parts of the point-to-multipoint LSP continue to function normally.
- You can configure a node to be both a transit and an egress router for different branch LSPs of the same point-to-multipoint LSP.
- You can enable link protection on a point-to-multipoint LSP. Link protection can provide a bypass LSP for each of the branch LSPs that make up the point-to-multipoint LSP. If any of the primary paths fail, traffic can be quickly switched to the bypass.
- You can configure branch LSPs either statically, dynamically, or as a combination of static and dynamic LSPs.
- You can enable *graceful Routing Engine switchover* (GRES) and graceful restart for point-to-multipoint LSPs at ingress and egress routers. The point-to-multipoint LSPs must be configured using either static routes or circuit cross-connect (CCC). GRES and graceful restart allow the traffic to be forwarded at the Packet Forwarding Engine based on the old state while the control plane recovers. Feature parity for GRES and graceful restart for MPLS point-to-multipoint LSPs on the Junos Trio chipset is supported in Junos OS Releases 11.1R2, 11.2R2, and 11.4.

Understanding Point-to-Multipoint LSPs

A point-to-multipoint MPLS label-switched path (LSP) is an LDP-signaled or RSVP-signaled LSP with a single source and multiple destinations. By taking advantage of the MPLS packet replication capability of the network, point-to-multipoint LSPs avoid unnecessary packet replication at the inbound (ingress) router. Packet replication takes place only when packets are forwarded to two or more different destinations requiring different network paths.

This process is illustrated in Figure 51 on page 783. Device PE1 is configured with a point-to-multipoint LSP to Routers PE2, PE3, and PE4. When Device PE1 sends a packet on the point-to-multipoint LSP to Routers P1 and P2, Device P1 replicates the packet and forwards it to Routers PE2 and PE3. Device P2 sends the packet to Device PE4.

Figure 51: Point-to-Multipoint LSPs



Following are some of the properties of point-to-multipoint LSPs:

- A point-to-multipoint LSP allows you to use MPLS for point-to-multipoint data distribution. This functionality is similar to that provided by IP multicast.
- You can add and remove branch LSPs from a main point-to-multipoint LSP without disrupting traffic. The unaffected parts of the point-to-multipoint LSP continue to function normally.
- You can configure a node to be both a transit and an outbound (egress) router for different branch LSPs of the same point-to-multipoint LSP.
- You can enable link protection on a point-to-multipoint LSP. Link protection can provide a bypass LSP for each of the branch LSPs that make up the point-to-multipoint LSP. If any primary paths fail, traffic can be quickly switched to the bypass.
- You can configure subpaths either statically or dynamically.
- You can enable graceful restart on point-to-multipoint LSPs.

Point-to-Multipoint LSP Configuration Overview

To set up a point-to-multipoint LSP:

1. Configure the primary LSP from the ingress router and the branch LSPs that carry traffic to the egress routers.
2. Specify a pathname on the primary LSP and this same path name on each branch LSP.

NOTE: By default, the branch LSPs are dynamically signaled by means of Constrained Shortest Path First (CSPF) and require no configuration. You can alternatively configure the branch LSPs as static paths.

Example: Configuring a Collection of Paths to Create an RSVP-Signaled Point-to-Multipoint LSP

IN THIS SECTION

- Requirements | 784
- Overview | 784
- Configuration | 786
- Verification | 809

This example shows how to configure a collection of paths to create an RSVP-signaled point-to-multipoint label-switched path (LSP).

Requirements

In this example, no special configuration beyond device initialization is required.

Overview

IN THIS SECTION

- Topology Diagram | 785

In this example, multiple routing devices serve as the transit, branch, and leaf nodes of a single point-to-multipoint LSP. On the provider edge (PE), Device PE1 is the ingress node. The branches go from PE1 to PE2, PE1 to PE3, and PE1 to PE4. Static unicast routes on the ingress node (PE1) point to the egress nodes.

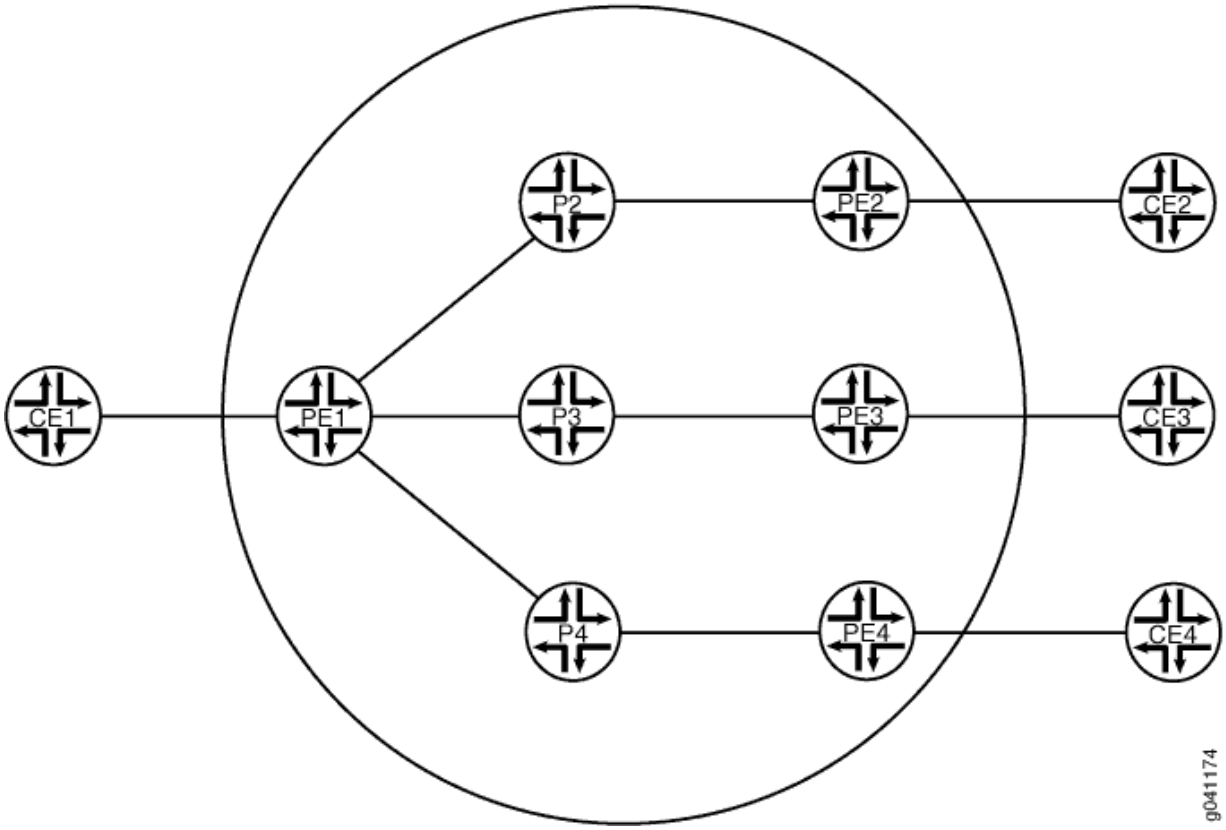
This example also demonstrates static routes with a next hop that is a point-to-multipoint LSP, using the `p2mp-lsp-next-hop` statement. This is useful when implementing filter-based forwarding.

NOTE: Another option is to use the `lsp-next-hop` statement to configure a regular point-to-point LSP to be the next hop. Though not shown in this example, you can optionally assign an independent preference and metric to the next hop.

Topology Diagram

Figure 52 on page 785 shows the topology used in this example.

Figure 52: RSVP-Signaled Point-to-Multipoint LSP



Configuration

IN THIS SECTION

- CLI Quick Configuration | **786**
- Configuring the Ingress Label-Switched Router (LSR) (Device PE1) | **788**
- Configuring the Transit and Egress LSRs (Devices P2, P3, P4, PE2, PE3, and PE4) | **790**
- Configuring Device CE1 | **804**
- Configuring Device CE2 | **805**
- Configuring Device CE3 | **806**
- Configuring Device CE4 | **808**

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the `[edit]` hierarchy level.

Device PE1

```
set interfaces ge-2/0/2 unit 0 description PE1-to-CE1
set interfaces ge-2/0/2 unit 0 family inet address 10.0.244.10/30
set interfaces fe-2/0/10 unit 1 description PE1-to-P2
set interfaces fe-2/0/10 unit 1 family inet address 2.2.2.1/24
set interfaces fe-2/0/10 unit 1 family mpls
set interfaces fe-2/0/9 unit 8 description PE1-to-P3
set interfaces fe-2/0/9 unit 8 family inet address 6.6.6.1/24
set interfaces fe-2/0/9 unit 8 family mpls
set interfaces fe-2/0/8 unit 9 description PE1-to-P4
set interfaces fe-2/0/8 unit 9 family inet address 3.3.3.1/24
set interfaces fe-2/0/8 unit 9 family mpls
set interfaces lo0 unit 1 family inet address 100.10.10.10/32
set protocols rsvp interface fe-2/0/10.1
set protocols rsvp interface fe-2/0/9.8
set protocols rsvp interface fe-2/0/8.9
set protocols rsvp interface lo0.1
set protocols mpls traffic-engineering bgp-igp
set protocols mpls label-switched-path PE1-PE2 to 100.50.50.50
```

```
set protocols mpls label-switched-path PE1-PE2 link-protection
set protocols mpls label-switched-path PE1-PE2 p2mp p2mp1
set protocols mpls label-switched-path PE1-PE3 to 100.70.70.70
set protocols mpls label-switched-path PE1-PE3 link-protection
set protocols mpls label-switched-path PE1-PE3 p2mp p2mp1
set protocols mpls label-switched-path PE1-PE4 to 100.40.40.40
set protocols mpls label-switched-path PE1-PE4 link-protection
set protocols mpls label-switched-path PE1-PE4 p2mp p2mp1
set protocols mpls interface fe-2/0/10.1
set protocols mpls interface fe-2/0/9.8
set protocols mpls interface fe-2/0/8.9
set protocols mpls interface lo0.1
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-2/0/2.0
set protocols ospf area 0.0.0.0 interface fe-2/0/10.1
set protocols ospf area 0.0.0.0 interface fe-2/0/9.8
set protocols ospf area 0.0.0.0 interface fe-2/0/8.9
set protocols ospf area 0.0.0.0 interface lo0.1
set routing-options static route 5.5.5.0/24 p2mp-lsp-next-hop p2mp1
set routing-options static route 7.7.7.0/24 p2mp-lsp-next-hop p2mp1
set routing-options static route 4.4.4.0/24 p2mp-lsp-next-hop p2mp1
set routing-options router-id 100.10.10.10
```

Device CE1

```
set interfaces ge-1/3/2 unit 0 family inet address 10.0.244.9/30
set interfaces ge-1/3/2 unit 0 description CE1-to-PE1
set routing-options static route 10.0.104.8/30 next-hop 10.0.244.10
set routing-options static route 10.0.134.8/30 next-hop 10.0.244.10
set routing-options static route 10.0.224.8/30 next-hop 10.0.244.10
```

Device CE2

```
set interfaces ge-1/3/3 unit 0 family inet address 10.0.224.9/30
set interfaces ge-1/3/3 unit 0 description CE2-to-PE2
set routing-options static route 10.0.244.8/30 next-hop 10.0.224.10
```

Device CE3

```
set interfaces ge-2/0/1 unit 0 family inet address 10.0.134.9/30
set interfaces ge-2/0/1 unit 0 description CE3-to-PE3
set routing-options static route 10.0.244.8/30 next-hop 10.0.134.10
```

Device CE4

```
set interfaces ge-3/1/3 unit 0 family inet address 10.0.104.10/30
set interfaces ge-3/1/3 unit 0 description CE4-to-PE4
set routing-options static route 10.0.244.8/30 next-hop 10.0.104.9
```

Configuring the Ingress Label-Switched Router (LSR) (Device PE1)

Step-by-Step Procedure

To configure Device PE1:

1. Configure the interfaces, interface encapsulation, and protocol families.

```
[edit interfaces]
user@PE1# set ge-2/0/2 unit 0 description PE1-to-CE1
user@PE1# set ge-2/0/2 unit 0 family inet address 10.0.244.10/30
user@PE1# set fe-2/0/10 unit 1 description PE1-to-P2
user@PE1# set fe-2/0/10 unit 1 family inet address 2.2.2.1/24
user@PE1# set fe-2/0/10 unit 1 family mpls
user@PE1# set fe-2/0/9 unit 8 description PE1-to-P3
user@PE1# set fe-2/0/9 unit 8 family inet address 6.6.6.1/24
user@PE1# set fe-2/0/9 unit 8 family mpls
user@PE1# set fe-2/0/8 unit 9 description PE1-to-P4
user@PE1# set fe-2/0/8 unit 9 family inet address 3.3.3.1/24
user@PE1# set fe-2/0/8 unit 9 family mpls
user@PE1# set lo0 unit 1 family inet address 100.10.10.10/32
```

2. Enable RSVP, MPLS, and OSPF on the interfaces.

```
[edit protocols]
user@PE1# set rsvp interface fe-2/0/10.1
user@PE1# set rsvp interface fe-2/0/9.8
```

```
user@PE1# set rsvp interface fe-2/0/8.9
user@PE1# set rsvp interface lo0.1
user@PE1# set mpls interface fe-2/0/10.1
user@PE1# set mpls interface fe-2/0/9.8
user@PE1# set mpls interface fe-2/0/8.9
user@PE1# set mpls interface lo0.1
user@PE1# set ospf area 0.0.0.0 interface ge-2/0/2.0
user@PE1# set ospf area 0.0.0.0 interface fe-2/0/10.1
user@PE1# set ospf area 0.0.0.0 interface fe-2/0/9.8
user@PE1# set ospf area 0.0.0.0 interface fe-2/0/8.9
user@PE1# set ospf area 0.0.0.0 interface lo0.1
```

3. Configure the MPLS point-to-multipoint LSPs.

```
[edit protocols]
user@PE1# set mpls label-switched-path PE1-PE2 to 100.50.50.50
user@PE1# set mpls label-switched-path PE1-PE2 p2mp p2mp1
user@PE1# set mpls label-switched-path PE1-PE3 to 100.70.70.70
user@PE1# set mpls label-switched-path PE1-PE3 p2mp p2mp1
user@PE1# set mpls label-switched-path PE1-PE4 to 100.40.40.40
user@PE1# set mpls label-switched-path PE1-PE4 p2mp p2mp1
```

4. (Optional) Enable link protection on the LSPs.

   Link protection helps to ensure that traffic sent over a specific interface to a neighboring router can continue to reach the router if that interface fails.

```
[edit protocols]
user@PE1# set mpls label-switched-path PE1-PE2 link-protection
user@PE1# set mpls label-switched-path PE1-PE3 link-protection
user@PE1# set mpls label-switched-path PE1-PE4 link-protection
```

5. Enable MPLS to perform traffic engineering for OSPF.

```
[edit protocols]
user@PE1# set mpls traffic-engineering bgp-igp
```

   This causes the ingress routes to be installed in the inet.0 routing table. By default, MPLS performs traffic engineering for BGP only. You need to enable MPLS traffic engineering on the ingress LSR only.

6. Enable traffic engineering for OSPF.

```
[edit protocols]
user@PE1# set ospf traffic-engineering
```

This causes the shortest-path first (SPF) algorithm to take into account the LSPs configured under MPLS.

7. Configure the router ID.

```
[edit routing-options]
user@PE1# set router-id 100.10.10.10
```

8. Configure static IP unicast routes with the point-to-multipoint LSP name as the next hop for each route.

```
[edit routing-options]
user@PE1# set static route 5.5.5.0/24p2mp-lsp-next-hop p2mp1
user@PE1# set static route 7.7.7.0/24 p2mp-lsp-next-hop p2mp1
user@PE1# set static route 4.4.4.0/24 p2mp-lsp-next-hop p2mp1
```

9. If you are done configuring the device, commit the configuration.

```
[edit]
user@PE1# commit
```

Configuring the Transit and Egress LSRs (Devices P2, P3, P4, PE2, PE3, and PE4)

Step-by-Step Procedure

To configure the transit and egress LSRs:

1. Configure the interfaces, interface encapsulation, and protocol families.

```
[edit]
user@P2# set interfaces fe-2/0/10 unit 2 description P2-to-PE1
user@P2# set interfaces fe-2/0/10 unit 2 family inet address 2.2.2.2/24
user@P2# set interfaces fe-2/0/10 unit 2 family mpls
```

```
user@P2# set interfaces fe-2/0/9 unit 10 description P2-to-PE2
user@P2# set interfaces fe-2/0/9 unit 10 family inet address 5.5.5.1/24
user@P2# set interfaces fe-2/0/9 unit 10 family mpls
user@P2# set interfaces lo0 unit 2 family inet address 100.20.20.20/32
user@PE2# set interfaces ge-2/0/3 unit 0 description PE2-to-CE2
user@PE2# set interfaces ge-2/0/3 unit 0 family inet address 10.0.224.10/30
user@PE2# set interfaces fe-2/0/10 unit 5 description PE2-to-P2
user@PE2# set interfaces fe-2/0/10 unit 5 family inet address 5.5.5.2/24
user@PE2# set interfaces fe-2/0/10 unit 5 family mpls
user@PE2# set interfaces lo0 unit 5 family inet address 100.50.50.50/32
user@P3# set interfaces fe-2/0/10 unit 6 description P3-to-PE1
user@P3# set interfaces fe-2/0/10 unit 6 family inet address 6.6.6.2/24
user@P3# set interfaces fe-2/0/10 unit 6 family mpls
user@P3# set interfaces fe-2/0/9 unit 11 description P3-to-PE3
user@P3# set interfaces fe-2/0/9 unit 11 family inet address 7.7.7.1/24
user@P3# set interfaces fe-2/0/9 unit 11 family mpls
user@P3# set interfaces lo0 unit 6 family inet address 100.60.60.60/32
user@PE3# set interfaces ge-2/0/1 unit 0 description PE3-to-CE3
user@PE3# set interfaces ge-2/0/1 unit 0 family inet address 10.0.134.10/30
user@PE3# set interfaces fe-2/0/10 unit 7 description PE3-to-P3
user@PE3# set interfaces fe-2/0/10 unit 7 family inet address 7.7.7.2/24
user@PE3# set interfaces fe-2/0/10 unit 7 family mpls
user@PE3# set interfaces lo0 unit 7 family inet address 100.70.70.70/32
user@P4# set interfaces fe-2/0/10 unit 3 description P4-to-PE1
user@P4# set interfaces fe-2/0/10 unit 3 family inet address 3.3.3.2/24
user@P4# set interfaces fe-2/0/10 unit 3 family mpls
user@P4# set interfaces fe-2/0/9 unit 12 description P4-to-PE4
user@P4# set interfaces fe-2/0/9 unit 12 family inet address 4.4.4.1/24
user@P4# set interfaces fe-2/0/9 unit 12 family mpls
user@P4# set interfaces lo0 unit 3 family inet address 100.30.30.30/32
user@PE4# set interfaces ge-2/0/0 unit 0 description PE4-to-CE4
user@PE4# set interfaces ge-2/0/0 unit 0 family inet address 10.0.104.9/30
user@PE4# set interfaces fe-2/0/10 unit 4 description PE4-to-P4
user@PE4# set interfaces fe-2/0/10 unit 4 family inet address 4.4.4.2/24
user@PE4# set interfaces fe-2/0/10 unit 4 family mpls
user@PE4# set interfaces lo0 unit 4 family inet address 100.40.40.40/32
```

2. Enable RSVP, MPLS, and OSPF on the interfaces.

```
[edit]
user@P2# set protocols rsvp interface fe-2/0/10.2
user@P2# set protocols rsvp interface fe-2/0/9.10
```

```
user@P2# set protocols rsvp interface lo0.2
user@P2# set protocols mpls interface fe-2/0/10.2
user@P2# set protocols mpls interface fe-2/0/9.10
user@P2# set protocols mpls interface lo0.2
user@P2# set protocols ospf area 0.0.0.0 interface fe-2/0/10.2
user@P2# set protocols ospf area 0.0.0.0 interface fe-2/0/9.10
user@P2# set protocols ospf area 0.0.0.0 interface lo0.2
user@PE2# set protocols rsvp interface fe-2/0/10.5
user@PE2# set protocols rsvp interface lo0.5
user@PE2# set protocols mpls interface fe-2/0/10.5
user@PE2# set protocols mpls interface lo0.5
user@PE2# set protocols ospf area 0.0.0.0 interface ge-2/0/3.0
user@PE2# set protocols ospf area 0.0.0.0 interface fe-2/0/10.5
user@PE2# set protocols ospf area 0.0.0.0 interface lo0.5
user@P3# set protocols rsvp interface fe-2/0/10.6
user@P3# set protocols rsvp interface fe-2/0/9.11
user@P3# set protocols rsvp interface lo0.6
user@P3# set protocols mpls interface fe-2/0/10.6
user@P3# set protocols mpls interface fe-2/0/9.11
user@P3# set protocols mpls interface lo0.6
user@P3# set protocols ospf area 0.0.0.0 interface fe-2/0/10.6
user@P3# set protocols ospf area 0.0.0.0 interface fe-2/0/9.11
user@P3# set protocols ospf area 0.0.0.0 interface lo0.6
user@PE3# set protocols rsvp interface fe-2/0/10.7
user@PE3# set protocols rsvp interface lo0.7
user@PE3# set protocols mpls interface fe-2/0/10.7
user@PE3# set protocols mpls interface lo0.7
user@PE3# set protocols ospf area 0.0.0.0 interface ge-2/0/1.0
user@PE3# set protocols ospf area 0.0.0.0 interface fe-2/0/10.7
user@PE3# set protocols ospf area 0.0.0.0 interface lo0.7
user@P4# set protocols rsvp interface fe-2/0/10.3
user@P4# set protocols rsvp interface fe-2/0/9.12
user@P4# set protocols rsvp interface lo0.3
user@P4# set protocols mpls interface fe-2/0/10.3
user@P4# set protocols mpls interface fe-2/0/9.12
user@P4# set protocols mpls interface lo0.3
user@P4# set protocols ospf area 0.0.0.0 interface fe-2/0/10.3
user@P4# set protocols ospf area 0.0.0.0 interface fe-2/0/9.12
user@P4# set protocols ospf area 0.0.0.0 interface lo0.3
user@PE4# set protocols rsvp interface fe-2/0/10.4
user@PE4# set protocols rsvp interface lo0.4
user@PE4# set protocols mpls interface fe-2/0/10.4
user@PE4# set protocols mpls interface lo0.4
```

```
user@PE4# set protocols ospf area 0.0.0.0 interface ge-2/0/0.0
user@PE4# set protocols ospf area 0.0.0.0 interface fe-2/0/10.4
user@PE4# set protocols ospf area 0.0.0.0 interface lo0.4
```

3. Enable traffic engineering for OSPF.

```
[edit]
user@P2# set protocols ospf traffic-engineering
user@P3# set protocols ospf traffic-engineering
user@P4# set protocols ospf traffic-engineering
user@PE2# set protocols ospf traffic-engineering
user@PE3# set protocols ospf traffic-engineering
user@PE4# set protocols ospf traffic-engineering
```

This causes the shortest-path first (SPF) algorithm to take into account the LSPs configured under MPLS.

4. Configure the router IDs.

```
[edit]
user@P2# set routing-options router-id 100.20.20.20
user@P3# set routing-options router-id 100.60.60.60
user@P4# set routing-options router-id 100.30.30.30
user@PE2# set routing-options router-id 100.50.50.50
user@PE3# set routing-options router-id 100.70.70.70
user@PE4# set routing-options router-id 100.40.40.40
```

5. If you are done configuring the devices, commit the configuration.

```
[edit]
user@host# commit
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols`, and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

Device PE1

```
user@PE1# show interfaces
ge-2/0/2 {
    unit 0 {
        description R1-to-CE1;
        family inet {
            address 10.0.244.10/30;
        }
    }
}
fe-2/0/10 {
    unit 1 {
        description PE1-to-P2;
        family inet {
            address 2.2.2.1/24;
        }
        family mpls;
    }
}
fe-2/0/9 {
    unit 8 {
        description PE1-to-P2;
        family inet {
            address 6.6.6.1/24;
        }
        family mpls;
    }
}
fe-2/0/8 {
    unit 9 {
        description PE1-to-P3;
        family inet {
            address 3.3.3.1/24;
        }
        family mpls;
    }
}
lo0 {
    unit 1 {
        family inet {
            address 100.10.10.10/32;
```

```
        }
    }
}
```

```
user@PE1# show protocols
rsvp {
    interface fe-2/0/10.1;
    interface fe-2/0/9.8;
    interface fe-2/0/8.9;
    interface lo0.1;
}
mpls {
    traffic-engineering bgp-igp;
    label-switched-path PE1-to-PE2 {
        to 100.50.50.50;
        link-protection;
        p2mp p2mp1;
    }
    label-switched-path PE1-to-PE3 {
        to 100.70.70.70;
        link-protection;
        p2mp p2mp1;
    }
    label-switched-path PE1-to-PE4 {
        to 100.40.40.40;
        link-protection;
        p2mp p2mp1;
    }
    interface fe-2/0/10.1;
    interface fe-2/0/9.8;
    interface fe-2/0/8.9;
    interface lo0.1;
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface ge-2/0/2.0;
        interface fe-2/0/10.1;
        interface fe-2/0/9.8;
        interface fe-2/0/8.9;
        interface lo0.1;
```

```
    }
}
```

```
user@PE1# show routing-options
static {
    route 5.5.5.0/24 {
        p2mp-lsp-next-hop p2mp1;
    }
    route 7.7.7.0/24 {
        p2mp-lsp-next-hop p2mp1;
    }
    route 4.4.4.0/24 {
        p2mp-lsp-next-hop p2mp1;
    }
}
router-id 100.10.10.10;
```

Device P2

```
user@P2# show interfaces
fe-2/0/10 {
    unit 2 {
        description P2-to-PE1;
        family inet {
            address 2.2.2.2/24;
        }
        family mpls;
    }
fe-2/0/9 {
    unit 10 {
        description P2-to-PE2;
        family inet {
            address 5.5.5.1/24;
        }
        family mpls;
    }
}
lo0 {
    unit 2 {
        family inet {
            address 100.20.20.20/32;
```

```
        }
    }
}
```

```
user@P2# show protocols
rsvp {
    interface fe-2/0/10.2;
    interface fe-2/0/9.10;
    interface lo0.2;
}
mpls {
    interface fe-2/0/10.2;
    interface fe-2/0/9.10;
    interface lo0.2;
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface fe-2/0/10.2;
        interface fe-2/0/9.10;
        interface lo0.2;
    }
}
```

```
user@P2# show routing-options
router-id 100.20.20.20;
```

Device P3

```
user@P3# show interfaces
fe-2/0/10 {
    unit 6 {
        description P3-to-PE1;
        family inet {
            address 6.6.6.2/24;
        }
        family mpls;
    }
}
```

```
fe-2/0/9 {
    unit 11 {
        description P3-to-PE3;
        family inet {
            address 7.7.7.1/24;
        }
        family mpls;
    }
}
lo0 {
    unit 6 {
        family inet {
            address 100.60.60.60/32;
        }
    }
}
```

```
user@P3# show protocols
rsvp {
    interface fe-2/0/10.6;
    interface fe-2/0/9.11;
    interface lo0.6;
}
mpls {
    interface fe-2/0/10.6;
    interface fe-2/0/9.11;
    interface lo0.6;
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface fe-2/0/10.6;
        interface fe-2/0/9.11;
        interface lo0.6;
    }
}
```

```
user@P2# show routing-options
router-id 100.60.60.60;
```

Device P4

```
user@P4# show interfaces
fe-2/0/10 {
    unit 3 {
        description P4-to-PE1;
        family inet {
            address 3.3.3.2/24;
        }
        family mpls;
    }
}
fe-2/0/9 {
    unit 12 {
        description P4-to-PE4;
        family inet {
            address 4.4.4.1/24;
        }
        family mpls;
    }
}
lo0 {
    unit 3 {
        family inet {
            address 100.30.30.30/32;
        }
    }
}
```

```
user@P4# show protocols
rsvp {
    interface fe-2/0/10.3;
    interface fe-2/0/9.12;
    interface lo0.3;
}
mpls {
    interface fe-2/0/10.3;
    interface fe-2/0/9.12;
    interface lo0.3;
}
ospf {
```

```
    traffic-engineering;
    area 0.0.0.0 {
        interface fe-2/0/10.3;
        interface fe-2/0/9.12;
        interface lo0.3;
    }
}
```

```
user@P3# show routing-options
router-id 100.30.30.30;
```

Device PE2

```
user@PE2# show interfaces
    ge-2/0/3 {
        unit 0 {
            description PE2-to-CE2;
            family inet {
                address 10.0.224.10/30;
            }
        }
    }
    fe-2/0/10 {
        unit 5 {
            description PE2-to-P2;
            family inet {
                address 5.5.5.2/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 5 {
            family inet {
                address 100.50.50.50/32;
            }
        }
```

```
    }
}
```

```
user@PE2# show protocols
rsvp {
    interface fe-2/0/10.5;
    interface lo0.5;
}
mpls {
    interface fe-2/0/10.5;
    interface lo0.5;
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface ge-2/0/3.0;
        interface fe-2/0/10.5;
        interface lo0.5;
    }
}
```

```
user@PE2# show routing-options
router-id 100.50.50.50;
```

Device PE3

```
user@PE3# show interfaces
    ge-2/0/1 {
        unit 0 {
            description PE3-to-CE3;
            family inet {
                address 10.0.134.10/30;
            }
        }
    }
    fe-2/0/10 {
        unit 7 {
            description PE3-to-P3;
            family inet {
```

```
                address 7.7.7.2/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 7 {
            family inet {
                address 100.70.70.70/32;
            }
        }
    }
}
```

```
user@PE3# show protocols
rsvp {
    interface fe-2/0/10.7;
    interface lo0.7;
}
mpls {
    interface fe-2/0/10.7;
    interface lo0.7;
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface ge-2/0/1.0;
        interface fe-2/0/10.7;
        interface lo0.7;
    }
}
```

```
user@PE3# show routing-options
router-id 100.70.70.70;
```

Device PE4

```
user@PE4# show interfaces
    ge-2/0/0 {
```

```
        unit 0 {
            description PE4-to-CE4;
            family inet {
                address 10.0.104.9/30;
            }
        }
    }
    fe-2/0/10 {
        unit 4 {
            description PE4-to-P4;
            family inet {
                address 4.4.4.2/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 4 {
            family inet {
                address 100.40.40.40/32;
            }
        }
    }
}
```

```
user@PE4# show protocols
rsvp {
    interface fe-2/0/10.4;
    interface lo0.4;
}
mpls {
    interface fe-2/0/10.4;
    interface lo0.4;
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface ge-2/0/0.0;
        interface fe-2/0/10.4;
        interface lo0.4;
```

```
    }
}
```

```
user@PE4# show routing-options
router-id 100.40.40.40;
```

Configuring Device CE1

Step-by-Step Procedure

To configure Device CE1:

1. Configure an interface to Device PE1.

```
[edit interfaces]
user@CE1# set ge-1/3/2 unit 0 family inet address 10.0.244.9/30
user@CE1# set ge-1/3/2 unit 0 description CE1-to-PE1
```

2. Configure static routes from Device CE1 to the three other customer networks, with Device PE1 as the next hop.

```
[edit routing-options]
user@CE1# set static route 10.0.104.8/30 next-hop 10.0.244.10
user@CE1# set static route 10.0.134.8/30 next-hop 10.0.244.10
user@CE1# set static route 10.0.224.8/30 next-hop 10.0.244.10
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@CE1# commit
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@CE1# show interfaces
ge-1/3/2 {
    unit 0 {
        family inet {
            address 10.0.244.9/30;
            description CE1-to-PE1;
        }
    }
}
```

```
user@CE1# show routing-options
static {
    route 10.0.104.8/30 next-hop 10.0.244.10;
    route 10.0.134.8/30 next-hop 10.0.244.10;
    route 10.0.224.8/30 next-hop 10.0.244.10;
}
```

Configuring Device CE2

Step-by-Step Procedure

To configure Device CE2:

1. Configure an interface to Device PE2.

```
[edit interfaces]
user@CE2# set ge-1/3/3 unit 0 family inet address 10.0.224.9/30
user@CE2# set ge-1/3/3 unit 0 description CE2-to-PE2
```

2. Configure a static route from Device CE2 to CE1, with Device PE2 as the next hop.

```
[edit routing-options]
user@CE2# set static route 10.0.244.8/30 next-hop 10.0.224.10
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@CE2# commit
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@CE2# show interfaces
ge-1/3/3 {
    unit 0 {
        family inet {
            address 10.0.224.9/30;
            description CE2-to-PE2;
        }
    }
}
```

```
user@CE2# show routing-options
static {
    route 10.0.244.8/30 next-hop 10.0.224.10;
}
```

Configuring Device CE3

Step-by-Step Procedure

To configure Device CE3:

1. Configure an interface to Device PE3.

```
[edit interfaces]
user@CE3# set ge-2/0/1 unit 0 family inet address 10.0.134.9/30
user@CE3# set ge-2/0/1 unit 0 description CE3-to-PE3
```

2. Configure a static route from Device CE3 to CE1, with Device PE3 as the next hop.

```
[edit routing-options]
user@CE3# set static route 10.0.244.8/30 next-hop 10.0.134.10
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@CE3# commit
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@CE3# show interfaces
ge-2/0/1 {
    unit 0 {
        family inet {
            address 10.0.134.9/30;
            description CE3-to-PE3;
        }
    }
}
```

```
user@CE3# show routing-options
static {
    route 10.0.244.8/30 next-hop 10.0.134.10;
}
```

Configuring Device CE4

Step-by-Step Procedure

To configure Device CE4:

1. Configure an interface to Device PE4.

```
[edit interfaces]
user@CE4# set ge-3/1/3 unit 0 family inet address 10.0.104.10/30
user@CE4# set ge-3/1/3 unit 0 description CE4-to-PE4
```

2. Configure a static route from Device CE4 to CE1, with Device PE4 as the next hop.

```
[edit routing-options]
user@CE4# set static route 10.0.244.8/30 next-hop 10.0.104.9
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@CE4# commit
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@CE4# show interfaces
ge-3/1/3 {
    unit 0 {
        family inet {
            address 10.0.104.10/30;
            description CE4-to-PE4;
        }
```

```
    }
}
```

```
user@CE4# show routing-options
static {
    route 10.0.244.8/30 next-hop 10.0.104.9;
}
```

Verification

IN THIS SECTION

- Verifying Connectivity | 809
- Verifying the State of the Point-to-Multipoint LSP | 810
- Checking the Forwarding Table | 811

Confirm that the configuration is working properly.

Verifying Connectivity

Purpose

Make sure that the devices can ping each other.

Action

Run the `ping` command from CE1 to the interface on CE2 connecting to PE2.

```
user@CE1> ping 10.0.224.9
PING 10.0.224.9 (10.0.224.9): 56 data bytes
64 bytes from 10.0.224.9: icmp_seq=0 ttl=61 time=1.387 ms
64 bytes from 10.0.224.9: icmp_seq=1 ttl=61 time=1.394 ms
64 bytes from 10.0.224.9: icmp_seq=2 ttl=61 time=1.506 ms
^C
--- 10.0.224.9 ping statistics ---
```

```
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.387/1.429/1.506/0.055 ms
```

Run the `ping` command from CE1 to the interface on CE3 connecting to PE3.

```
user@CE1> ping 10.0.134.9
PING 10.0.134.9 (10.0.134.9): 56 data bytes
64 bytes from 10.0.134.9: icmp_seq=0 ttl=61 time=1.068 ms
64 bytes from 10.0.134.9: icmp_seq=1 ttl=61 time=1.062 ms
64 bytes from 10.0.134.9: icmp_seq=2 ttl=61 time=1.053 ms
^C
--- 10.0.134.9 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.053/1.061/1.068/0.006 ms
```

Run the `ping` command from CE1 to the interface on CE4 connecting to PE4.

```
user@CE1> ping 10.0.104.10
PING 10.0.104.10 (10.0.104.10): 56 data bytes
64 bytes from 10.0.104.10: icmp_seq=0 ttl=61 time=1.079 ms
64 bytes from 10.0.104.10: icmp_seq=1 ttl=61 time=1.048 ms
64 bytes from 10.0.104.10: icmp_seq=2 ttl=61 time=1.070 ms
^C
--- 10.0.104.10 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.048/1.066/1.079/0.013 ms
```

Verifying the State of the Point-to-Multipoint LSP

Purpose

Make sure that the ingress, transit, and egress LSRs are in the Up state.

Action

Run the `show mpls lsp p2mp` command on all of the LSRs. Only the ingress LSR is shown here.

```
user@PE1> show mpls lsp p2mp
Ingress LSP: 1 sessions
```

```
P2MP name: p2mp1, P2MP branch count: 3
To              From            State Rt P     ActivePath       LSPname
100.40.40.40    100.10.10.10    Up     0 *                      PE1-PE4
100.70.70.70    100.10.10.10    Up     0 *                      PE1-PE3
100.50.50.50    100.10.10.10    Up     0 *                      PE1-PE2
Total 3 displayed, Up 3, Down 0
...
```

Checking the Forwarding Table

Purpose

Make sure that the routes are set up as expected by running the `show route forwarding-table` command. Only the routes to the remote customer networks are shown here.

Action

```
user@PE1> show route forwarding-table
Routing table: default.inet
Internet:
Destination        Type RtRef Next hop           Type Index NhRef Netif
...
10.0.104.8/30      user     0 3.3.3.2            ucst  1006     6 fe-2/0/8.9
10.0.134.8/30      user     0 6.6.6.2            ucst  1010     6 fe-2/0/9.8
10.0.224.8/30      user     0 2.2.2.2            ucst  1008     6 fe-2/0/10.1
...
```

Configuring Primary and Branch LSPs for Point-to-Multipoint LSPs

IN THIS SECTION

- Configuring the Primary Point-to-Multipoint LSP | **812**
- Configuring a Branch LSP for Point-to-Multipoint LSPs | **812**

A point-to-multipoint MPLS label-switched path (LSP) is an RSVP LSP with multiple destinations. By taking advantage of the MPLS packet replication capability of the network, point-to-multipoint LSPs

avoid unnecessary packet replication at the ingress router. For more information about point-to-multipoint LSPs, see "Point-to-Multipoint LSPs Overview" on page 781.

To configure a point-to-multipoint LSP, you need to configure the primary LSP from the ingress router and the branch LSPs that carry traffic to the egress routers, as described in the following sections:

Configuring the Primary Point-to-Multipoint LSP

A point-to-multipoint LSP must have a configured primary point-to-multipoint LSP to carry traffic from the ingress router. The configuration of the primary point-to-multipoint LSP is similar to a signaled LSP. See "Configuring the Ingress Router for MPLS-Signaled LSPs" on page 594 for more information. In addition to the conventional LSP configuration, you need to specify a path name for the primary point-to-multipoint LSP by including the `p2mp` statement:

```
p2mp p2mp-lsp-name;
```

You can include this statement at the following hierarchy levels:

- `[edit protocols mpls label-switched-path lsp-name]`
- `[edit logical-systems logical-system-name protocols mpls label-switched-path lsp-name]`

You can enable the optimization timer for point-to-multipoint LSPs. See "Optimizing Signaled LSPs" on page 622 for more information.

Configuring a Branch LSP for Point-to-Multipoint LSPs

The primary point-to-multipoint LSP sends traffic to two or more branch LSPs carrying traffic to each of the egress provider edge (PE) routers. In the configuration for each of these branch LSPs, the point-to-multipoint LSP path name you specify must be identical to the path name configured for the primary point-to-multipoint LSP. See "Configuring the Primary Point-to-Multipoint LSP" on page 812 for more information.

To associate a branch LSP with the primary point-to-multipoint LSP, specify the point-to-multipoint LSP name by including the `p2mp` statement:

```
p2mp p2mp-lsp-name;
```

You can include this statement at the following hierarchy levels:

- `[edit protocols mpls label-switched-path lsp-name]`
- `[edit logical-systems logical-system-name protocols mpls label-switched-path lsp-name]`

NOTE: Any change in any of the branch LSPs of a point-to-multipoint LSP, either due to a user action or an automatic adjustment made by the router, causes the primary and branch LSPs to be resignaled. The new point-to-multipoint LSP is signaled first before the old path is taken down.

The following sections describe how you can configure the branch LSP as a dynamically signaled path using Constrained Shortest Path First (CSPF), as a static path, or as a combination of dynamic and static paths:

Configuring the Branch LSP as a Dynamic Path

By default, the branch LSP for a point-to-multipoint LSP is signaled dynamically using CSPF and requires no configuration.

When a point-to-multipoint LSP is changed, either by the addition or deletion of new destinations or by the recalculation of the path to existing destinations, certain nodes in the tree might receive data from more than one incoming interface. This can happen under the following conditions:

- Some of the branch LSPs to destinations are statically configured and might intersect with statically or dynamically calculated paths to other destinations.
- When a dynamically calculated path for a branch LSP results in a change of incoming interface for one of the nodes in the network, the older path is not immediately torn down after the new one has been signaled. This ensures that any data in transit relying on the older path can reach its destination. However, network traffic can potentially use either path to reach the destination.
- A faulty router at the ingress calculates the paths to two different branch destinations such that a different incoming interface is chosen for these branch LSPs on a router node common to these branch LSPs.

Configuring the Branch LSP as a Static Path

You can configure the branch LSP for a point-to-multipoint LSP as a static path. See "Configuring Static LSPs" on page 690 for more information.

Configuring Inter-Domain Point-to-Multipoint LSPs

An inter-domain P2MP LSP is a P2MP LSP that has one or more sub-LSPs (branches) that span multiple domains in a network. Examples of such domains include IGP areas and autonomous systems (ASs). A sub-LSP of an inter-domain P2MP LSP may be intra-area, inter-area, or inter-AS, depending on the location of the egress node (leaf) with respect to the ingress node (source).

On the ingress node, a name is assigned to the inter-domain P2MP LSP and shared by all constituent sub-LSPs. Each sub-LSP is configured separately, with its own egress node and optionally an explicit path. The location of the egress node of the sub-LSP with respect to the ingress node determines whether the sub-LSP is intra-area, inter-area, or inter-AS.

Inter-domain P2MP LSPs can be used to transport traffic in the following applications in a multi-area or multi-AS network:

- Layer 2 broadcast and multicast over MPLS
- Layer 3 BGP/MPLS VPN
- VPLS

On each domain boundary node (ABR or ASBR) along the path of the P2MP LSP, the `expand-loose-hop` statement must be configured at the `[edit protocols mpls]` hierarchy level so that CSPF can extend a loose-hop ERO (usually the first entry of the ERO list carried by RSVP Path message) towards the egress node or the next domain boundary node.

CSPF path computation for inter-domain P2MP LSPs:

- CSPF path computation is supported on each sub-LSP for inter-domain P2MP LSPs. A sub-LSP may be intra-area, inter-area, or inter-AS. CSPF treats an inter-area or inter-AS sub-LSP in the same manner as an inter-domain P2P LSP.
- On an ingress node or a domain boundary node (ABR or ASBR), CSPF can perform an Explicit Route Object (ERO) expansion per-RSVP query. The destination queried could be an egress node or a received loose-hop ERO. If the destination resides in a neighboring domain that the node is connected to, CSPF generates either a sequence of strict-hop EROs towards it or a sequence of strict-hop EROs towards another domain boundary node that can reach the destination.
- If RSVP fails to signal a path through a previously selected domain bounday node, RSVP attempts to signal a path through other available domain boundary nodes in a round-robin fashion.
- When a sub-LSP is added or removed to or from an inter-domain P2MP LSP, causing its path (branch) to be merged or pruned with or from the current P2MP tree, the paths being taken by the other sub-LSPs should not be affected, helping to prevent traffic disruption on those sub-LSPs.

Be aware of the following when deploying inter-domain P2MP LSPs in your network:

- Periodic path re-optimization is supported for inter-domain P2MP LSPs on ingress nodes. It can be turned on for an inter-domain P2MP LSP by configuring the `optimize-timer` statement at the `[edit protocols mpls label-switched-path` *`lsp-name`*`]` hierarchy level with the same interval for every sub-LSP.
- Only link protection bypass LSPs are supported for inter-domain P2MP LSPs. To enable it for an inter-domain P2MP LSP, link-protection must be configured for all sub-LSPs and on all of the RSVP interfaces that the P2MP LSP might travel through.

- Only OSPF areas are supported for inter-domain P2MP LSPs. IS-IS levels are not supported.

Configuring Link Protection for Point-to-Multipoint LSPs

Link protection helps to ensure that traffic going over a specific interface to a neighboring router can continue to reach this router if that interface fails. When link protection is configured for an interface and a point-to-multipoint LSP that traverses this interface, a bypass LSP is created that handles this traffic if the interface fails. The bypass LSP uses a different interface and path to reach the same destination.

To extend link protection to all of the paths used by a point-to-multipoint LSP, link protection must be configured on each router that each branch LSP traverses. If you enable link protection on a point-to-multipoint LSP, you must enable link protection on all of the branch LSPs.

The Internet draft draft-ietf-mpls-rsvp-te-p2mp-01.txt, *Extensions to RSVP-TE for Point to Multipoint TE LSPs*, describes link protection for point-to-multipoint LSPs.

To enable link protection on point-to-multipoint LSPs, complete the following steps:

1. Configure link protection on each branch LSP. To configure link protection, include the `link-protection` statement:

   ```
   link-protection;
   ```

   You can include this statement at the following hierarchy levels:

   - `[edit protocols mpls label-switched-path` *`branch-lsp-name`*`]`
   - `[edit logical-systems` *`logical-system-name`* `protocols mpls label-switched-path` *`branch-lsp-name`*`]`

2. Configure link protection for each RSVP interface on each router that the branch LSP traverses. For information about how to configure link protection on RSVP interfaces, see "Configuring Link Protection on Interfaces Used by LSPs" on page 468.

For more information on how to configure link protection, see "Configuring Node Protection or Link Protection for LSPs" on page 478.

Configuring Graceful Restart for Point-to-Multipoint LSPs

You can configure graceful restart on point-to-multipoint LSPs. Graceful restart allows a router undergoing a restart to inform its adjacent neighbors of its condition. The restarting router requests a grace period from the neighbor or peer, which can then cooperate with the restarting router. The restarting router can still forward MPLS traffic during the restart period; convergence in the network is not disrupted. The restart is not apparent to the rest of the network, and the restarting router is not removed from the network topology. RSVP graceful restart can be enabled on both transit routers and ingress routers.

To enable graceful restart on a router handling point-to-multipoint LSP traffic, include the `graceful-restart` statement:

```
graceful-restart;
```

You can include this statement at the following hierarchy levels:

- `[edit routing-options]`
- `[edit logical-systems` *`logical-system-name`* `routing-options]`

The graceful restart configuration for point-to-multipoint LSPs is identical to that of point-to-point LSPs. For more information on how to configure graceful restart, see "Configuring RSVP Graceful Restart" on page 1282.

Configuring a Multicast RPF Check Policy for Point-to-Multipoint LSPs

IN THIS SECTION

- Example: Configuring Multicast RPF Check Policy for a Point-to-Multipoint LSP | **817**

You can control whether a reverse path forwarding (RPF) check is performed for a source and group entry before installing a route in the multicast forwarding cache. This makes it possible to use point-to-multipoint LSPs to distribute multicast traffic to PIM islands situated downstream from the egress routers of the point-to-multipoint LSPs.

By configuring the `rpf-check-policy` statement, you can disable RPF checks for a source and group pair. You would typically configure this statement on the egress routers of a point-to-multipoint LSP, because the interface receiving the multicast traffic on a point-to-multipoint LSP egress router might not always be the RPF interface.

You can also configure a routing policy to act upon a source and group pair. This policy behaves like an import policy, so if no policy term matches the input data, the default policy action is “acceptance.” An accept policy action enables RPF checks. A reject policy action (applied to all source and group pairs that are not accepted) disables RPF checks for the pair.

To configure a multicast RPF check policy for a point-to-multipoint LSP, specify the RPF check policy using the `rpf-check-policy` statement:

```
rpf-check-policy policy;
```

You can include this statement at the following hierarchy levels:

- `[edit routing-options multicast]`
- `[edit logical-systems logical-system-name routing-options multicast]`

You also must configure a policy for the multicast RPF check. You configure policies at the `[edit policy-options]` hierarchy level. For more information, see the Routing Policies, Firewall Filters, and Traffic Policers User Guide.

NOTE: When you configure the `rpf-check-policy` statement, the Junos OS cannot perform RPF checks on incoming traffic and therefore cannot detect traffic arriving on the wrong interface. This might cause routing loops to form.

Example: Configuring Multicast RPF Check Policy for a Point-to-Multipoint LSP

Configure a policy to ensure that an RPF check is not performed for sources with prefix `128.83/16` or longer that belong to groups having a prefix of `228/8` or longer:

```
[edit]
policy-options {
    policy-statement rpf-sg-policy {
        from {
            route-filter 228.0.0.0/8 orlonger;
            source-address-filter 128.83.0.0/16 orlonger;
        }
        then {
            reject;
        }
    }
}
```

Configuring Ingress PE Router Redundancy for Point-to-Multipoint LSPs

You can configure one or more PE routers as part of a backup PE router group to enable ingress PE router redundancy. You accomplish this by configuring the IP addresses of the backup PE routers (at least one backup PE router is required) and the local IP address used by the local PE router.

You must also configure a full mesh of point-to-point LSPs between the primary and backup PE routers. You also need to configure BFD on these LSPs. See "Configuring BFD for RSVP-Signaled LSPs" on page 213 and "Configuring BFD for LDP LSPs" on page 1408 for more information.

To configure ingress PE router redundancy for point-to-multipoint LSPs, include the `backup-pe-group` statement:

```
backup-pe-group pe-group-name {
    backups [addresses];
    local-address address;
}
```

For a list of hierarchy levels at which you can include these statements, see the statement summary sections for these statements.

After you configure the ingress PE router redundancy backup group, you must also apply the group to a static route on the PE router. This ensures that the static route is active (installed in the forwarding table) when the local PE router is the designated forwarder for the backup PE group. You can only associate a backup PE router group with a static route that also has the `p2mp-lsp-next-hop` statement configured. For more information, see "Configuring Static Unicast Routes for Point-to-Multipoint LSPs" on page 690.

Configuring a Service to Correlate Point-to-Multipoint sub-LSPs with FPCs

In addition to acting as the ingress or egress for a given sub-LSP, the Packet Forwarding Engine on an FPC also serves as a transit point for other sub-LSPs of the same point-to-multipoint LSP. If an FPC fails, then all the sub-LSPs that it serves are affected.

You can configure a service that enables you to monitor the correlation between FPCs and the point-to-multipoint sub-LSPs—branch paths—that are on an LSR. This information helps you evaluate the effect a failed FPC has on the correlated sub-LSPs. When tracing is enabled, the service also provides syslog messages in the event of an FPC outage that provide detailed information about the sub-LSPs affected.

You can configure a service that enables you to monitor the correlation between FPCs and the point-to-multipoint sub-LSPs—branch paths—on an LSR. An FPC can act as an ingress, egress, or transit point for more than one sub-LSP of the same point-to-multipoint LSP. If an FPC fails, then all the sub-LSPs that it serves are affected.

The information provided by this service helps you evaluate the effect a failure in any FPC has on the correlated sub-LSPs and the point-to-multipoint network. You can use this knowledge to help plan controlled FPC outages.

You can also enable tracing of some or all service operations. The service then provides syslog messages with detailed information about the affected sub-LSPs that facilitates analysis of an FPC outage.

To enable the monitoring and correlation of sub-LSPs and FPCs in the point-to-multipoint network:

1. Configure the point-to-multipoint polling (`p2mp_polling_duration`) and FPC polling (`fpc_polling_duration`) by setting the frequency duration (in seconds) in the config.xml file located at the **/etc/**

p2mp_lsp_correlation directory. You can also enable log levels in the config.xml file to configure traceoptions and the logs are created at the **/var/log/p2mp_lsp_correlation** directory. The log level and message types are as follows:

```
5 = DEBUG
4 = INFO
3 = WARNING
2 = ERROR
1 = CRITICAL
```

The following is a sample config.xml file:

```
user@host:~# cat /etc/p2mp_lsp_correlation/config.xml
 <p2mp_sub_lsp_config>
      <p2mp_polling_duration>240</p2mp_polling_duration>
      <fpc_polling_duration>60</fpc_polling_duration>
      <log_level>5</log_level>
</p2mp_sub_lsp_config>
```

- `p2mp_polling_duration`–Refreshes the database by executing various RE/PFE RPC requests. The default value for point-to-multipoint polling duration is 240.
- `fpc_polling_duration`–Polls for status of the FPC/PFE to log the impact of point-to-multipoint sub-LSPs. The default for FPC polling duration is 60.

NOTE: The config.xml file is applicable only for Junos OS Evolved. You need to restart the application after making changes to the config.xml file.

2. Enable the service.

```
[edit services]
user@host# set p2mp-sublsp-correlation
```

3. Configure tracing of the service operations.

```
[edit services]
user@host# set p2mp-sublsp-correlation traceoptions flag all
```

NOTE: The `set p2mp-sublsp-correlation traceoptions flag all` command is not applicable for Junos OS Evolved.

When an FPC on an LSR fails or goes offline, all point-to-multipoint sub-LSPs on that FPC are affected. If you have previously enabled FPC correlation for the point-to-multipoint LSPs, and configured tracing for the correlation service, then upon FPC failure messages are logged that provide details about the affected sub-LSPs.

In this case, you need to examine the system log messages and the FPC correlation table to analyze the impact of an FPC failure.

The following is a sample system log output showing information about the point-to-multipoint sub-LSP when the impacted FPC goes offline:

```
Aug  5 12:47:33  host mdiag[24321]: MDIAGD_P2MP_SUBLSP_IMPACTED: FPC 0
    PFEInst 0 Role (I,E,T) DOWN P2MP-Tunnel-Name p2mp-2-456 Sub-LSP-Dest 4.4.4.4 Sub-LSP-
    Name lsp-2-4 Tunnel-ID 53322 LSP-ID 1 Src-Addr 2.2.2.2 Sub-Group-ID 10 Ingress-
    Interface ae8.0 Egress-Interface et-0/0/7.0
```

To view the point-to-multipoint sub-LSP correlation information for ingress interface, use the `show services p2mp-sublsp-correlation ingress-interface` command as follows:

```
user@host> show services p2mp-sublsp-correlation ingress-interface ae8.0
Last Refreshed : Aug 05 2021 12:06:50

SG-ID = Sub-Group-ID, Tun-ID =  Tunnel-ID
FPC ROLE: I = Ingress, E = Egress, T = Transit

P2MP  Sub-LSP  Sub-LSP  Tun   LSP  Source   SG  Ingress  Egress
Name   Dest    Name     ID     ID  Address  ID  Interface Interface

bud-p-68 8.8.8.8  bud-8  53323  1  2.2.2.2  18  ae8.0     et-0/0/5.0
bud-p-68 6.6.6.6  bud-6  53323  1  2.2.2.2  12  ae8.0     et-0/0/9.0
bud-p-68 7.7.7.7  bud-7  53323  1  2.2.2.2  17  ae8.0     et-0/0/7.0
p2mp-2-6 4.4.4.4  lsp-4  53322  1  2.2.2.2  10  ae8.0     et-0/0/7.0
p2mp-2-6 5.5.5.5  lsp-5  53322  1  2.2.2.2  15  ae8.0     et-0/0/5.0
p2mp-2-6 6.6.6.6  lsp-6  53322  1  2.2.2.2  12  ae8.0     et-0/0/9.0
```

To view the point-to-multipoint sub-LSP correlation information for egress interface, use the `show services p2mp-sublsp-correlation egress-interface` command as follows:

```
user@host> show services p2mp-sublsp-correlation egress-interface et-0/0/7.0
Last Refreshed : Aug 05 2021 12:06:50

SG-ID = Sub-Group-ID, Tun-ID =  Tunnel-ID
FPC ROLE: I = Ingress, E = Egress, T = Transit

P2MP  Sub-LSP  Sub-LSP  Tun   LSP  Source   SG  Ingress  Egress
Name   Dest    Name     ID     ID  Address  ID  Interface Interface

bud-p-68 7.7.7.7  bud-7  53323  1  2.2.2.2  17  ae8.0     et-0/0/7.0
p2mp-2-6 4.4.4.4  lsp-4  53322  1  2.2.2.2  10  ae8.0     et-0/0/7.0
```

To view the correlation information for FPC, use the `show services p2mp-sublsp-correlation fpc 0` command as follows:

```
user@host> show services p2mp-sublsp-correlation fpc 0
Last Refreshed : Aug 05 2021 12:06:50

SG-ID = Sub-Group-ID, Tun-ID =  Tunnel-ID
FPC ROLE: I = Ingress, E = Egress, T = Transit

P2MP  Sub-LSP  Sub-LSP  Tun   LSP  Source   SG  Ingress  Egress    FPC/PFE
Name   Dest    Name     ID     ID  Address  ID  Interface Interface   Role

bud-p-68 8.8.8.8  bud-8  53323  1  2.2.2.2  18  ae8.0     et-0/0/5.0  I,E,
bud-p-68 6.6.6.6  bud-6  53323  1  2.2.2.2  12  ae8.0     et-0/0/9.0  I,E,T
bud-p-68 7.7.7.7  bud-7  53323  1  2.2.2.2  17  ae8.0     et-0/0/7.0  I,E,
p2mp-2-6 4.4.4.4  lsp-4  53322  1  2.2.2.2  10  ae8.0     et-0/0/7.0  I,E,T
p2mp-2-6 5.5.5.5  lsp-5  53322  1  2.2.2.2  15  ae8.0     et-0/0/5.0  I,E,T
p2mp-2-6 6.6.6.6  lsp-6  53322  1  2.2.2.2  12  ae8.0     et-0/0/9.0  I,E,
```

To view the correlation information for PFE instance, use the `show services p2mp-sublsp-correlation fpc 0 pfe-instance 0` command as follows:

```
user@host> show services p2mp-sublsp-correlation fpc 0 pfe-instance 0
Last Refreshed : Aug 05 2021 12:06:50
```

```
SG-ID = Sub-Group-ID, Tun-ID =  Tunnel-ID
FPC ROLE: I = Ingress, E = Egress, T = Transit

P2MP  Sub-LSP  Sub-LSP  Tun   LSP  Source   SG  Ingress  Egress    FPC/PFE
Name   Dest    Name     ID     ID  Address  ID  Interface Interface   Role

bud-p-68 8.8.8.8  bud-8  53323  1  2.2.2.2  18  ae8.0     et-0/0/5.0  I,E,
bud-p-68 6.6.6.6  bud-6  53323  1  2.2.2.2  12  ae8.0     et-0/0/9.0  I,E,T
bud-p-68 7.7.7.7  bud-7  53323  1  2.2.2.2  17  ae8.0     et-0/0/7.0  I,E,
p2mp-2-6 4.4.4.4  lsp-4  53322  1  2.2.2.2  10  ae8.0     et-0/0/7.0  I,E,T
p2mp-2-6 5.5.5.5  lsp-5  53322  1  2.2.2.2  15  ae8.0     et-0/0/5.0  I,E,T
p2mp-2-6 6.6.6.6  lsp-6  53322  1  2.2.2.2  12  ae8.0     et-0/0/9.0  I,E,
```

Enabling Point-to-Point LSPs to Monitor Egress PE Routers

Configuring an LSP with the `associate-backup-pe-groups` statement enables it to monitor the status of the PE router to which it is configured. You can configure multiple backup PE router groups using the same router's address. A failure of this LSP indicates to all of the backup PE router groups that the destination PE router is down. The `associate-backup-pe-groups` statement is not tied to a specific backup PE router group. It applies to all groups that are interested in the status of the LSP to that address.

To allow an LSP to monitor the status of the egress PE router, include the `associate-backup-pe-groups` statement:

```
associate-backup-pe-groups;
```

This statement can be configured at the following hierarchy levels:

- `[edit protocols mpls label-switched-path` *`lsp-name`*`]`
- `[edit logical-systems` *`logical-system-name`* `protocols mpls label-switched-path` *`lsp-name`*`]`

If you configure the `associate-backup-pe-groups` statement, you must configure BFD for the point-to-point LSP. For information about how to configure BFD for an LSP, see "Configuring BFD for MPLS IPv4 LSPs" on page 213 and "Configuring BFD for LDP LSPs" on page 1408.

You also must configure a full mesh of point-to-point LSPs between the PE routers in the backup PE router group. A full mesh is required so that each PE router within the group can independently determine the status of the other PE routers, allowing each router to independently determine which PE router is currently the designated forwarder for the backup PE router group.

If you configure multiple LSPs with the `associate-backup-pe-groups` statement to the same destination PE router, the first LSP configured is used to monitor the forwarding state to that PE router. If you configure multiple LSPs to the same destination, make sure to configure similar parameters for the LSPs. With this

configuration scenario, a failure notification might be triggered even though the remote PE router is still up.

Preserving Point-to-Multipoint LSP Functioning with Different Junos OS Releases

In Junos OS Release 9.1 and earlier, Resv messages that include the S2L_SUB_LSP object are rejected by default. In Junos OS Release 9.2 and later, such messages are accepted by default. To ensure proper functioning of point-to-multipoint LSPs in a network that includes both devices running Junos OS Release 9.1 and earlier and devices running Junos 9.2 and later, you must include the `no-p2mp-sublsp` statement in the configuration of the devices running Junos 9.2 and later:

```
no-p2mp-sublsp;
```

You can include this statement at the following hierarchy levels:

- `[edit protocols rsvp]`
- `[edit logical-systems` *`logical-system-name`* `protocols rsvp]`

Re-merge Behavior on Point-to-Multipoint LSP Overview

IN THIS SECTION

- Benefits of Controlling the P2MP LSP Re-merge | **823**
- What is P2MP LSP Re-merge? | **824**
- Modify the Default P2MP LSP Re-merge Behavior | **825**

This section talks about the benefits and overview of controlling the re-merge behavior on RSVP Point-to-Multipoint (P2MP) LSPs.

Benefits of Controlling the P2MP LSP Re-merge

- Reduces the RSVP signalling load on the ingress (headend routers) by avoiding path computation of sub LSPs which creates a re-merge condition.
- Saves the network bandwidth by rejecting the P2MP sub LSP re-merge at the transit node.

What is P2MP LSP Re-merge?

In a P2MP MPLS LSP network, the term re-merge refers to the case of an ingress (headend) or transit node (re-merge node) that creates a re-merge branch intersecting the P2MP LSP at another node down the tree. This may occur due to events such as an error in path calculation, an error in manual configuration, or network topology changes during the establishment of the P2MP LSP.

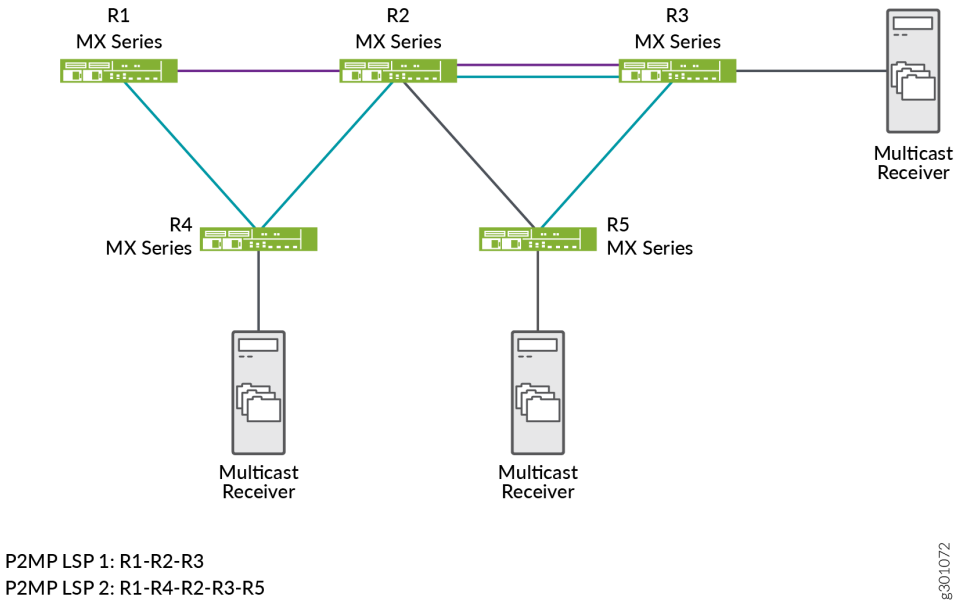
RFC 4875 defines the following two approaches for handling the P2MP LSP re-merge:

- First, the node detecting the re-merge allows the re-merge case to persist, but data from all but one incoming interface is dropped at the re-merge node. This works by default without any configuration.
- Second, the re-merge node initiates the pruning of the re-merge sub LSPs through signaling.

On Juniper Networks MX Series routers, the first approach (as defined by RFC 4875) works by default. The second approach can be implemented by one of the following CLI configuration statements depending upon where the Juniper Networks MX Series routers are placed (ingress node or transit node) in the P2MP RSVP MPLS network:

- `no-re-merge`—This CLI configuration statement when enabled at the ingress (headend) router avoids the path computation of P2MP sub LSPs which creates a re-merge condition. When this CLI configuration statement is configured at the ingress, then configuring the `no-p2mp-re-merge` CLI configuration statement at the transit router is not required.
- `no-p2mp-re-merge`—This CLI configuration statement when enabled at the transit router changes the default behavior of allowing the P2MP sub LSP sessions re-merge to rejecting the re-merge. This CLI configuration statement is primarily required when the ingress (headend router) is not a Juniper Networks MX Series router.
- `single-abr`—This command when enabled reduces re-merge condition beyond the inter-area, or inter-domain, or inter-AS RSVP P2MP LSPs.

The following topology explains the re-merge behavior in a P2MP LSP network:



In this topology, R1 acts as an ingress (headend) router and R2 as the transit (re-merge node) router. There are two sub LSP sessions created in this network, LSP 1 and LSP 2. LSP 1 is a session established among R1, R2, and R3 devices. LSP 2 is a session established between R1, R4, R2, R3, and R5 devices. By default, the transit router allows the re-merge to happen from both the sub LSPs and drops one of the sub LSP branch traffic at the re-merge node. You can control this re-merge behavior by enabling the `no-re-merge` CLI configuration statement at the ingress router, or the `no-p2mp-re-merge` CLI configuration statement at the transit router.

If you enable the `no-re-merge` CLI configuration statement at the ingress router (R1), only one of the two sub LSP sessions is established. For example, if LSP 1 (R1-R2-R3) session is established first, then the other sub LSP session (LSP 2) will not be established.

If you enable the `no-p2mp-re-merge` CLI configuration statement at the transit router (R2), the transit router rejects the re-merge of one of the sub LSPs and sends a path error message to the ingress router (R1) preventing the ingress router from creating the second P2MP LSP re-merge branch. You can use the `show rsvp statistics` CLI command to view the path error message.

Modify the Default P2MP LSP Re-merge Behavior

You can modify the default re-merge behavior either at the ingress (headend) node, or at the transit node in a P2MP RSVP MPLS network.

On the ingress (headend router), disable the default re-merge behavior so that the ingress router does not do the path computation of sub LSPs which creates the re-merge condition. The default behavior allows the path computation of sub LSPs.

```
[edit protocols]
user@host#set mpls p2mp-lsp no-re-merge
```

On the transit router, disable the default re-merge behavior so that the transit router rejects the re-merge of sub LSPs.

```
[edit protocols]
user@host#set rsvp no-p2mp-re-merge
```

For inter-area, or inter-domain, or inter-AS RSVP P2MP LSPs, use the `single-abr` CLI configuration statement at the ingress (headend router) so that all the P2MP sub LSPs prefer to select the same exit router (ABR or ASBR), thereby reducing the re-merge condition.

```
[edit protocols]
user@host#set mpls p2mp-lsp single-abr
```

RELATED DOCUMENTATION

Basic MPLS Configuration | 46

Pop-and-Forward LSP Configuration

IN THIS SECTION

- Benefits of RSVP-TE Pop-and-Forward LSP Tunnels | 827
- Pop-and-Forward LSP Tunnel Terminology | 827
- Pop-and-Forward LSP Tunnel Label and Signaling | 828
- Pop-and-Forward LSP Tunnel Label Stacking | 829
- Pop-and-Forward LSP Tunnel Link Protection | 831

RSVP-TE Pop-and-Forward LSP Tunnel Supported and Unsupported Features | 832

Pop-and-forward LSPs introduces the notion of pre-installed per traffic engineering link pop labels that are shared by RSVP-TE LSPs that traverse these links and significantly reducing the required forwarding plane state . A transit label-switching router (LSR) allocates a unique pop label per traffic engineering link with a forwarding action to pop the label and forward the packet over that traffic engineering link should the label appear at the top of the packet. These pop labels are sent back in the RESV message of the LSP at each LSR and further recorded in the record route object (RRO). The label stack is constructed from the recorded labels in the RRO and pushed by the ingress label edge router (LER), as each transit hop performs a pop-and-forward action on its label. The pop-and-forward tunnels enhances the RSVP-TE control plane feature benefits with the simplicity of the shared MPLS forwarding plane.

Benefits of RSVP-TE Pop-and-Forward LSP Tunnels

- **Scaling advantage of RSVP-TE**—Any platform-specific label space limit on an LSR is prevented from being a constraint to the control plane scaling on that interface.
- **Reduced forwarding plane state**—The transit labels on a traffic engineering link are shared across RSVP-TE tunnels traversing the link, and are used independent of the ingress and egress devices of the LSPs, thereby significantly reducing the required forwarding plane state.
- **Reduced transit data plane state**—Because the pop labels are allocated per traffic engineering link and shared across LSPs, the total label state in the forwarding plane is reduced to a function of the number of RSVP neighbors on that interface.
- **Faster LSP setup time**—The forwarding plane state is not programmed during the LSP setup and teardown. As a result, the control plane need not wait sequentially at each hop for the forwarding plane to be programmed prior to sending the label upstream in the RESV message, resulting in reduced LSP setup time.
- **Backward compatibility**—This allows backward compatibility with transit LSRs that provide regular labels in RESV messages. Labels can be mixed across transit hops in a single MPLS RSVP-TE LSP. Certain LSRs can use traffic engineering link labels and others can use regular labels. The ingress can construct a label stack appropriately based on what type of label is recorded from every transit LSR.

Pop-and-Forward LSP Tunnel Terminology

The following terminology is used in the implementation of RSVP-TE pop-and-forward LSP tunnels:

- **Pop label**—An incoming label at an LSR that is popped and forwarded over a specific traffic-engineering link to a neighbor.

- **Swap label**—An incoming label at an LSR that is swapped to an outgoing label and forwarded over a specific downstream traffic engineering link.
- **Delegation label**—An incoming label at an LSR that is popped. A new set of labels is pushed before the packet is forwarded.
- **Delegation hop**— A transit hop that allocates a delegation label.
- **Application label depth (AppLD)**—Maximum number of application or service labels (for example, VPN, LDP, or IPv6 explicit-null labels) that can be beneath the RSVP transport labels. It is configured on a per-node basis, and is equally applicable for all LSPs, and is neither signaled nor advertised.
- **Outbound label depth (OutLD)**—Maximum number of labels that can be pushed before a packet is forwarded. This is local to the node, and is neither signaled nor advertised.
- **Additional transport label depth (AddTLD)**—Maximum number of other transport labels that can be added (for example, bypass label). This is a per-LSP parameter that is neither signaled or advertised. The value is discerned by checking if the LSP has been signaled with link protection (AddTLD=1) or without link protection (AddTLD=0).
- **Effective transport label depth (ETLD)**— Number of transport labels that the LSP hop can potentially send to its downstream hop. This value is signaled per LSP in the hop attributes subobject. The hop attributes subobject is added to the record route object (RRO) in the path message.

Pop-and-Forward LSP Tunnel Label and Signaling

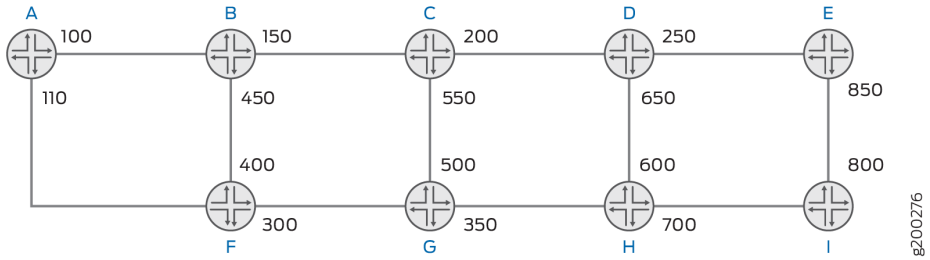
Every traffic engineering link is allocated a pop label that is installed in the mpls.0 routing table with a forwarding action to pop the label and forward the packet over the traffic engineering link to the downstream neighbor of the RSVP-TE tunnel.

For pop-and-forward LSP tunnels, the pop label for the traffic engineering link is allocated when the first RESV message for a pop-and-forward transit LSP arrives over that traffic engineering link. This is done to avoid preallocating pop labels and installing them in networks where pop-and-forward LSPs are not configured.

NOTE: For the pop-and-forward LSP tunnels to function effectively, we recommend that you configure the `maximum-labels` statement on all the interfaces in the RSVP-TE network.

Figure 53 on page 829 displays pop labels at all interfaces for neighboring devices.

Figure 53: Pop-and-Forward LSP Tunnel Labels



There are two pop-and-forward LSP tunnels—T1 and T2. Tunnel T1 is from Device A to Device E on path A-B-C-D-E. Tunnel T2 is from Device F to Device E on path F-B-C-D-E. Both the tunnels, T1 and T2, share the same traffic engineering links B-C, C-D, and D-E.

As RSVP-TE signals the setup of the pop-and-forward tunnel T1, the LSR D receives the RESV message from the egress E. Device D checks the next-hop traffic engineering link (D-E) and provides the pop label (250) in the RESV message for the tunnel. The label is sent in the label object and is also recorded in the label subobject (with the pop label bit set) carried in the RRO. Similarly, Device C provides the pop label (200) for the next- hop traffic engineering link C-D and Device B provides the pop label (150) for the next-hop traffic engineering link B-C. For the tunnel T2, the transit LSRs provide the same pop labels as described for tunnel T1.

Both the label edge routers (LERs), Device A and Device F, push the same stack of labels [150(top), 200, 250] for tunnels T1 and T2, respectively. The recorded labels in the RRO are used by the ingress LER to construct a stack of labels.

The pop-and-forward LSP tunnel labels are compatible with transit interfaces that use swap labels. Labels can be mixed across transit hops in a single MPLS RSVP-TE LSP, where certain LSRs can use pop labels and others can use swap labels. The ingress device constructs the appropriate label stack based on the label type recorded from every transit LSR.

Pop-and-Forward LSP Tunnel Label Stacking

Construction of Label Stack at the Ingress

The ingress LER checks the type of label received from each transit hop as recorded in the RRO in the RESV message and generates the appropriate label stack to use for the pop-and-forward tunnel.

The following logic is used by the ingress LER while constructing the label stack:

- Each RRO label subobject is processed starting with the label subobject from the first downstream hop.

- Any label provided by the first downstream hop is always pushed on the label stack. If the label type is a pop label, then any label from the succeeding downstream hop is also pushed on the constructed label stack.
- If the label type is a swap label, then any label from the succeeding downstream hop is not pushed on the constructed label stack.

Auto-Delegation of Label Stack

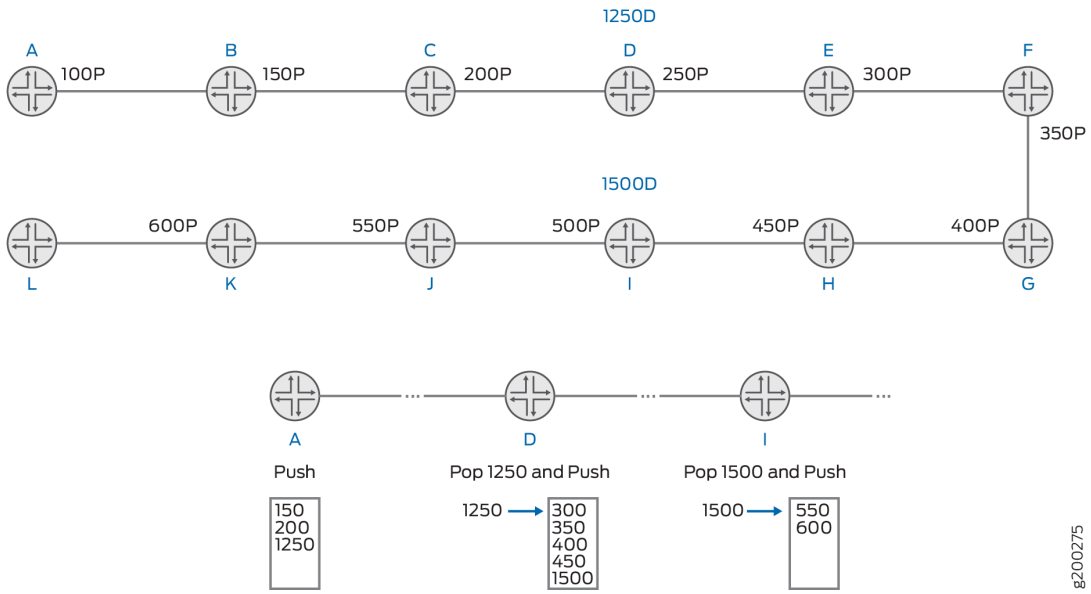
The ingress device runs the Constrained Shortest Path First (CSPF) to compute the path, and if the hop length is greater than the OutLD-AppLD-AddTLD, the ingress device cannot impose the entire label stack to reach the egress device.

When requesting RSVP-TE to signal the path, the ingress device always requests autodelegation for the LSP, where one or more transit hops automatically select themselves as delegation hops to push the label stack to reach the next delegation hop. Junos OS uses an algorithm based on the received Effective Transport Label-Stack Depth (ETLD), that each transit executes to decide whether it should autoselect itself as a delegation hop. This algorithm is based on the section on ETLD in the Internet draft draft-ietf-mpls-rsvp-shared-labels-00.txt (expires September 11 2017), *Signaling RSVP-TE Tunnels on a Shared MPLS Forwarding Plane*.

The label stack imposed by the ingress device delivers the packet up to the first delegation hop. Each delegation hop's label stack also includes the delegation label of the next delegation hop at the bottom of the stack.

Figure 54 on page 831 displays labels at every device interface, where Device D and Device I are delegation hops, *[Label]* P is the pop label, and *[Label]* D is the delegation label. The RSVP-TE pop-and-forward LSP tunnel is A-B-C-D-E-F-G-H-I-J-K-L. Delegation label 1250 represents (300, 350, 400, 450, 1500); Delegation label 1500 represents (550, 600).

Figure 54: Pop-and-Forward LSP Tunnel Pop and Delegation Labels



In this approach, for the tunnel, the ingress LER Device A pushes (150, 200, 1250). At LSR Device D, the delegation label 1250 gets popped and labels 300, 350, 400, 450, and 1500 get pushed. At LSR Device I, the delegation label 1500 gets popped and the remaining set of labels (550, 600) get pushed. In Junos OS, the pop and push action occurs as a swap to the bottom label of the outgoing stack and push the remaining labels.

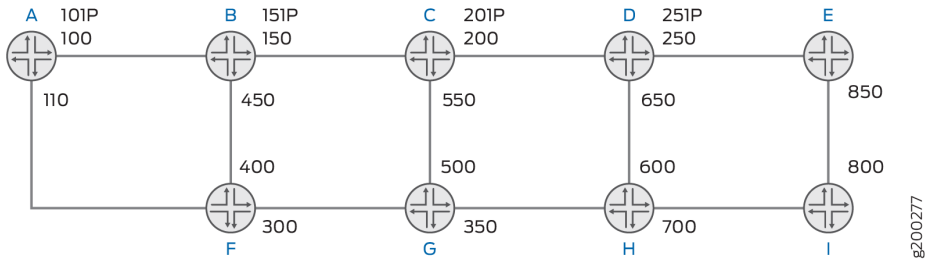
A delegation label and the LSP segment that it covers can be shared by multiple pop-and-forward LSPs. A LSP delegation segment consist of an ordered set of hops (IP addresses and labels) as seen in the RESV RRO. The delegation label (and the segment that it covers) is not owned by a particular LSP, but can be shared. When all LSPs using a delegation label are deleted, the delegation label (and route) is deleted.

Pop-and-Forward LSP Tunnel Link Protection

To provide link protection at a point of local repair (PLR) with a pop-and-forward data plane, the LSR allocates a separate pop label for the traffic engineering link that is used for the RSVP-TE tunnels that request link protection from the ingress device. No signaling extensions are required to support link protection for the RSVP-TE tunnels over the pop-and-forward data plane.

Figure 55 on page 832 displays pop labels at every device interface; labels marked with P are pop labels that offer link protection for the traffic-engineering link.

Figure 55: Pop-and Forward LSP Tunnel Link Protection



At each LSR, link-protected pop labels can be allocated for each traffic engineering link, and a link-protecting facility bypass LSP (which is not a pop-and-forward LSP, but rather a normal bypass LSP) can be created to protect the traffic engineering link. These labels can be sent in the RESV message by the LSR for LSPs requesting link protection over the specific traffic engineering link. Because the facility bypass terminates at the next hop (merge point), the incoming pop label on the packet at the PLR is what the merge point expects.

For example, LSR Device B can install a facility bypass LSP for the link-protected pop label 151. When the traffic engineering link B-C is up, LSR Device B pops 151 and sends the packet to C. If the traffic engineering link B-C is down, the LSR can pop 151 and send the packet through the facility backup to Device C.

RSVP-TE Pop-and-Forward LSP Tunnel Supported and Unsupported Features

Junos OS supports the following features with RSVP-TE pop-and-forward LSP tunnels:

- Pop labels per RSVP neighbor for unprotected LSP.
- Pop labels per RSVP neighbor for LSPs requesting link protection using facility bypass
- Autodelegation of LSP segment.
- Mixed label mode, where certain transit LSRs do not support pop-and-forward LSP tunnels
- LSP ping and traceroute
- All existing CSPF constraint.
- Load balancing of traffic between pop-and-forward LSPs and regular point-to-point RSVP-TE LSP.
- Autobandwidth, LDP tunneling, and TE++ container LSP.
- Aggregated Ethernet interface.
- Virtual platforms support, such as Juniper Networks vMX Virtual Router.

- 64-bit support
- Logical systems

Junos OS does not support the following functionality for RSVP-TE pop-and-forward LSP tunnels:

- Node link protection
- Detour protection for MPLS fast reroute
- Point-to-multipoint LSPs.
- Switch-away LSP.
- Generalized MPLS (GMPLS) LSPs (including bidirectional LSPs, associated LSPs, VLAN user-to-network interface [UNI] and so on)
- IP Flow Information Export (protocol) (IPFIX) inline flow sampling for MPLS template
- RFC 3813, *Multiprotocol Label Switching (MPLS) Label Switching Router (LSR) Management Information Base (MIB)*
- IPv4 Explicit-null (Inserting label 0 at the bottom of the label stack is not supported. If there are service labels beneath the RSVP-TE pop-and-forward label stack, because the penultimate hop for the LSP copies the EXP value to the service label, this can allow continuity of class of service (CoS) across the MPLS forwarding plane).
- Ultimate-hop popping (UHP)
- Graceful Routing Engine switchover (GRES)
- Nonstop active routing (NSR)

Segment Routing LSP Configuration

IN THIS SECTION

- Enabling Distributed CSPF for Segment Routing LSPs | **834**
- Static Segment Routing Label Switched Path | **841**
- Routing Engine-based S-BFD for Segment-Routing Traffic Engineering with First-Hop Label Resolution | **880**

- Configuring Static Adjacency Segment Identifier for Aggregate Ethernet Member Links Using Single-Hop Static LSP | **911**
- Non-Router-ID Endpoints in Segment Routing Traffic Engineering | **913**
- Computing Delay Optimized Intradomain and Interdomain Segment Routing Paths | **914**
- Understanding SRv6 TE Tunnel Local Path Computation | **919**
- Configuring SRv6 TE Tunnel Local Path Computation | **920**
- Understand BGP VPN Option C MPLS over SRv6 | **921**
- Platform-Specific Segment Routing LSP Behavior | **925**

Enabling Distributed CSPF for Segment Routing LSPs

IN THIS SECTION

- Distributed CSPF Computation Constraints | **835**
- Distributed CSPF Computation Algorithm | **835**
- Distributed CSPF Computation Database | **836**
- Configuring Distributed CSPF Computation Constraints | **836**
- Distributed CSPF Computation | **838**
- Interaction Between Distributed CSPF Computation and SR-TE Features | **838**
- Distributed CSPF Computation Sample Configurations | **839**

With the distributed Constrained Shortest Path First (CSPF) for segment routing LSP feature, you can compute a segment routing LSP locally on the ingress device according to the constraints you have configured. With this feature, the LSPs are optimized based on the configured constraints and metric type (traffic-engineering or IGP). The LSPs are computed to utilize the available ECMP paths to the destination with segment routing label stack compression enabled or disabled.

Use Feature Explorer to confirm platform and release support for specific features.

Review the "Platform-Specific Segment Routing LSP Behavior" on page 925 section for notes related to your platform.

Distributed CSPF Computation Constraints

Segment routing LSP paths are computed when all the configured constraints are met.

The distributed CSPF computation feature supports the following subset of constraints specified in the Internet draft, draft-ietf-spring-segment-routing-policy-03.txt, *Segment Routing Policy for Traffic Engineering*:

- Inclusion and exclusion of administrative groups.
- Inclusion of loose or strict hop IP addresses.

> **NOTE**: You can specify only router IDs in the loose or strict hop constraints. Labels and other IP addresses cannot be specified as loose or strict hop constraints in Junos OS Release 19.2R1-S1.

- Maximum number of segment IDs (SIDs) in the segment list.
- Maximum number of segment lists per candidate segment routing path.

The distributed CSPF computation feature for segment routing LSPs does not support the following types of constraints and deployment scenarios:

- IPV6 addresses.
- Inter domain segment routing traffic engineering (SR-TE) LSPs.
- Unnumbered interfaces.
- Multiple protocols routing protocols such as, OSPF, ISIS, and BGP-LS, enabled at the same time.
- Computation with prefixes or anycast addresses as destinations.
- Including and excluding interface IP addresses as constraints.

Distributed CSPF Computation Algorithm

The distributed CSPF computation feature for segment routing LSPs uses the label stack compression algorithm with CSPF.

Label Stack Compression Enabled

A compressed label stack represents a set of paths from a source to a destination. It generally consists of node SIDs and adjacency SIDs. When label stack compression is enabled, the result of the computation

is a set of paths that maximize ECMP to the destination, with minimum number of SIDs in the stack, while conforming to constraints.

Label Stack Compression Disabled

The multipath CSPF computation with label stack compression disabled finds up to *N* segment lists to destination, where:

- The cost of all segment lists is equal to and the same as the shortest traffic-engineering metric to reach the destination.
- Each segment list is comprised of adjacency SIDs.
- The value of *N* is the maximum number of segment lists allowed for the candidate path by configuration.
- No two segment lists are identical.
- Each segment list satisfies all the configured constraints.

Distributed CSPF Computation Database

The database used for SR-TE computation has all links, nodes, prefixes and their characteristics irrespective of whether traffic-engineering is enabled in those advertising nodes. In other words, it is the union of the traffic-engineering database (TED) and the IGP link state database of all domains that the computing node has learnt from. As a result, for CSPF to work, you must include the `igp-topology` statement at the `[edit protocols isis traffic-engineering]` hierarchy level.

Configuring Distributed CSPF Computation Constraints

You can use a compute profile to logically group the computation constraints. These compute profiles are referenced by the segment routing paths for computing the primary and secondary segment routing LSPs.

To configure a compute profile, include the *compute-profile* statement at the `[edit protocols source-packet-routing]` hierarchy level.

The configuration for the supported computation constraints include:

- **Administrative groups**

  You can configure *admin-groups* under the `[edit protocols mpls]` hierarchy level. Junos OS applies the administrative group configuration to the segment routing traffic-engineering (SR-TE) interfaces.

To configure the computation constraints you can specify three categories for a set of administrative groups. The computation constraint configuration can be common to all candidate segment routing paths, or it can be under individual candidate paths.

- `include-any`—Specifies that any link with at least one of the configured administrative groups in the list is acceptable for the path to traverse.
- `include-all`—Specifies that any link with all of the configured administrative groups in the list is acceptable for the path to traverse.
- `exclude`—Specifies that any link which does not have any of the configured administrative groups in the list is acceptable for the path to traverse.

NOTE: Administrative groups are only advertised if you either:

- Enable RSVP on the interfaces.
- Configure `edit protocols isis traffic-engineering advertisement always` if you dont want to enable RSVP.

- **Explicit path**

  You can specify a series of router IDs in the compute profile as a constraint for computing the SR-TE candidate paths. Each hop has to be an IPv4 address and can be of type strict or loose. If the type of a hop is not configured, strict is used. You must include the `compute` option under the *segment-list* statement when specifying the explicit path constraint.

- **Maximum number of segment lists (ECMP paths)**

  You can associate a candidate path with a number of dynamic segment-lists. The paths are ECMP paths, where each segment-list translates into a next hop gateway with active weight. These paths are a result of path computation with or without compression.

  You can configure this attribute using the `maximum-computed-segment-lists` *`maximum-computed-segment-lists`* option under the *compute-profile* configuration statement. This configuration determines the maximum number of such segment lists computed for a given primary and secondary LSP.

- **Maximum segment list depth**

  The maximum segment list depth computation parameter ensures that amongst the ECMP paths that satisfy all other constraints such as administrative group, only the paths that have segment lists less than or equal to the maximum segment list depth are used. When you configure this parameter as a constraint under the compute-profile, it overrides the `maximum-segment-list-depth` configuration under the `[edit protocols source-packet-routing]` hierarchy level, if present.

You can configure this attribute using the `maximum-segment-list-depth` *`maximum-segment-list-depth`* option under the *compute-profile* configuration statement.

- **Protected or unprotected adjacency SIDs**

  You can configure protected or unprotected adjacency SID as a constraint under the *compute-profile* to avoid links with the specified SID type.

- **Metric type**

  You can specify the type of metric on the link to be used for computation. By default, SR-TE LSPs use traffic-engineering metrics of the links for computation. The traffic-engineering metric for links is advertised by traffic-engineering extensions of IGP protocols. However, you may also choose to use the IGP-metric for computation by using the metric-type configuration in the compute profile.

  You can configure this attribute using the `metric-type (`*`igp | te`*`)` option under the *compute-profile* configuration statement.

Distributed CSPF Computation

The SR-TE candidate paths are computed locally such that they satisfy the configured constraints. When label stack compression is disabled, the multi-path CSPF computation result is a set of adjacency SID stacks. When label stack compression is enabled, the result is a set of compressed label stacks (composed of adjacent SIDs and node SIDs).

When secondary paths are computed, the links, nodes and SRLGs taken by the primary paths are not avoided for computation. For more information on primary and secondary paths, see "Configuring Primary and Secondary LSPs" on page 683.

For any LSPs with unsuccessful computation result, the computation is retried as traffic-engineering database (TED) changes.

Interaction Between Distributed CSPF Computation and SR-TE Features

Weights Associated With Paths of an SR-TE Policy

You can configure weights against computed and static SR-TE paths, which contribute to the next hops of the route. However, a single path that has computation enabled can result in multiple segment lists. These computed segment lists are treated as ECMP amongst themselves. You can assign hierarchical ECMP weights to these segments, considering the weights assigned to each of the configured primaries.

BFD Liveliness Detection

You can configure BFD liveliness detection for the computed primary or secondary paths. Every computed primary or secondary path can result in multiple segment lists, as a result, the BFD

parameters configured against the segment lists are applied to all the computed segment lists. If all the active primary paths go down, the pre-programmed secondary path (if provided) becomes active.

inherit-label-nexthops

You are not required to explicitly enable the `inherit-label-nexthops` configuration under the `[edit protocols source-packet-routing segment-list` *`segment-list-name`*`]` hierarchy for the computed primary or secondary paths, as it is a default behavior.

Auto-Translate Feature

You can configure the auto-translate feature on the segment lists, and the primary or secondary paths with the auto-translate feature reference these segment lists. On the other hand, the primary or secondary on which compute feature is enabled cannot reference any segment list. As a result, you cannot enable both the compute feature and the auto-translate feature for a given primary or secondary path. However, you could have an LSP configured with a primary path with compute type and another with auto-translate type.

Distributed CSPF Computation Sample Configurations

Example 1

In Example 1,

- The non-computed primary path references a configured segment-list. In this example, the configured segment list *static_sl1* is referenced, and it also serves as the name for this primary path.
- A computed primary should have a name configured, and this name should not reference any configured segment list. In this example, *compute_segment1* is not a configured segment list.
- The *compute_profile_red* compute-profile is applied to the primary path with the name *compute_segment1*.
- The *compute_profile_red* compute-profile includes a segment list of type `compute`, which is used to specify the explicit path constraint for the computation.

```
[edit protocols source-packet-routing]
segment-list static_sl1{
    hop1 label 80000
}
segment-list exp_path1 {
    hop1 ip-address 10.1.1.1 loose
    hop2 ip-address 10.2.2.2
```

```
    compute
}
compute-profile compute_profile_red {
    include-any red
    segment-list exp_path1
    maximum-segment-list-depth 5
}
```

The weights for computed path next-hops and static next-hops are 2 and 3, respectively. Assuming the next-hops for computed paths are *comp_nh1*, *comp_nh2*, and *comp_nh3*, and the next-hop for static path is *static_nh*, the weights are applied as follows:

Next-Hop	Weight
comp_nh1	2
comp_nh2	2
comp_nh3	2
static_nh	9

Example 2

In Example 2, both the primary and secondary paths can be of compute type and can have their own compute-profiles.

```
[edit protocols source-packet-routing]
compute-profile compute_profile_green{
    include-any green
    maximum-segment-list-depth 5
}
compute-profile compute_profile_red{
    include-any red
    maximum-segment-list-depth 8
}
```

Example 3

In Example 3, when compute is mentioned under a primary or secondary path, it results in local computation of a path to the destination without any constraints or other parameters for the computation.

```
[edit protocols source-packet-routing]
source-routing-path srte_colored_policy1 {
    to 10.5.5.5
    color 5
    binding-sid 10001
    primary {
        compute_segment1 {
            compute
        }
    }
}
```

Static Segment Routing Label Switched Path

IN THIS SECTION

- Static Segment Routing LSP in MPLS Networks | **841**
- Example: Configuring Static Segment Routing Label Switched Path | **860**

The segment routing architecture enables the ingress devices in a core network to steer traffic through explicit paths. You can configure these paths using segment lists to define the paths that the incoming traffic should take. The incoming traffic may be labeled or IP traffic, causing the forwarding operation at the ingress device to be either a label swap, or a destination-based lookup.

Static Segment Routing LSP in MPLS Networks

IN THIS SECTION

- Introduction to Segment Routing LSPs | **842**
- Benefits of using Segment Routing LSPs | **843**

- Colored Static Segment Routing LSP | **843**
- Non-Colored Static Segment Routing LSP | **844**
- Static Segment Routing LSP Provisioning | **850**
- Static Segment Routing LSP Limitations | **850**
- Color-Based Mapping of VPN Services | **851**
- Tunnel Templates for PCE-Initiated Segment Routing LSPs | **859**

Source packet routing or segment routing is a control-plane architecture that enables an ingress devices in a core network to steer traffic through a specific set of nodes and links in the network without relying on the intermediate nodes in the network to determine the actual path it should take. You can configure these paths using segment lists to define the paths that the incoming traffic should take. The incoming traffic may be labeled or IP traffic, causing the forwarding operation at the ingress device to be either a label swap, or a destination-based lookup.

Introduction to Segment Routing LSPs

Segment routing leverages the source routing paradigm. A device steers a packet through an ordered list of instructions, called segments. A segment can represent any instruction, topological or service-based. A segment can have a local semantic to a segment routing node or to a global node within a segment routing domain. Segment routing enforces a flow through any topological path and service chain while maintaining per-flow state only at the ingress device to the segment routing domain. Segment routing can be directly applied to the MPLS architecture with no change on the forwarding plane. A segment is encoded as an MPLS label. An ordered list of segments is encoded as a stack of labels. The segment to process is on the top of the stack. Upon completion of a segment, the related label is popped from the stack.

Segment routing LSPs can either be dynamic or static in nature.

Dynamic segment routing LSPs—When a segment routing LSP is created either by an external controller and downloaded to an ingress device through Path Computation Element Protocol (PCEP) extensions, or from a BGP segment routing policy through BGP segment routing extensions, the LSP is dynamically provisioned. The segment list of the dynamic segment routing LSP is contained in the PCEP Explicit Route Object (ERO), or the BGP segment routing policy of the LSP.

Static segment routing LSPs—When a segment routing LSP is created on the ingress device through local configuration, the LSP is statically provisioned.

A static segment routing LSP can further be classified as colored and non-colored LSPs based on the configuration of the `color` statement at the `[edit protocols source-packet-routing source-routing-path lsp-name]` hierarchy level.

For example:

```
[edit protocols]
    source-packet-routing {
    source-routing-path lsp_name {
        to destination_address;
        color color_value;
        binding-sid binding-label;
        primary segment_list_1_name weight weight;
        ...
        primary segment_list_n_name weight weight;
        secondary segment_list_n_name;
        sr-preference sr_preference_value;
    }
}
```

Here, each primary and secondary statement refers to a segment list.

```
[edit protocols]
source-packet-routing {
    segment-list segment_list_name {
        hop_1_name label sid_label;
        ...
        hop_n_name label sid_label;
    }
}
```

Benefits of using Segment Routing LSPs

- Static segment routing does not rely on per LSP forwarding state on transit routers. Hence, removing the need of provisioning and maintaining per LSP forwarding state in the core.
- Provide higher scalability to MPLS networks.

Colored Static Segment Routing LSP

A static segment routing LSP configured with the `color` statement is called a colored LSP.

Understanding Colored Static Segment Routing LSPs

Similar to a BGP segment routing policy, the ingress route of the colored LSP is installed in the `inetcolor.0` or `inet6color.0` routing tables, with `destination-ip-address, color` as key for mapping IP traffic.

A static colored segment routing LSP may have a binding SID, for which a route is installed in the `mpls.0` routing table. This binding SID label is used to map labeled traffic to the segment routing LSP. The gateways of the route are derived from the segment list configurations under the primary and secondary paths.

Segment List of Colored Segment Routing LSPs

The colored static segment routing LSPs already provide support for first hop label mode of resolving an LSP. However, first hop IP mode is not supported for colored segment routing LSPs. A commit check feature is introduced to ensure that all the segment lists contributing to the colored routes have the minimum label present for all hops. If this requirement is not met, the commit is blocked.

Non-Colored Static Segment Routing LSP

A static segment routing LSP that is configured without the `color` statement is a non-colored LSP. Similar to PCEP segment routing tunnels, the ingress route is installed in the `inet.3` or `inet6.3` routing tables.

Junos OS supports non-colored static segment routing LSPs on ingress routers. You can provision non-colored static segment routing LSP by configuring one source routed path and one or more segment lists. These segment lists can be used by multiple non-colored segment routing LSPs.

Understanding Non-Colored Segment Routing LSPs

The non-colored segment routing LSP has a unique name and a destination IP address. An ingress route to the destination is installed in the inet.3 routing table with a default preference of 8 and a metric of 1. This route allows non-colored services to be mapped to the segment routing LSP pertaining to the destination. In case the non-colored segment routing LSP does not require an ingress route then the ingress route can be disabled. A non-colored segment routing LSP uses binding SID label to achieve segment routing LSP stitching. This label that can be used to model the segment routing LSP as a segment that may be further used to construct other segment routing LSPs in a hierarchical manner. The transit of the binding SID label, by default, has a preference of 8 and a metric of 1.

Statically configured non-colored segment routing LSPs on the ingress device are reported to the Path Computation Element (PCE) through a Path Computation Element Protocol (PCEP) session. These non-colored segment routing LSPs may have binding service identifier (SID) labels associated with them. With this feature, the PCE can use this binding SID label in the label stack to provision PCE-initiated segment routing LSP paths.

A non-colored segment routing LSP can have a maximum of 8 primary paths. If there are multiple operational primary paths then the packet forwarding engine (PFE) distributes traffic over the paths based on the load balancing factors like the weight configured on the path. This is equal cost multi path (ECMP) if none of the paths have a weight configured on them or weighted ECMP if at least one of the paths has a non-zero weight configured on the paths. In both the cases, when one or some of the paths fail, the PFE rebalances the traffic over the remaining paths that automatically leads to achieving path protection. A non-colored segment routing LSP can have a secondary path for dedicated path protection. Upon failure of a primary path, the PFE rebalances the traffic to the remaining functional primary paths. Otherwise, the PFE switches the traffic to the backup path, hence achieving path protection. A non-colored segment routing LSP may specify a metric at `[edit protocols source-packet-routing source-routing-path` *`lsp-name`*`]` for its ingress and binding-SID routes. Multiple non-colored segment routing LSPs have the same destination address that contribute to the next hop of the ingress route.

Multiple non-colored segment routing LSPs have the same destination address that contribute to the next hop of the ingress route. Each path ,either primary or secondary, of each segment routing LSP is considered as a gateway candidate, if the path is functional and the segment routing LSP has the best preference of all these segment routing LSPs. However, the maximum number of gateways that the next-hop can hold cannot exceed the RPD multi-path limit, which is 128 by default. Extra paths are pruned, firstly secondary paths and then primary paths. A given segment list may be referred multiple times as primary or secondary paths by these segment routing LSPs. In this case, there are multiple gateways, each having a unique segment routing LSP tunnel ID. These gateways are distinct, although they have identical outgoing label stack and interface. A non-colored segment routing LSP and a colored segment routing LSP may also have the same destination address. However, they correspond to different destination addresses for ingress routes, as the colored segment routing LSP's destination address is constructed with both its destination address and color.

NOTE: In the case where a static non-colored segment routing LSP and a PCEP-created segment routing LSP co-exist and have the same to address that contributes to the same ingress route, if they also have the same preference. Otherwise, the segment routing LSP with the best preference is installed for the route.

Segment List of Non-Colored Segment Routing LSPs

A segment list consists of a list of hops. These hops are based on the SID label or an IP address. The number of SID labels in the segment list should not exceed the maximum segment list limit. Maximum segment-list binding to a LSP tunnel is increased from 8 to 128, with maximum 1000 tunnels per system. A maximum of 128 primary paths are supported per static segment routing LSP. You can configure the maximum segment list limit at the `[edit protocols source-packet-routing]` hierarchy level.

The first hop of the non-colored static LSPs provides support for SID labels, in addition to IP addresses. With the first hop label support, MPLS fast reroute (FRR) and weighted equal-cost multipath is enabled for resolving the static non-colored segment routing LSPs, similar to colored static LSPs.

For the first-hop label mode to take effect, you must include the `inherit-label-nexthops` statement globally or individually for a segment list, and the first hop of the segment list must include both IP address and label. If the first hop includes only IP address, the `inherit-label-nexthops` statement does not have any effect.

You can configure `inherit-label-nexthops` at any one of the following hierarchies. The `inherit-label-nexthops` statement takes effect only if the segment list first hop includes both IP address and label.

- **Segment list level**—At the `[edit protocols source-packet-routing segment-list` *`segment-list-name`*`]` hierarchy level.
- **Globally**—At the `[edit protocols source-packet-routing]` hierarchy level.

When the `inherit-label-nexthops` statement is configured globally, it takes precedence over the segment-list level configuration, and the `inherit-label-nexthops` configuration is applied to all the segment lists. When the `inherit-label-nexthops` statement is not configured globally, only segment lists with both labels and IP address present in the first hop, and configured with `inherit-label-nexthops` statement are resolved using SID labels.

For dynamic non-colored static LSPs, that is the PCEP-driven segment routing LSPs, the `inherit-label-nexthops` statement must be enabled globally, as the segment-level configuration is not applied.

Table 1 describes the mode of segment routing LSP resolution based on the first hop specification.

Table 22: Non-Colored Static LSP Resolution Based on First Hop Specification

First Hop Specification	Mode of LSP Resolution

IP address only
 	
For example:
 	
`segment-list path-1 {`
`  hop-1 ip-address 172.16.12.2;`
`  hop-2 label 1000012;`
`  hop-3 label 1000013;`
`  hop-4 label 1000014;`
`}`
	The segment list is resolved using the IP address.

Table 22: Non-Colored Static LSP Resolution Based on First Hop Specification *(Continued)*

First Hop Specification	Mode of LSP Resolution

SID only
 	
For example:
 	
`segment-list path-2 {`
`  hop-1 label 1000011;`
`  hop-2 label 1000012;`
`  hop-3 label 1000013;`
`  hop-4 label 1000014;`
`}`
	The segment list is resolved using SID labels.

IP address and SID (without the `inherit-label-nexthops` configuration)
 	
For example:
 	
`segment-list path-3 {`
`  hop1 {`
`  **label 801006**;`
`  **ip-address 172.16.1.2**;`
`  }`
`  hop-2 label 1000012;`
`  hop-3 label 1000013;`
`  hop-4 label 1000014;`
`}`
	By default, the segment list is resolved using IP address.

Table 22: Non-Colored Static LSP Resolution Based on First Hop Specification *(Continued)*

First Hop Specification	Mode of LSP Resolution

IP address and SID (with the `inherit-label-nexthops` configuration)
 	
For example:
 	
`segment-list path-3 {`
`  **inherit-label-nexthops**;`
`  hop1 {`
`  **label 801006**;`
`  **ip-address 172.16.1.2**;`
`  }`
`  hop-2 label 1000012;`
`  hop-3 label 1000013;`
`  hop-4 label 1000014;`
`}`
	The segment list is resolved using SID labels.

You can use the `show route` *ip-address* `protocol spring-te active-path table inet.3` command to view the non-colored segment routing traffic-engineered LSPs having multiple segment lists installed in the inet.3 routing table.

For example:

```
user@host> show route 10.7.7.7 protocol spring-te active-path table inet.3
inet.3: 42 destinations, 59 routes (41 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

10.7.7.7/32         *[SPRING-TE/8] 00:01:25, metric 1, metric2 0
                    > to 10.11.1.2 via et-0/0/0.1, Push 801007
                      to 10.21.1.2 via et-0/0/2.1, Push 801007
                      to 10.102.1.2 via et-0/0/0.2, Push 801007, Push 801002(top)
                      to 10.21.1.2 via et-0/0/2.2, Push 801007, Push 801005(top)
                      to 10.103.1.2 via et-0/0/0.3, Push 801007, Push 801003(top)
                      to 10.203.1.2 via et-0/0/2.3, Push 801007, Push 801006(top)
                      to 10.104.1.2 via et-0/0/0.4, Push 801007, Push 801003, Push 801002(top)
                      to 10.204.1.2 via et-0/0/2.4, Push 801007, Push 801006, Push 801005(top)
```

NOTE: The first hop type of segment lists of a static segment routing LSP can cause a commit to fail, if:

- Different segment lists of a tunnel have different first hop resolution types. This is applicable to both colored and non-colored static segment routing LSPs. However, this does not apply for PCEP-driven LSPs; a system log message is generated for the mismatch in the first hop resolution type at the time of computing the path.

  For example:

  ```
  segment-list path-1 {
      hop-1 ip-address 172.16.12.2;
      hop-2 label 1000012;
      hop-3 label 1000013;
      hop-4 label 1000014;
  }
  segment-list path-2 {
      hop-1 label 1000011;
      hop-2 label 1000012;
      hop-3 label 1000013;
      hop-4 label 1000014;
  }
  source-routing-path lsp1 {
      to 172.16.10.1;
      primary {
          path-1;
          path-2;
      }
  }
  ```

  The commit of tunnel *lsp1* fails, as path-1 is of IP address mode and path-2 is of label mode.

- The binding SID is enabled for the static non-colored LSP whose segment list type is SID label.

  For example:

  ```
  segment-list path-3 {
      hop-1 label 1000011;
  ```

```
            hop-2 label 1000012;
            hop-3 label 1000013;
            hop-4 label 1000014;
        }
        source-routing-path lsp1 {
            to 172.16.10.1;
            binding-sid 333;
            primary {
                path-3;
            }
        }
```

Static Segment Routing LSP Provisioning

Segment provisioning is performed on per-router basis. For a given segment on a router, a unique service identifier (SID) label is allocated from a desired label pool which may be from the dynamic label pool for an adjacency SID label or from the segment routing global block (SRGB) for a prefix SID or node SID. The adjacency SID label can be dynamically allocated, which is the default behavior, or be allocated from a local static label pool (SRLB). A route for the SID label is then installed in the mpls.0 table.

Junos OS allows static segment routing LSPs by configuring the `segment` statement at the `[edit protocols mpls static-label-switched-path` *`static-label-switched-path`*`]` hierarchy level. A static segment LSP is identified by a unique SID label that falls under Junos OS static label pool. You can configure the Junos OS static label pool by configuring the `static-label-range` *`static-label-range`* statement at the `[edit protocols mpls label-range]` hierarchy level.

Static Segment Routing LSP Limitations

- Junos OS currently has a limitation that the next hop cannot be built to push more than the maximum segment list depth labels. So, a segment list with more than the maximum SID labels (excluding the SID label of the first hop which is used to resolve forwarding next-hop) is not usable for colored or non-colored segment routing LSPs. Also, the actual number allowed for a given segment routing LSP may be even lower than the maximum limit, if an MPLS service is on the segment routing LSP or the segment routing LSP is on a link or a node protection path. In all cases, the total number of service labels, SID labels, and link or node protection labels must not exceed the maximum segment list depth. You can configure the maximum segment list limit at `[edit protocols source-packet-routing]` hierarchy level. Multiple non-colored segment routing LSPs with less than or equal to the maximum SID labels can be stitched together to construct a longer segment routing LSP. This is called segment routing LSP stitching. It can be achieved using binding-SID label.

- The segment routing LSP stitching is actually performed at path level. If a non-colored segment routing LSP has multiple paths that is multiple segment lists, each path can be independently stitched to another non-colored segment routing LSP at a stitching point. A non-colored segment routing LSP which is dedicated to stitching may disable ingress route installation by configuring `no-ingress` statement at `[edit protocols source-packet-routing source-routing-path lsp-name]` hierarchy level.
- A maximum of 128 primary paths and 1 secondary path are supported per non-colored static segment routing LSP. If there is a violation in configuration, commit check fails with an error.
- Maximum segment-list binding to a LSP tunnel is increased from 8 to 128, with maximum 1000 tunnels per system. A maximum of 128 primary paths are supported per static segment routing LSP. As a limitation, the maximum sensor support for LSP path is 32000 only.
- If any segment-list is configured with more labels than the maximum segment list depth, the configuration commit check fails with an error.

Color-Based Mapping of VPN Services

You can specify color as a protocol next hop constraint (in addition to the IPv4 or IPv6 address) for resolving transport tunnels over static colored and BGP segment routing traffic-engineered (SR-TE) LSPs. This is called the color-IP protocol next hop resolution, where you are required to configure a resolution-map and apply to the VPN services. With this feature, you can enable color-based traffic steering of Layer 2 and Layer 3 VPN services.

Junos OS supports colored SR-TE LSPs associated with a single color. The color-based mapping of VPN services feature is supported on static colored LSPs and BGP SR-TE LSPs.

VPN Service Coloring

In general, a VPN service may be assigned a color on the egress router where the VPN NLRI is advertised, or on an ingress router where the VPN NLRI is received and processed.

You can assign a color to the VPN services at different levels:

- Per routing instance.
- Per BGP group.
- Per BGP neighbor.
- Per prefix.

Once you assign a color, the color is attached to a VPN service in the form of BGP color extended community.

You can assign multiple colors to a VPN service, referred to as multi-color VPN services. In such cases, the last color attached is considered as the color of the VPN service, and all other colors are ignored.

Multiple colors are assigned by egress and/or ingress devices through multiple policies in the following order:

- BGP export policy on the egress device.
- BGP import policy on the ingress device.
- VRF import policy on the ingress device.

The two modes of VPN service coloring are:

Egress Color Assignment

In this mode, the egress device (that is, the advertiser of the VPN NLRI) is responsible for coloring the VPN service. To enable this mode, you can define a routing policy, and apply it in the VPN service's routing-instance `vrf-export`, group export, or group neighbor export at the `[edit protocols bgp]` hierarchy level. The VPN NLRI is advertised by BGP with the specified color extended community.

For example:

```
[edit policy-options]
community red-comm {
    members color:0:50;
}
```

```
[edit policy-options]
policy-statement pol-color {
    term t1 {
        from {
            [any match conditions];
        }
        then {
            community add red-comm;
            accept;
        }
```

```
    }
}
```

```
[edit routing-instances]
vpn-X {
    ...
    vrf-export pol-color ...;
}
```

Or

NOTE: When you apply the routing policy as an export policy of a BGP group or BGP neighbor, you must include the `vpn-apply-export` statement at the BGP, BGP group, or BGP neighbor level in order for the policy to take an effect on the VPN NLRI.

```
[edit protocols bgp]
group PEs {
    ...
    neighbor PE-A {
        export pol-color ...;
        vpn-apply-export;
    }
}
```

The routing policies are applied to Layer 3 VPN prefix NLRIs, Layer 2 VPN NRLIs, and EVPN NLRIs. The color extended community is inherited by all the VPN routes, imported, and installed in the target VRFs on one or multiple ingress devices.

Ingress Color Assignment

In this mode, the ingress device (that is, the receiver of the VPN NLRI) is responsible for coloring the VPN service. To enable this mode, you can define a routing policy, and apply it to the VPN service's routing-instance `vrf-import`, group import, or group neighbor import at the `[edit protocols bgp]` hierarchy level. All the VPN routes matching the routing policy is attached with the specified color extended community.

For example:

```
[edit routing-options]
community red-comm {
    members color:0:50;
}
```

```
[edit policy-options]
policy-statement pol-color {
    term t1 {
        from {
            [any match conditions];
        }
        then {
            community add red-comm;
            accept;
        }
    }
}
```

```
[edit routing-instances]
vpn-Y {
    ...
    vrf-import pol-color ...;
}
```

Or

```
[edit protocols bgp]
group PEs {
    ...
    neighbor PE-B {
        import pol-color ...;
    }
}
```

Specifying VPN Service Mapping Mode

To specify flexible VPN service mapping modes, you must define a policy using the `resolution-map` statement, and refer the policy in a VPN service's routing-instance `vrf-import`, group import, or group neighbor import at the `[edit protocols bgp]` hierarchy level. All the VPN routes matching the routing policy are attached with the specified resolution-map.

For example:

```
[edit policy-options]
resolution-map map-A {
    <mode-1>;
    <mode-2>;
    ...
}
policy-statement pol-resolution {
    term t1 {
        from {
            [any match conditions];
        }
        then {
            resolution-map map-A;
            accept;
        }
    }
}
```

You can apply import policy to the VPN service's routing-instance.

```
[edit routing-instances]
vpn-Y {
    ...
    vrf-import pol-resolution ...;
}
```

You can also apply the import policy to a BGP group or BGP neighbor.

```
[edit protocols bgp]
group PEs {
    ...
    neighbor PE-B {
```

```
        import pol-resolution ...;
    }
}
```

NOTE: Each VPN service mapping mode should have a unique name defined in the resolution-map. Only a single entry of IP-color is supported in the resolution-map, where the VPN route(s) are resolved using a colored-IP protocol next hop in the form of `ip-address:color`.

Color-IP Protocol Next Hop Resolution

The protocol next hop resolution process is enhanced to support colored-IP protocol next hop resolution. For a colored VPN service, the protocol next hop resolution process takes a color and a resolution-map, builds a colored-IP protocol next hop in the form of *IP-address:color*, and resolves the protocol next hop in the inet6color.0 routing table.

You must configure a policy to support multipath resolution of colored Layer 2 VPN, Layer 3 VPN, or EVPN services over colored LSPs. The policy must then be applied with the relevant RIB table as the resolver import policy.

For example:

```
[edit policy-options]
policy-statement mpath {
    then multipath-resolve;
}
```

```
[edit routing-options]
resolution {
    rib bgp.l3vpn.0 {
        inetcolor-import mpath;
    }
}

resolution {
    rib bgp.l3vpn-inet6.0 {
        inet6color-import mpath;
    }
}
```

```
resolution {
    rib bgp.l2vpn.0 {
        inetcolor-import mpath;
    }
}

resolution {
    rib mpls.0 {
        inetcolor-import mpath;
    }
}

resolution {
    rib bgp.evpn.0 {
        inetcolor-import mpath;
    }
}
```

Fallback to IP Protocol Next Hop Resolution

If a colored VPN service does not have a resolution-map applied to it, the VPN service ignores its color and falls back to the IP protocol next hop resolution. Conversely, if a non-colored VPN service has a resolution-map applied to it, the resolution-map is ignored, and the VPN service uses the IP protocol next hop resolution.

The fallback is a simple process from colored SR-TE LSPs to LDP LSPs, by using a RIB group for LDP to install routes in inet{6}color.0 routing tables. A longest prefix match for a colored-IP protocol next hop ensures that if a colored SR-TE LSP route does not exist, an LDP route with a matching IP address should be returned.

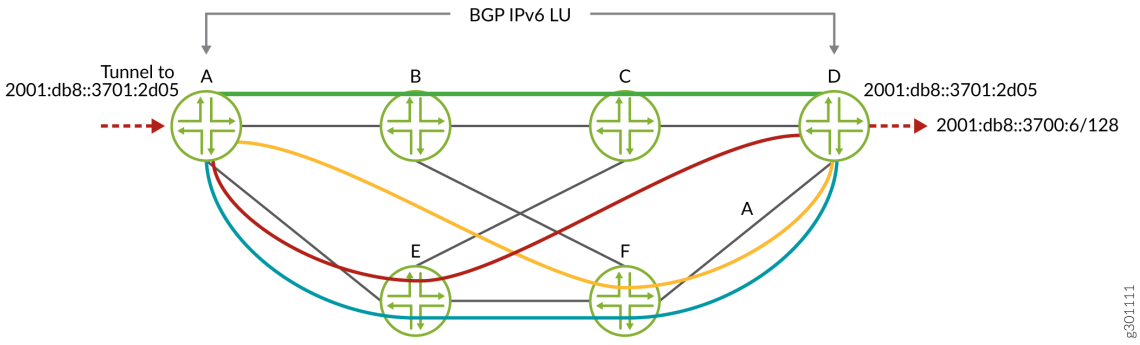
BGP Labeled Unicast Color-based Mapping over SR-TE

BGP Labeled Unicast (BGP-LU) can resolve IPv4 or IPv6 routes over segment routing–traffic engineering (SR-TE) for both IPv4 and IPv6 address families. BGP-LU supports mapping a BGP community color and defining a `resolution map` for SR-TE. A colored protocol next hop is constructed and it is resolved on a colored SR-TE tunnel in the `inetcolor.0` or `inet6color.0` table. BGP uses `inet.3` and `inet6.3` tables for non-color based mapping. This enables you to advertise BGP-LU IPv6 and IPv4 prefixes with an IPv6 next-hop address in IPv6-only networks where routers do not have any IPv4 addresses configured. With this feature, Currently we support BGP IPv6 LU over SR-TE with IS-IS underlay.

In Figure 1, the controller configures 4 colored tunnels in an IPv6 core network configured with SR-TE. Each colored tunnel takes a different path to the destination router D depending on the defined

resolution map. The controller configures a colored SR-TE tunnel to 2001:db8::3701:2d05 interface in router D . BGP imports policies to assign a color and resolution map to the received prefix 2001:db8::3700:6/128. Based on the assigned community color, BGP-LU resolves the colored next hop for BGP IPv6 LU prefix according to the assigned resolution map policy.

Figure 56: BGP IPv6 LU over colored IPv6 SR-TE



BGP-LU supports the following scenarios:

- BGP IPv4 LU over colored BGP IPv4 SR-TE, with ISIS/OSPF IPv4 SR extensions.
- BGP IPv4 LU over static colored and non-colored IPv4 SR-TE, with ISIS/OSPF IPv4 SR extensions.
- BGP IPv6 LU over colored BGP IPv6 SR-TE, with ISIS IPv6 SR extensions.
- BGP IPv6 LU over static colored and non-colored IPv6 SR-TE, with ISIS IPv6 SR extensions.
- IPv6 Layer 3 VPN services with IPv6 local address and IPv6 neighbor address.
- IPv6 Layer 3 VPN services over BGP IPv6 SR-TE, with ISIS IPv6 SR extensions.
- IPv6 Layer 3 VPN services over static-colored and non-colored IPv6 SR-TE, with ISIS IPv6 SR extensions.

Supported and Unsupported Features for Color-Based Mapping of VPN Services

The following features and functionality are supported with color-based mapping of VPN services:

- BGP Layer 2 VPN (Kompella Layer 2 VPN)
- BGP EVPN
- Resolution-map with a single IP-color option.
- Colored IPv4 and IPv6 protocol next hop resolution.

- Routing information base (also known as routing table) group based fallback to LDP LSP in inetcolor.0 routing table.
- Colored SR-TE LSP.
- Virtual platforms.
- 64-bit Junos OS.
- Logical systems.
- BGP labeled unicast.

The following features and functionality are not supported with color-based mapping of VPN services:

- Colored MPLS LSPs, such as RSVP, LDP, BGP-LU, static.
- Layer 2 circuit
- FEC-129 BGP auto-discovered and LDP-signaled Layer 2 VPN.
- VPLS
- MVPN
- IPv4 and IPv6 using resolution-map.

Tunnel Templates for PCE-Initiated Segment Routing LSPs

You can configure a tunnel template for PCE-initiated segment routing LSPs to pass down two additional parameters for these LSPs - Bidirectional forwarding detection (BFD) and LDP tunneling.

When a PCE-Initiated segment routing LSP is being created, the LSP is checked against policy statements (if any) and if there is a match, the policy applies the configured template for that LSP. The template configuration is inherited only if it is not provided by the LSP source (PCEP); for example, metric.

To configure a template:

1. Include the *source-routing-path-template* statement at the `[edit protocols source-packet-routing]` hierarchy level. You can configure the additional BFD and LDP tunneling parameters here.
2. Include the source-routing-path-template-map statement at the `[edit protocols source-packet-routing]` hierarchy level to list the policy statements against which the PCE-initiated LSP should be checked.
3. Define a policy to list the LSPs on which the template should be applied.

The `from` statement can include either the LSP name or LSP regular expression using the `lsp` and `lsp-regex` match conditions. These options are mutually exclusive, so you can specify only one option at a given point in time.

The `then` statement must include the `sr-te-template` option with an accept action. This applies the template to the PCE-initiated LSP.

Take the following into consideration when configuring a template for PCE-initiated LSPs:

- Template configuration is not applicable to staticalyy configured segment routing LSPs, or any other client's segment routing LSP.
- PCEP-provided configuration has precedence over template configuration.
- PCEP LSP does not inherit template segment-list configuration.

Example: Configuring Static Segment Routing Label Switched Path

IN THIS SECTION

- Requirements | **860**
- Overview | **861**
- Configuration | **861**
- Verification | **875**

This example shows how to configure static segment routing label switched paths (LSPs) in MPLS networks. This configuration helps to bring higher scalability to MPLS networks.

Requirements

This example uses the following hardware and software components:

- Seven MX Series 5G Universal Routing Platforms
- Junos OS Release 18.1 or later running on all the routers

Before you begin, be sure you configure the device interfaces.

Overview

IN THIS SECTION

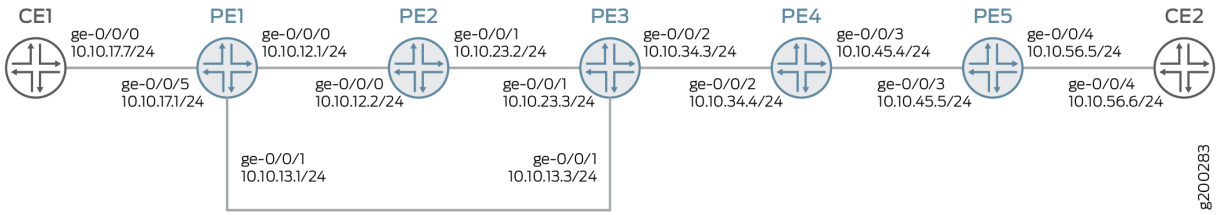
- Topology | **861**

Junos OS a set of explicit segment routing paths are configured on the ingress router of a non-colored static segment routing tunnel by configuring the `segment-list` statement at the `[edit protocols source-packet-routing]` hierarchy level. You can configure segment routing tunnel by configuring the `source-routing-path` statement at `[edit protocols source-packet-routing]` hierarchy level. The segment routing tunnel has a destination address and one or more primary paths and optionally secondary paths that refer to the segment list. Each segment list consists of a sequence of hops. For non-colored static segment routing tunnel, the first hop of the segment list specifies an immediate next hop IP address and the second to Nth hop specifies the segment identifies (SID) labels corresponding to the link or node which the path traverses. The route to the destination of the segment routing tunnel is installed in inet.3 table.

Topology

In this example, configure layer 3 VPN on the provider edge routers PE1 and PE5. Configure the MPLS protocol on all the routers. The segment routing tunnel is configured from router PE1 to router PE5 with a primary path configured on router PE1 and router PE5 . Router PE1 is also configured with secondary path for path protection. The transit routers PE2 to PE4 are configured with adjacency SID labels with label pop and an outgoing interface.

Figure 57: Static Segment Routing Label Switched Path



Configuration

IN THIS SECTION

- CLI Quick Configuration | **862**

- Configuring Device PE1 | 866
- Configuring Device PE2 | 872
- Results | 874

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

PE1

```
set interfaces ge-0/0/0 unit 0 family inet address 10.10.12.1/24
set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 5
set interfaces ge-0/0/1 unit 0 family inet address 10.10.13.1/24
set interfaces ge-0/0/1 unit 0 family mpls maximum-labels 5
set interfaces ge-0/0/5 unit 0 family inet address 10.10.17.1/24
set routing-options autonomous-system 65000
set routing-options forwarding-table export load-balance-policy
set routing-options forwarding-table chained-composite-next-hop ingress l3vpn
set protocols mpls interface ge-0/0/0.0
set protocols mpls interface ge-0/0/1.0
set protocols mpls label-range static-label-range 1000000 1000999
set protocols bgp group pe type internal
set protocols bgp group pe local-address 192.168.147.211
set protocols bgp group pe family inet-vpn unicast
set protocols bgp group pe neighbor 192.168.146.181
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols source-packet-routing segment-list sl-15-primary hop-1 ip-address 10.10.13.3
set protocols source-packet-routing segment-list sl-15-primary hop-2 label 1000134
set protocols source-packet-routing segment-list sl-15-primary hop-3 label 1000145
set protocols source-packet-routing segment-list sl-15-backup hop-1 ip-address 10.10.12.2
set protocols source-packet-routing segment-list sl-15-backup hop-2 label 1000123
set protocols source-packet-routing segment-list sl-15-backup hop-3 label 1000134
set protocols source-packet-routing segment-list sl-15-backup hop-4 label 1000145
set protocols source-packet-routing source-routing-path lsp-15 to 192.168.146.181
set protocols source-packet-routing source-routing-path lsp-15 binding-sid 1000999
```

```
set protocols source-packet-routing source-routing-path lsp-15 primary sl-15-primary
set protocols source-packet-routing source-routing-path lsp-15 secondary sl-15-backup
set policy-options policy-statement VPN-A-export term a from protocol ospf
set policy-options policy-statement VPN-A-export term a from protocol direct
set policy-options policy-statement VPN-A-export term a then community add VPN-A
set policy-options policy-statement VPN-A-export term a then accept
set policy-options policy-statement VPN-A-export term b then reject
set policy-options policy-statement VPN-A-import term a from protocol bgp
set policy-options policy-statement VPN-A-import term a from community VPN-A
set policy-options policy-statement VPN-A-import term a then accept
set policy-options policy-statement VPN-A-import term b then reject
set policy-options policy-statement bgp-to-ospf from protocol bgp
set policy-options policy-statement bgp-to-ospf from route-filter 10.10.0.0/16 orlonger
set policy-options policy-statement bgp-to-ospf then accept
set policy-options policy-statement load-balance-policy then load-balance per-packet
set policy-options community VPN-A members target:65000:1
set routing-instances VRF1 instance-type vrf
set routing-instances VRF1 interface ge-0/0/5.0
set routing-instances VRF1 route-distinguisher 192.168.147.211:1
set routing-instances VRF1 vrf-import VPN-A-import
set routing-instances VRF1 vrf-export VPN-A-export
set routing-instances VRF1 vrf-table-label
set routing-instances VRF1 protocols ospf export bgp-to-ospf
set routing-instances VRF1 protocols ospf area 0.0.0.0 interface ge-0/0/5.0
```

PE2

```
set interfaces ge-0/0/0 unit 0 family inet address 10.10.12.2/24
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 10.10.23.2/24
set interfaces ge-0/0/1 unit 0 family mpls
set protocols mpls static-label-switched-path adj-23 segment 1000123
set protocols mpls static-label-switched-path adj-23 segment next-hop 10.10.23.3
set protocols mpls static-label-switched-path adj-23 segment pop
set protocols mpls static-label-switched-path adj-21 segment 1000221
set protocols mpls static-label-switched-path adj-21 segment next-hop 10.10.12.1
set protocols mpls static-label-switched-path adj-21 segment pop
set protocols mpls interface ge-0/0/0.0
set protocols mpls interface ge-0/0/1.0
set protocols mpls label-range static-label-range 1000000 1000999
```

```
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
```

PE3

```
set interfaces ge-0/0/0 unit 0 family inet address 10.10.13.3/24
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 10.10.23.3/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 10.10.34.3/24
set interfaces ge-0/0/2 unit 0 family mpls
set protocols mpls static-label-switched-path adj-34 segment 1000134
set protocols mpls static-label-switched-path adj-34 segment next-hop 10.10.34.4
set protocols mpls static-label-switched-path adj-34 segment pop
set protocols mpls static-label-switched-path adj-32 segment 1000232
set protocols mpls static-label-switched-path adj-32 segment next-hop 10.10.23.2
set protocols mpls static-label-switched-path adj-32 segment pop
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols mpls label-range static-label-range 1000000 1000999
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
```

PE4

```
set interfaces ge-0/0/2 unit 0 family inet address 10.10.34.4/24
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 family inet address 10.10.45.4/24
set interfaces ge-0/0/3 unit 0 family mpls
set protocols mpls static-label-switched-path adj-45 segment 1000145
set protocols mpls static-label-switched-path adj-45 segment next-hop 10.10.45.5
set protocols mpls static-label-switched-path adj-45 segment pop
set protocols mpls static-label-switched-path adj-43 segment 1000243
set protocols mpls static-label-switched-path adj-43 segment next-hop 10.10.34.3
set protocols mpls static-label-switched-path adj-43 segment pop
set protocols mpls interface ge-0/0/2.0
set protocols mpls interface ge-0/0/3.0
set protocols mpls label-range static-label-range 1000000 1000999
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0
```

PE5

```
set interfaces ge-0/0/3 unit 0 family inet address 10.10.45.5/24
set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 5
set interfaces ge-0/0/4 unit 0 family inet address 10.10.56.5/24
set routing-options autonomous-system 65000
set protocols mpls interface ge-0/0/3.0
set protocols mpls label-range static-label-range 1000000 1000999
set protocols bgp group pe type internal
set protocols bgp group pe local-address 192.168.146.181
set protocols bgp group pe family inet-vpn unicast
set protocols bgp group pe neighbor 192.168.147.211
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols bfd sbfd local-discriminator 0.0.0.32 minimum-receive-interval 1000
set protocols source-packet-routing segment-list sl-51 hop-1 ip-address 10.10.45.4
set protocols source-packet-routing segment-list sl-51 hop-2 label 1000243
set protocols source-packet-routing segment-list sl-51 hop-3 label 1000232
set protocols source-packet-routing segment-list sl-51 hop-4 label 1000221
set protocols source-packet-routing source-routing-path lsp-51 to 192.168.147.211
set protocols source-packet-routing source-routing-path lsp-51 primary sl-51
set policy-options policy-statement VPN-A-export term a from protocol ospf
set policy-options policy-statement VPN-A-export term a from protocol direct
set policy-options policy-statement VPN-A-export term a then community add VPN-A
set policy-options policy-statement VPN-A-export term a then accept
set policy-options policy-statement VPN-A-export term b then reject
set policy-options policy-statement VPN-A-import term a from protocol bgp
set policy-options policy-statement VPN-A-import term a from community VPN-A
set policy-options policy-statement VPN-A-import term a then accept
set policy-options policy-statement VPN-A-import term b then reject
set policy-options policy-statement bgp-to-ospf from protocol bgp
set policy-options policy-statement bgp-to-ospf from route-filter 10.10.0.0/16 orlonger
set policy-options policy-statement bgp-to-ospf then accept
set policy-options community VPN-A members target:65000:1
set routing-instances VRF1 instance-type vrf
set routing-instances VRF1 interface ge-0/0/4.0
set routing-instances VRF1 route-distinguisher 192.168.146.181:1
set routing-instances VRF1 vrf-import VPN-A-import
set routing-instances VRF1 vrf-export VPN-A-export
set routing-instances VRF1 vrf-table-label
```

```
set routing-instances VRF1 protocols ospf export bgp-to-ospf
set routing-instances VRF1 protocols ospf area 0.0.0.0 interface ge-0/0/4.0
```

CE1

```
set interfaces ge-0/0/0 unit 0 family inet address 10.10.17.7/24
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
```

CE2

```
set interfaces ge-0/0/4 unit 0 family inet address 10.10.56.6/24
set protocols ospf area 0.0.0.0 interface ge-0/0/4.0
```

Configuring Device PE1

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure Device PE1:

1. Configure the interfaces.

```
[edit interfaces]
set ge-0/0/0 unit 0 family inet address 10.10.12.1/24
set ge-0/0/0 unit 0 family mpls maximum-labels 5
set ge-0/0/1 unit 0 family inet address 10.10.13.1/24
set ge-0/0/1 unit 0 family mpls maximum-labels 5
set ge-0/0/5 unit 0 family inet address 10.10.17.1/24
```

2. Configure autonomous system number and options to control packet forwarding routing options.

```
[edit routing-options]
set autonomous-system 65000
set forwarding-table export load-balance-policy
set forwarding-table chained-composite-next-hop ingress l3vpn
```

3. Configure the interfaces with the MPLS protocol and configure the MPLS label range.

```
[edit protocols mpls]
set interface ge-0/0/0.0
set interface ge-0/0/1.0
set label-range static-label-range 1000000 1000999
```

4. Configure the type of peer group, local address, protocol family for NLRIs in updates, and IP address of a neighbor for the peer group.

```
[edit protocols bgp]
set group pe type internal
set group pe local-address 192.168.147.211
set group pe family inet-vpn unicast
set group pe neighbor 192.168.146.181
```

5. Configure the protocol area interfaces.

```
[edit protocols ospf]
set area 0.0.0.0 interface ge-0/0/0.0
set area 0.0.0.0 interface lo0.0
set area 0.0.0.0 interface ge-0/0/1.0
```

6. Configure the IPv4 address and labels of primary and secondary paths for source routing-traffic engineering (TE) policies of protocol source packet routing (SPRING).

```
[edit protocols source-packet-routing segment-list]
set sl-15-primary hop-1 ip-address 10.10.13.3
set sl-15-primary hop-2 label 1000134
set sl-15-primary hop-3 label 1000145
set sl-15-backup hop-1 ip-address 10.10.12.2
set sl-15-backup hop-2 label 1000123
set sl-15-backup hop-3 label 1000134
set sl-15-backup hop-4 label 1000145
```

7. Configure destination IPv4 address, binding SID label, primary, and secondary source routing path for protocol SPRING.

```
[edit protocols source-packet-routing source-routing-path]
set lsp-15 to 192.168.146.181
set lsp-15 binding-sid 1000999
set lsp-15 primary sl-15-primary
set lsp-15 secondary sl-15-backup
```

8. Configure policy options.

```
[edit policy-options policy-statement]
set VPN-A-export term a from protocol ospf
set VPN-A-export term a from protocol direct
set VPN-A-export term a then community add VPN-A
set VPN-A-export term a then accept
set VPN-A-export term b then reject
set VPN-A-import term a from protocol bgp
set VPN-A-import term a from community VPN-A
set VPN-A-import term a then accept
set VPN-A-import term b then reject
set bgp-to-ospf from protocol bgp
set bgp-to-ospf from route-filter 10.10.0.0/16 orlonger
set bgp-to-ospf then accept
set load-balance-policy then load-balance per-packet
```

9. Configure BGP community information.

```
[edit policy-options]
set community VPN-A members target:65000:1
```

10. Configure routing instance VRF1 with instance type, interface, router distinguisher, VRF import, export and table label. Configure export policy and interface of area for protocol OSPF.

```
[edit routing-instances VRF1]
set instance-type vrf
set interface ge-0/0/5.0
set route-distinguisher 192.168.147.211:1
set vrf-import VPN-A-import
```

```
set vrf-export VPN-A-export
set vrf-table-label
set protocols ospf export bgp-to-ospf
set protocols ospf area 0.0.0.0 interface ge-0/0/5.0
```

Results

From configuration mode, confirm your configuration by entering the **show interfaces**, **show policy-options**, **show protocols**, **show routing-options**, and **show routing-instances** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@PE1# show
   interfaces {
    ge-0/0/0 {
        unit 0 {
            family inet {
                address 10.10.12.1/24;
            }
            family mpls {
                maximum-labels 5;
            }
        }
    }
    ge-0/0/1 {
        unit 0 {
            family inet {
                address 10.10.13.1/24;
            }
            family mpls {
                maximum-labels 5;
            }
        }
    }
    ge-0/0/5 {
        unit 0 {
            family inet {
                address 10.10.17.1/24;
            }
        }
```

```
    }
}
policy-options {
    policy-statement VPN-A-export {
        term a {
            from protocol [ ospf direct ];
            then {
                community add VPN-A;
                accept;
            }
        }
        term b {
            then reject;
        }
    }
    policy-statement VPN-A-import {
        term a {
            from {
                protocol bgp;
                community VPN-A;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    policy-statement bgp-to-ospf {
        from {
            protocol bgp;
            route-filter 10.10.0.0/16 orlonger;
        }
        then accept;
    }
    policy-statement load-balance-policy {
        then {
            load-balance per-packet;
        }
    }
    community VPN-A members target:65000:1;
}
routing-instances {
    VRF1 {
```

```
        instance-type vrf;
        protocols {
            ospf {
                area 0.0.0.0 {
                    interface ge-0/0/5.0;
                }
                export bgp-to-ospf;
            }
        }
        interface ge-0/0/5.0;
        route-distinguisher 192.168.147.211:1;
        vrf-import VPN-A-import;
        vrf-export VPN-A-export;
        vrf-table-label;
    }
}
routing-options {
    autonomous-system 65000;
    forwarding-table {
        export load-balance-policy;
        chained-composite-next-hop {
            ingress {
                l3vpn;
            }
        }
    }
}
protocols {
    bgp {
        group pe {
            type internal;
            local-address 192.168.147.211;
            family inet-vpn {
                unicast;
            }
            neighbor 192.168.146.181;
        }
    }
    mpls {
        label-range {
            static-label-range 1000000 1000999;
        }
        interface ge-0/0/0.0;
```

```
        interface ge-0/0/1.0;
    }
    ospf {
        area 0.0.0.0 {
            interface ge-0/0/0.0;
            interface lo0.0;
            interface ge-0/0/1.0;
        }
    }
    source-packet-routing {
        segment-list sl-15-primary {
            hop-1 ip-address 10.10.13.3;
            hop-2 label 1000134;
            hop-3 label 1000145;
        }
        segment-list sl-15-backup {
            hop-1 ip-address 10.10.12.2;
            hop-2 label 1000123;
            hop-3 label 1000134;
            hop-4 label 1000145;
        }
        source-routing-path lsp-15 {
            to 192.168.146.181;
            binding-sid 1000999;
            primary {
                sl-15-primary;
            }
            secondary {
                sl-15-backup;
            }
        }
    }
}
```

Configuring Device PE2

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

1. Configure the interfaces.

```
[edit interfaces]
set ge-0/0/0 unit 0 family inet address 10.10.12.2/24
set ge-0/0/0 unit 0 family mpls
set ge-0/0/1 unit 0 family inet address 10.10.23.2/24
set ge-0/0/1 unit 0 family mpls
```

2. Configure the static LSP for protocol MPLS.

```
[edit protocols mpls static-label-switched-path]
set adj-23 segment 1000123
set adj-23 segment next-hop 10.10.23.3
set adj-23 segment pop
set adj-21 segment 1000221
set adj-21 segment next-hop 10.10.12.1
set adj-21 segment pop
```

3. Configure interfaces and static label range for protocol MPLS.

```
[edit protocols mpls]
set interface ge-0/0/0.0
set interface ge-0/0/1.0
set label-range static-label-range 1000000 1000999
```

4. Configure interfaces for protocol OSPF.

```
[edit protocols ospf area 0.0.0.0]
set interface ge-0/0/0.0
set interface ge-0/0/1.0
```

Results

From configuration mode on router PE2, confirm your configuration by entering the **show interfaces** and **show protocols** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@PE2# show
interfaces {
    ge-0/0/0 {
        unit 0 {
            family inet {
                address 10.10.12.2/24;
            }
            family mpls;
        }
    }
    ge-0/0/1 {
        unit 0 {
            family inet {
                address 10.10.23.2/24;
            }
            family mpls;
        }
    }
}
protocols {
    mpls {
        label-range {
            static-label-range 1000000 1000999;
        }
        interface ge-0/0/0.0;
        interface ge-0/0/1.0;
        static-label-switched-path adj-23 {
            segment {
                1000123;
                next-hop 10.10.23.3;
                pop;
            }
        }
        static-label-switched-path adj-21 {
            segment {
```

```
                1000221;
                next-hop 10.10.12.1;
                pop;
            }
        }
    }
    ospf {
        area 0.0.0.0 {
            interface ge-0/0/0.0;
            interface ge-0/0/1.0;
        }
    }
}
```

Verification

IN THIS SECTION

- Verifying Route Entry of Routing Table inet.3 of Router PE1 | **875**
- Verifying Route Table Entries of Routing Table mpls.0 of Router PE1 | **876**
- Verifying SPRING Traffic Engineered LSP of Router PE1 | **877**
- Verifying SPRING Traffic Engineered LSPs on the Ingress Router of Router PE1 | **877**
- Verifying the Routing Table Entries of Routing Table mpls.0 of Router PE2 | **879**
- Verifying the Status of Static MPLS LSP Segments of Router PE2 | **879**

Confirm that the configuration is working properly.

Verifying Route Entry of Routing Table inet.3 of Router PE1

Purpose

Verify the route entry of routing table inet.3 of router PE1.

Action

From operational mode, enter the `show route table inet.3` command.

```
user@PE1> show route table inet.3
inet.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.146.181/32   *[SPRING-TE/8] 03:09:26, metric 1
                     > to 10.10.13.3 via ge-0/0/1.0, Push 1000145, Push 1000134(top)
                       to 10.10.12.2 via ge-0/0/0.0, Push 1000145, Push 1000134, Push 1000123(top)
```

Meaning

The output displays the ingress routes of segment routing tunnels.

Verifying Route Table Entries of Routing Table mpls.0 of Router PE1

Purpose

Verify the route entries of routing table mpls.0

Action

From operational mode, enter the `show route table mpls.0` command.

```
user@PE1> show route table mpls.0
mpls.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 03:25:52, metric 1
                      Receive
1                  *[MPLS/0] 03:25:52, metric 1
                      Receive
2                  *[MPLS/0] 03:25:52, metric 1
                      Receive
13                 *[MPLS/0] 03:25:52, metric 1
                      Receive
16                 *[VPN/0] 03:25:52
                    > via lsi.0 (VRF1), Pop
```

```
1000999            *[SPRING-TE/8] 03:04:03, metric 1
                    > to 10.10.13.3 via ge-0/0/1.0, Swap 1000145, Push 1000134(top)
                      to 10.10.12.2 via ge-0/0/0.0, Swap 1000145, Push 1000134, Push 1000123(top)
```

Meaning

The output displays the SID labels of segment routing tunnels.

Verifying SPRING Traffic Engineered LSP of Router PE1

Purpose

Verify SPRING traffic engineered LSPs on the ingress routers.

Action

From operational mode, enter the `show spring-traffic-engineering overview` command.

```
user@PE1> show spring-traffic-engineering overview
Overview of SPRING-TE:
  Route preference: 8
  Number of LSPs: 1 (Up: 1, Down: 0)
  External controllers:
            < Not configured >
```

Meaning

The output displays the overview of SPRING traffic engineered LSPs on the ingress router.

Verifying SPRING Traffic Engineered LSPs on the Ingress Router of Router PE1

Purpose

Verify SPRING traffic engineered LSPs on the ingress router.

Action

From operational mode, enter the `show spring-traffic-engineering lsp detail` command.

```
user@PE1# show spring-traffic-engineering lsp detail
Name: lsp-15
To: 192.168.146.181
State: Up
  Path: sl-15-primary
  Outgoing interface: ge-0/0/1.0
  BFD status: N/A (Up: 0, Down: 0)
  SR-ERO hop count: 3
    Hop 1 (Strict):
      NAI: IPv4 Adjacency ID, 0.0.0.0 -> 10.10.13.3
      SID type: None
    Hop 2 (Strict):
      NAI: None
      SID type: 20-bit label, Value: 1000134
    Hop 3 (Strict):
      NAI: None
      SID type: 20-bit label, Value: 1000145
  Path: sl-15-backup
  Outgoing interface: ge-0/0/0.0
  BFD status: N/A (Up: 0, Down: 0)
  SR-ERO hop count: 4
    Hop 1 (Strict):
      NAI: IPv4 Adjacency ID, 0.0.0.0 -> 10.10.12.2
      SID type: None
    Hop 2 (Strict):
      NAI: None
      SID type: 20-bit label, Value: 1000123
    Hop 3 (Strict):
      NAI: None
      SID type: 20-bit label, Value: 1000134
    Hop 4 (Strict):
      NAI: None
      SID type: 20-bit label, Value: 1000145


Total displayed LSPs: 1 (Up: 1, Down: 0)
```

Meaning

The output displays details of SPRING traffic engineered LSPs on the ingress router

Verifying the Routing Table Entries of Routing Table mpls.0 of Router PE2

Purpose

Verify the routing table entries of routing table mpls.0 of router PE2.

Action

From operational mode, enter the `show route table mpls.0` command.

```
user@PE2> show route table mpls.0
mpls.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 03:22:29, metric 1
                      Receive
1                  *[MPLS/0] 03:22:29, metric 1
                      Receive
2                  *[MPLS/0] 03:22:29, metric 1
                      Receive
13                 *[MPLS/0] 03:22:29, metric 1
                      Receive
1000123            *[MPLS/6] 03:22:29, metric 1
                    > to 10.10.23.3 via ge-0/0/1.0, Pop
1000123(S=0)       *[MPLS/6] 03:22:29, metric 1
                    > to 10.10.23.3 via ge-0/0/1.0, Pop
1000221            *[MPLS/6] 03:22:29, metric 1
                    > to 10.10.12.1 via ge-0/0/0.0, Pop
1000221(S=0)       *[MPLS/6] 03:22:29, metric 1
                    > to 10.10.12.1 via ge-0/0/0.0, Pop
```

Verifying the Status of Static MPLS LSP Segments of Router PE2

Purpose

Verify the status of MPLS LSP segments of router PE2.

Action

From operational mode, enter the `show mpls static-lsp` command.

```
user@PE2> show mpls static-lsp
Ingress LSPs:
Total 0, displayed 0, Up 0, Down 0

Transit LSPs:
Total 0, displayed 0, Up 0, Down 0

Bypass LSPs:
Total 0, displayed 0, Up 0, Down 0

Segment LSPs:
LSPname                              SID-label       State
adj-21                               1000221         Up
adj-23                               1000123         Up
Total 2, displayed 2, Up 2, Down 0
```

Meaning

The output displays the status of static MPLS LSP segments of router PE2.

Routing Engine-based S-BFD for Segment-Routing Traffic Engineering with First-Hop Label Resolution

IN THIS SECTION

- Understanding RE-based S-BFD for Segment-Routing Traffic Engineering with First-Hop Label Resolution | **881**
- S-BFD for SRv6 TE Paths | **882**
- Configuring RE-based S-BFD for Segment-Routing Traffic Engineering with First-Hop Label Resolution | **883**
- Auto Derivation of Remote Discriminator (RD) for S-BFD Session | **886**
- Example: Configuring Seamless Bidirectional Forwarding Detection (S-BFD) for SR-TE | **887**

You can run seamless Bidirectional Forwarding Detection (S-BFD) over non-colored and colored label-switched paths (LSPs) with first-hop label resolution, using S-BFD as a fast mechanism to detect path failures.

Understanding RE-based S-BFD for Segment-Routing Traffic Engineering with First-Hop Label Resolution

IN THIS SECTION

- S-BFD Static Segment-Routing Tunnels for First-Hop Labels | **881**
- Limitations | **882**

S-BFD Static Segment-Routing Tunnels for First-Hop Labels

Segment-routing architecture enables ingress nodes in a core network to steer traffic through explicit paths through the network. The segment-routing traffic engineering (TE) next hop is a list or lists of segment identifiers (SIDs). These segment lists represent paths in the network that you want incoming traffic to take. The incoming traffic may be labeled or IP traffic and the forwarding operation at the ingress node may be a label swap or a destination-based lookup to steer the traffic onto these segment-routing TE paths in the forwarding path.

You can run seamless BFD (S-BFD) over non-colored and colored static segment-routing LSPs with first-hop label resolution and use S-BFD as a fast mechanism to detect path failures and to trigger global convergence. S-BFD requires fewer state changes than BFD requires, thus speeding up the reporting of path failures.

Given a segment-routing tunnel with one or multiple primary LSPs and optionally a secondary LSP, you can enable S-BFD on any of those LSPs. If an S-BFD session goes down, the software updates the segment-routing tunnel's route by deleting the next hops of the failed LSPs. If the first-hop label of the LSP points to more than one immediate next hop, the kernel continues to send S-BFD packets if *at least* one next hop is available (underlying next-hop reachability failure detection must be faster than the duration of the S-BFD detection timer).

NOTE: This model is supported for auto-translate-derived LSPs.

LSP-level S-BFD

An S-BFD session is created for each unique label-stack+address-family combination. You can configure identical segment lists and enable S-BFD for all of them. The segment lists that have identical label-stack

+address-family combinations share the same S-BFD session. The source address for the S-BFD session is set to the least configured loopback address (except the special addresses) under the main instance.

NOTE: Ensure that the chosen source address is routable.

The address family of an LSP is derived based on the address family of the “to” address in the segment-routing TE tunnel. The state of the LSP with S-BFD configured also depends on whether BFD is up—if S-BFD is configured for an LSP, the LSP route isn’t calculated until S-BFD reports the path is alive.

S-BFD Parameters

The following S-BFD parameters are supported for segment-routing TE paths:

- Remote discriminator
- Minimum interval
- Multiplier
- No router alert option

In S-BFD, each responder may have multiple discriminators. The discriminators may be advertised by IGP to other routers, or they may be statically configured on these routers. On an initiator, a particular discriminator is chosen as the remote discriminator for an S-BFD session by static configuration, based on the decision or resolution made by you or by a central controller. When multiple LSPs are created with the same key label stack and S-BFD is enabled on each of them with different parameters, the aggressive value of each parameter is used in the shared S-BFD session. For the discriminator parameter, the lowest value is considered as most aggressive. Similarly for the router alert option, if one of the configurations no router alert is configured, the derived S-BFD parameter will have no router alert option.

Limitations

- Global and local repair is supported for MX Series devices.
- Even though S-BFD detects failures depending on the configured timer values, convergence time depends on the global repair time (*seconds*).

S-BFD for SRv6 TE Paths

You can run S-BFD over SRv6 TE paths to quickly detect path failures. Each path configured with S-BFD within a SRv6 TE tunnel can send probes to the destination of the path. These probes follow the SIDs of the TE path and report failures for any SIDs within the path. When failues are detected, the

corresponding SRv6 TE tunnel path will be brought down so traffic can be distributed onto backup paths.

S-BFD for SRv6 is supported in distributed mode on ingress routers and distributed mode on egress routers.

To configure S-BFD for a SRv6 TE path on an ingress router, you must configure a local discriminator with the `sbfd local-discriminator` *`number`* configuration statement at the `[edit protocols bfd]` hierarchy level. You also need to configure a remote discriminator with the `sbfd remote-discriminator` *`number`* configuration statement at the `[edit protocols source-packet-routing source-routing-path` *`name`* `primary` *`name`* `bfd-liveness-detection]` hierarchy level.

To configure S-BFD for SRv6 TE paths on an egress router, you must configure the `sbfd local-discriminator` *`number`* `local-ipv6-address` *`address`* configuration statement at the `[edit protocols bfd]` hierachy level. The `loca-discriminator` at the responder must match the `remote-discriminator` configured on the SRv6 TE path at the ingress router

For S-BFD responders that only supports IPv6 local host address, you can enforce the use of an IPv6 local host address by using the `bfd-liveness-detection sbfd destination-ipv6-local-host` configuration statement at the `[edit protocols source-packet-routing source-routing-path` *`lsp-path-name`* `primary` *`segment-list-name`*`]` hierarchy level.

Configuring RE-based S-BFD for Segment-Routing Traffic Engineering with First-Hop Label Resolution

To enable LSP-level S-BFD for a segment list, you configure the `bfd-liveness-detection` configuration statement at the `[edit protocols source-packet-routing source-routing-path` *`lsp-path-name`* `primary` *`segment-list-name`*`]` hierarchy and the `[edit protocols source-packet-routing source-routing-path` *`lsp-path-name`* `secondary` *`segment-list-name`*`]` hierarchy.

The following steps give the basic configuration and then operation of S-BFD with first-hop label resolution:

- The steps immediately below describe the outlines of the basic *configuration*:

  1. On an ingress router, you configure one or more segment lists with first-hop labels for a static segment-routing tunnel to use as primary and secondary paths.

  2. On the ingress router, you configure the static segment-routing tunnel, with either multiple primary paths (for load balancing), or one primary path and one secondary path (for path protection). Each primary or secondary path (LSP) refers to one of the segment lists you configured already, creating routes using the next hops derived from the first-hop labels from contributing LSPs.

  3. On the ingress router, you enable per-packet load-balancing so that routes resolving over ingress routes and the binding-SID route (which all have first-hop labels) install all active paths in the Packet Forwarding Engine. An S-BFD session under an LSP protects all routes that use that LSP.

4. On the egress router of the segment-routing tunnel, you configure S-BFD responder mode with a local discriminator D, creating a distributed S-BFD listener session for D on each FPC.

5. On the ingress router, you configure S-BFD for any LSP for which you want to see path-failure detection. You specify a remote-discriminator D to match the local S-BFD discriminator of the egress router. An S-BFD initiator session is created with the LSP label-stack+address-family combination as the key, if an initiator session doesn't already exist for the current LSP key. The S-BFD parameters in the case of a matching BFD session are reevaluated with the new shared LSPs taken into consideration. When the S-BFD parameters are derived, the aggressive value of each parameter is chosen.

The steps immediately below describe basic *operation* :

1. The S-BFD initiator session runs in a centralized mode on the Routing Engine. The software tracks S-BFD session up and down states.

2. When the S-BFD state moves to UP, the LSP is considered for the relevant routes.

3. On the ingress router, if the software detects an S-BFD session DOWN, a session-down notification is sent to the routing daemon (RPD), and RPD deletes the next hop of the failed LSPs from the segment-routing tunnel's route.

4. The total traffic loss in the procedure is bound to the S-BFD failure detection time and the global repair time. The S-BFD failure detection time is determined by the S-BFD minimum-interval and multiplier parameters. The global repair time depends on the segment-routing TE process time and the redownload of the routes to forwarding.

LSP label stacks are changed as follows:

1. The particular LSP is detached from the existing S-BFD session. If the existing S-BFD session has at least one LSP referring to it, the old BFD session is preserved, but the S-BFD parameters are re-evaluated so that the aggressive parameter values among the existing LSP sessions is chosen. Also, the name of the S-BFD session is updated to the least one if there is a change. If the old S-BFD session has no more LSPs attached to it, that S-BFD session is removed.

2. The software attempts to find an existing BFD session that matches the new-label-stack+address-family combination value; if such a match exists, the LSP refers to that existing S-BFD session. The S-BFD session is re-evaluated for any change in parameters or session name similarly to the re-evaluations in step 1.

3. If there is no matching BFD session in the system, a new BFD session is created, and the S-BFD parameters are derived from this LSP.

NOTE: An S-BFD session's minimum interval and multiplier determine the failure detection time for the session. The repair time additionally depends on the global repair time.

The following output shows configuration statements you would use for a colored LSP with primary LSPs:

```
[edit protocols]
source-packet-routing {
    source-routing-path lsp_name {
        to destination_address;
        color color_value;
        binding-sid binding-label;
        primary segment_list_1_name weight weight;
            ... {
            bfd-liveness-detection {
                sbfd {
                    remote-discriminator value;
                }
            }
        }
    }
}
```

At the responder side, you must configure the correct discriminator:

```
[edit protocols bfd]
sbfd {
    local-discriminator value;
}
```

By default, router alerts are configured for S-BFD packets. When the MPLS header is removed at the responder end, the packet is sent to the host for processing without a destination address lookup for the packet. If you enable the no-router-alert option on the ingress router, the router-alert option is removed

from the IP options and hence from the egress side. You must configure the destination address explicitly in lo0; otherwise the packet is discarded, and S-BFD remains down.

```
[edit interfaces lo0 unit 0 family inet]
address 127.0.0.1/32;
```

You can use a new trace flag, `bfd`, to trace BFD activities:

```
user@host# set protocols source-packet-routing traceoptions flag bfd
```

Auto Derivation of Remote Discriminator (RD) for S-BFD Session

IN THIS SECTION

- Overview | 886

Overview

You can use the auto-derived remote discriminator for S-BFD sessions for the SR-TE paths. With this feature, you need not configure a `remote-discriminator` in the S-BFD configuration on the ingress or transit device and a matching local-discriminator on its respective endpoint. Instead, the egress PE device will now accept IP addresses as its local discriminator. To allow IP address as local-discriminator in BFD, use the `set protocols bfd sbfd local-discriminator-ip` configuration.

You can also use a common S-BFD template with the S-BFD configurations on multiple controller provisioned SR-TE policies. In these S-BFD sessions, Junos OS automatically derives the remote discriminator from the tunnel endpoint for matching SR-TE policies.

NOTE:

- We support auto derivation of RD only for IPv4 endpoints, not for IPv6 endpoints.
- We do not support auto derivation of RD for color-only tunnels. If RD is not configured (by auto derivation of RD) for statically configured SR-TE color-only tunnels, the system will display a commit error. If RD is not configured (by auto

derivation of RD) for dynamic color-only tunnels by using SR-TE template configuration, Junos OS skips the `sbfd` configuration for that tunnel.

Example: Configuring Seamless Bidirectional Forwarding Detection (S-BFD) for SR-TE

SUMMARY

Configure S-BFD for segment routing paths to provide fast, reliable failure detection in the forwarding plane , enabling fast reroute (FRR) mechanisms in a segment routing network. You can monitor segment routing policies and explicitly defined segment routing paths, to ensure traffic engineering constraints are met and paths are operational.

IN THIS SECTION

- Example Prerequisites | **887**
- Before You Begin | **888**
- Functional Overview | **888**
- Topology Overview | **889**
- Topology Illustration | **890**
- Configure S-BFD over SR-TE on R0 | **890**
- Verification | **894**
- Appendix 1: Set Commands on All Devices | **900**
- Appendix 2: Show Configuration Output on R0 | **907**

TIP:

Table 23: Readability Score and Time Estimates

Readability Score	• Flesch reading ease: 34 • Flesch-Kincaid reading grade level: 11.9
Reading Time	Less than 15 minutes.
Configuration Time	Less than an hour.

Example Prerequisites

Table 24: Prerequisites for configuring S-BFD for SR-TE

Hardware requirements	PTX Series Routers and MX Series Routers
Software requirements	Junos OS Release 23.1R1 or later

Before You Begin

Table 25: Before You Begin

Benefits	**Simple configuration**-The support for auto-derived remote discriminator (RD) from tunnel endpoint address makes the S-BFD configuration user friendly. **Scalable option**-You can apply a common S-BFD template on multiple controller-provisioned SR-TE policies with auto-derived RD is use. **Minimal traffic diruption**-The one-way path monitoring is ideal for asymmetric segement routing paths, simplifying the monitoring architecture for detecting LSP failures. **Less overhead**-S-BFD doe not create a separate BFD session for each hop, reducing overhead.
Know more	Routing Engine-based S-BFD for Segment-Routing Traffic Engineering with First-Hop Label Resolution
Hands-on experience	vLab Sandbox: Segment Routing - Basic
Learn more	Junos Segment Routing for MPLS

Functional Overview

This table provides a quick summary of the configuration components deployed in this example.

Table 26: Functional Overview

Policies	
prefix-sid	Defines the segment IDs (SIDs) to reach the egress device, R3.
Protocols	
IS-IS	IS-IS is enabled on all the devices.

Source Packet Routing	There are two segment routing paths, `srpath1` and `srpath2`, on which seamless BFD (S-BFD) is configured. There is a segment routing LSP (srlsp1) between R0 and R3. Device R3 has the BFD local discriminator configured on it.
MPLS	MPLS is enabled on all the devices.
Primary verification tasks	• Verify the SR-TE LSP is up. • Verify the S-BFD session is up with auto-derived remote discriminator (RD) value.

Topology Overview

In this example, the ingress device (R0) initiates a seamless BFD (S-BFD) session over segment routing paths to the egress device (R3). The segment routing path is defined using segment lists (SIDs) that specify the hops. Device R3 responds with a plain IPv4 packet that is encrypted with the SID information that is automatically generated as auto-RD.

Table 27: Topology Overview

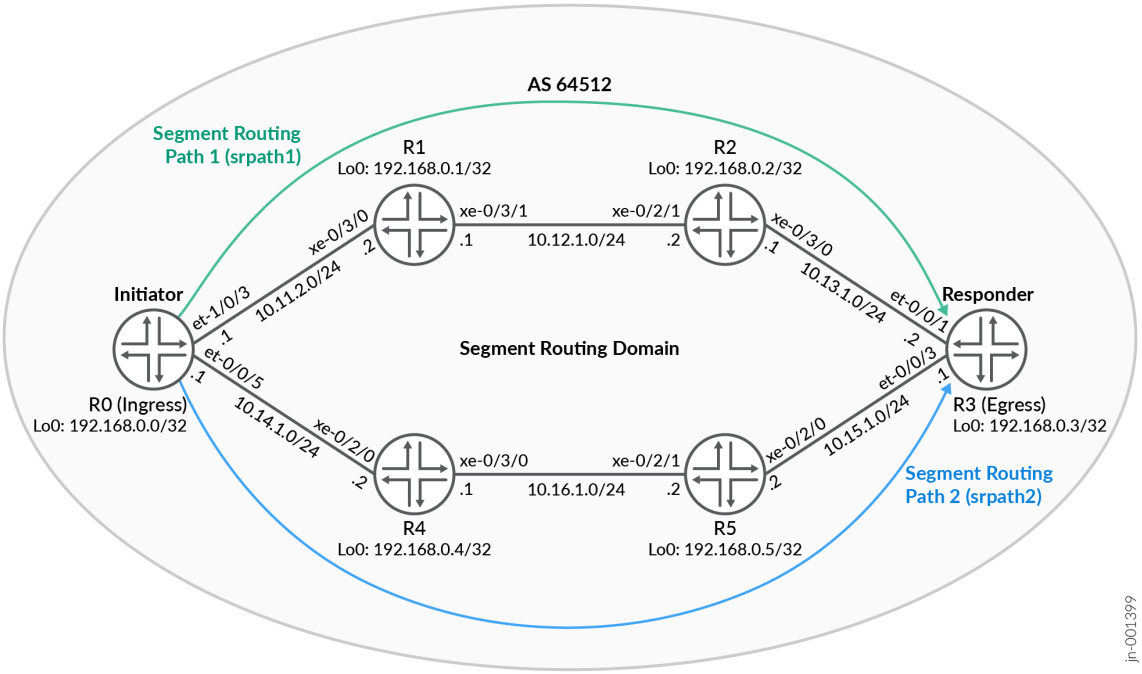
Hostname	Role	Function
R0	Ingress Device (Initiator)	R0 initiates the BFD session over the segment routing path to Device R3.
R1	Transit Device	One of the segment in the first segment routing path (srpath1) from R0 to R3.
R2	Transit Device	One of the segment in the first segment routing path (srpath1) from R0 to R3.
R3	Egress Device (Responder)	Responds with a plain IPv4 packet that is encrypted with the SID information that is automatically generated as auto-RD.

Table 27: Topology Overview *(Continued)*

Hostname	Role	Function
R4	Transit Device	One of the segment in the second segment routing path (srpath2) from R0 to R3.
R5	Transit Device	One of the segment in the second segment routing path (srpath2) from R0 to R3.

Topology Illustration

Figure 58: Auto-Derivation of Remote Discriminator (auto-RD) for S-BFD session in SR-TE



Configure S-BFD over SR-TE on R0

NOTE: For complete sample configurations on the DUT, see:

- "Appendix 1: Set Commands on All Devices" on page 900
- "Appendix 2: Show Configuration Output on R0" on page 907

1. Configure the device interfaces.

```
[edit interfaces]
user@R0# set et-0/0/5 unit 0 description "CONNECTED TO R4"
user@R0# set et-0/0/5 unit 0 family inet address 10.14.1.1/24
user@R0# set et-0/0/5 unit 0 family iso
user@R0# set et-0/0/5 unit 0 family mpls
```

```
user@R0# set et-1/0/3 unit 0 description "CONNECTED TO R1"
user@R0# set et-1/0/3 unit 0 family inet address 10.11.2.1/24
user@R0# set et-1/0/3 unit 0 family iso
user@R0# set et-1/0/3 unit 0 family mpls
```

```
user@R0# set lo0 unit 0 family inet address 192.168.0.0/32
user@R0# set lo0 unit 0 family iso address 49.0001.000a.0a0a.0a00
```

2. Configure the device router-ID and assign it to an autonomus system.

```
[edit routing-options]
user@R0# set router-id 192.168.0.0
user@R0# set autonomous-system 64512
```

3. Configure the `test` RIB group to learn IPv4 and MPLS routes, and apply it to protocols IS-IS.

```
[edit routing-options]
user@R0# set rib-groups test import-rib inet.0
user@R0# set rib-groups test import-rib inet.3
```

```
[edit protocols]
user@R0# set isis rib-group inet test
```

4. Define the `prefix-sid` policy to accept node segment ID. Export the configured policy into protocols IS-IS.

```
[edit policy-options]
user@R0# set policy-statement prefix-sid term 1 from route-filter 192.168.0.0/32 exact
user@R0# set policy-statement prefix-sid term 1 then prefix-segment index 100
user@R0# set policy-statement prefix-sid term 1 then prefix-segment node-segment
user@R0# set policy-statement prefix-sid term 1 then accept
```

```
[edit protocols]
user@R0# set isis export prefix-sid
```

5. Enable IS-IS on the device interfaces excluding the loopback and management interfaces.

```
[edit protocols]
user@R0# set isis interface et-0/0/5.0 level 2 metric 20
user@R0# set isis interface et-0/0/5.0 node-link-protection
user@R0# set isis interface et-0/0/5.0 point-to-point
user@R0# set isis interface et-1/0/3.0 level 2 metric 20
user@R0# set isis interface et-1/0/3.0 node-link-protection
user@R0# set isis interface et-1/0/3.0 point-to-point
user@R0# set isis interface all ldp-synchronization
user@R0# set isis interface all node-link-protection
user@R0# set isis interface fxp0.0 disable
user@R0# set isis interface lo0.0 disable
```

6. Configure the other IS-IS paramters.

```
[edit protocols]
user@R0# set isis level 1 disable
user@R0# set isis level 2 wide-metrics-only
user@R0# set isis level 2 preference 180
user@R0# set isis backup-spf-options remote-backup-calculation
user@R0# set isis backup-spf-options use-source-packet-routing
user@R0# set isis traffic-engineering l3-unicast-topology
user@R0# set isis traffic-engineering family inet shortcuts
user@R0# set isis ignore-attached-bit
```

7. Configure source packet routing parameters under IS-IS.

```
[edit protocols]
user@R0# set isis source-packet-routing adjacency-segment hold-time 240000
user@R0# set isis source-packet-routing srgb start-label 800000
user@R0# set isis source-packet-routing srgb index-range 80000
user@R0# set isis source-packet-routing explicit-null
user@R0# set isis source-packet-routing traffic-statistics statistics-granularity per-
interface
```

8. Enable MPLS on the device interfaces, excluding the management interface.

```
[edit protocols]
user@R0# set mpls interface all
user@R0# set mpls interface fxp0.0 disable
```

9. Configure the first segment routing path, `srpath1`, to reach Device R3. The node segments of this path traverse through Device R1 and Device R2 to reach R3.

```
[edit protcols]
user@R0# set source-packet-routing segment-list srpath1 auto-translate unprotected
user@R0# set source-packet-routing segment-list srpath1 hop1 ip-address 10.11.2.2
user@R0# set source-packet-routing segment-list srpath1 hop2 ip-address 192.168.0.2
user@R0# set source-packet-routing segment-list srpath1 hop2 label-type node
user@R0# set source-packet-routing segment-list srpath1 hop3 ip-address 192.168.0.3
user@R0# set source-packet-routing segment-list srpath1 hop3 label-type node
```

10. Configure the second segment routing path, `srpath2`, to reach Device R3. The node segments of this path traverse through Device R4 and Device R5 to reach R3.

```
[edit protocols]
user@R0# set source-packet-routing segment-list srpath2 auto-translate unprotected
user@R0# set source-packet-routing segment-list srpath2 hop1 ip-address 10.14.1.2
```

11. Configure the segment routing label-switched path (LSP) to Device R3.

```
[edit protocols]
user@R0# set source-packet-routing source-routing-path srlsp1 to 192.168.0.3
user@R0# set source-packet-routing source-routing-path srlsp1 primary srpath1 weight 23
user@R0# set source-packet-routing source-routing-path srlsp1 primary srpath1 bfd-liveness-
```

```
detection sbfd
user@R0# set source-packet-routing source-routing-path srlsp1 primary srpath1 bfd-liveness-
detection minimum-interval 1000
user@R0# set source-packet-routing source-routing-path srlsp1 primary srpath2 weight 45
user@R0# set source-packet-routing source-routing-path srlsp1 primary srpath2 bfd-liveness-
detection sbfd
user@R0# set source-packet-routing source-routing-path srlsp1 primary srpath2 bfd-liveness-
detection minimum-interval 1000
```

NOTE: At the responder side (Device R3), you must configure the correct discriminator:

```
[edit protocols]
user@R3# set bfd sbfd local-discriminator-ip 192.168.0.3
```

12. Configure source-packet-routing to inherit label information from segment list.

```
[edit protocols]
user@R0# set source-packet-routing inherit-label-nexthops
```

Verification

IN THIS SECTION

- BFD Session Verification | **895**
- LSP Path Verification | **896**
- S-BFD Session Verification on Device R0 | **898**
- S-BFD Session Verification on Device R3 | **899**

Use the list of show commands to verify the feature in this example.

Table 28: Verification Commands

Command	Verification Task
show bfd session	Verify that the BFD session is up.

Table 28: Verification Commands *(Continued)*

Command	Verification Task
show spring-traffic-engineering	Verify the segment routing LSP using the `lsp` command option and the auto derived RD value using the `sbfd` command option.
show bfd seamless session	Verify the S-BFD session on Device R3.

BFD Session Verification

Purpose

Verify the status of the BFD session.

Action

From operational mode, enter the **show bfd session extensive** command to view the status of the BFD session.

```
user@R0> show bfd session extensive
                                                Detect   Transmit
Address                 State     Interface     Time     Interval  Multiplier
127.0.0.1               Up                    0.000     2.000        3
 Client SPRING-TE, TX interval 1.000, RX interval 1.000
 Local diagnostic None, remote diagnostic None
 Remote state AdminDown, version 1
 Replicated
 Session type: Multi hop BFD (Seamless)
 Min async interval 1.000, min slow interval 2.000
 Adaptive async TX interval 2.000, RX interval 2.000
 Local min TX interval 2.000, minimum RX interval 1.000, multiplier 3
 Remote min TX interval 0.000, min RX interval 0.000, multiplier 0
 Local discriminator 18, remote discriminator 3232235523
 Echo TX interval 2.000, echo detection interval 6.000
 Echo mode disabled/inactiveno-absorb, no-refresh
 Path-Name V4-srte_bfd_session-4
  Session ID: 0
```

```
                                                Detect   Transmit
Address                 State     Interface     Time     Interval  Multiplier
127.0.0.1               Up                      3.000     1.000        3
 Client SPRING-TE, TX interval 1.000, RX interval 1.000
 Session up time 00:18:16
 Local diagnostic None, remote diagnostic None
 Remote state Up, version 1
 Replicated
 Session type: Multi hop BFD (Seamless)
 Min async interval 1.000, min slow interval 1.000
 Adaptive async TX interval 1.000, RX interval 1.000
 Local min TX interval 1.000, minimum RX interval 1.000, multiplier 3
 Remote min TX interval 1.000, min RX interval 0.000, multiplier 3
 Local discriminator 17, remote discriminator 3232235523
 Echo TX interval 0.000, echo detection interval 0.000
 Echo mode disabled/inactive
 Remote is control-plane independent
 Path-Name V4-srte_bfd_session-3
  Session ID: 0

2 sessions, 2 clients
Cumulative transmit rate 1.5 pps, cumulative receive rate 1.0 pps
```

Meaning

The status of the BFD sessions are up.

The sessions are named as `V4-srte_bfd_session-4` and `V4-srte_bfd_session-3`. Because many LSPs can share the same BFD session, when the first LSP with a unique combination of `label-stack + address-family` comes up, S-BFD uses the `address-family + lsp-name` combination for the session name. If later, more LSPs share the same session, the name of the session can change to `address-family + least-lsp-name`, without affecting the state of the S-BFD session.

LSP Path Verification

Purpose

Verify that the segment routing LSP (srlsp1) is configured and the S-BFD session status is visible. Verify the path that srlsp1 is traversing to reach Device R3.

Action

From the operational mode, enter the **show spring-traffic-engineering lsp name srlsp1 detail** command to understand the path the segment routing LSP is traversing.

```
user@R0> show spring-traffic-engineering lsp name srlsp1 detail
E = Entropy-label Capability

Name: srlsp1
  Tunnel-source: Static configuration
  Tunnel Forward Type: SRMPLS
  To: 192.168.0.3
  Te-group-id: 0
  State: Up
    Path: srpath1
    Path Status: NA
    Outgoing interface: NA
    Auto-translate status: Enabled Auto-translate result: Success
    Compute Status:Disabled , Compute Result:N/A , Compute-Profile Name:N/A
    BFD status: Up BFD name: V4-srte_bfd_session-3
    BFD remote-discriminator: 3232235523(auto-derived)
    Segment ID : 128
    ERO Valid: true
      SR-ERO hop count: 3
        Hop 1 (Strict):
          NAI: IPv4 Adjacency ID, 0.0.0.0 -> 10.11.2.2
          SID type: 20-bit label, Value: 16
        Hop 2 (Loose):
          NAI: IPv4 Node ID, Node address: 192.168.0.2
          SID type: 20-bit label, Value: 801002
        Hop 3 (Loose):
          NAI: IPv4 Node ID, Node address: 192.168.0.3
          SID type: 20-bit label, Value: 801003
    Path: srpath2
    Path Status: NA
    Outgoing interface: NA
    Auto-translate status: Enabled Auto-translate result: Success
    Compute Status:Disabled , Compute Result:N/A , Compute-Profile Name:N/A
    BFD status: Down BFD name: V4-srte_bfd_session-4
    BFD remote-discriminator: 3232235523(auto-derived)
    Segment ID : 256
    ERO Valid: true
```

```
      SR-ERO hop count: 1
        Hop 1 (Strict):
          NAI: IPv4 Adjacency ID, 0.0.0.0 -> 10.14.1.2
          SID type: 20-bit label, Value: 17


Total displayed LSPs: 1 (Up: 1, Down: 0, Initializing: 0)
```

Meaning

Verify the hops the LSP is traversing to reach the last hop, Device R3.

Output for each LSP shows the S-BFD status as well as the S-BFD name it is referring to. Because there might not be one S-BFD session per LSP, the LSP-level S-BFD counters are not displayed.

S-BFD Session Verification on Device R0

Purpose

Verify the status of the S-BFD sessions over the two segment routing paths on Device R0.

Action

From the operational mode, enter the **show spring-traffic-engineering sbfd detail**command to verify the status of the S-BFD session.

```
user@R0> show spring-traffic-engineering sbfd detail
BFD name: V4-srte_bfd_session-4
BFD status: Up
BFD remote-discriminator: 3232235523
BFD local-discriminator: 18
Referencing LSPs:
  srlsp1:srpath2
      SR-ERO hop count: 1
        Hop 1 (Strict):
          NAI: IPv4 Adjacency ID, 0.0.0.0 -> 10.14.1.2
          SID type: 20-bit label, Value: 17

BFD name: V4-srte_bfd_session-3
BFD status: Up
BFD remote-discriminator: 3232235523
```

```
BFD local-discriminator: 17
Referencing LSPs:
  srlsp1:srpath1
      SR-ERO hop count: 3
        Hop 1 (Strict):
          NAI: IPv4 Adjacency ID, 0.0.0.0 -> 10.11.2.2
          SID type: 20-bit label, Value: 16
        Hop 2 (Loose):
          NAI: IPv4 Node ID, Node address: 192.168.0.2
          SID type: 20-bit label, Value: 801002
        Hop 3 (Loose):
          NAI: IPv4 Node ID, Node address: 192.168.0.3
          SID type: 20-bit label, Value: 801003


Total displayed BFD sessions: 2 (Up: 2, Down: 0)
```

Meaning

The S-BFD sessions are up and Device R0 can reach Device R3.

S-BFD Session Verification on Device R3

Purpose

Verify the status of the S-BFD sessions on the BFD responder, Device R3.

Action

From the operational mode, enter the **show bfd seamless session extensive**command to verify the status of the S-BFD session.

```
user@R3> show bfd seamless session extensive

    Receive
Type     Descriminator               Table      Address
State    Interval
Local    3232235523 (192.168.0.3)       default
Up       0.050
 Logical system id 0, Route table id 0
 ISSU State: false
```

```
 Authentication: false

1 local sessions, 0 remote sessions
```

Meaning

The S-BFD session is up and the configured BFD local discriminator corresponds with the BFD remote discriminator value.

Appendix 1: Set Commands on All Devices

```
R0
set system host-name R1
set chassis network-services enhanced-ip
set interfaces et-0/0/5 unit 0 description "CONNECTED TO R4"
set interfaces et-0/0/5 unit 0 family inet address 10.14.1.1/24
set interfaces et-0/0/5 unit 0 family iso
set interfaces et-0/0/5 unit 0 family mpls
set interfaces et-1/0/3 unit 0 description "CONNECTED TO R1"
set interfaces et-1/0/3 unit 0 family inet address 10.11.2.1/24
set interfaces et-1/0/3 unit 0 family iso
set interfaces et-1/0/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.0.0/32
set interfaces lo0 unit 0 family iso address 49.0001.000a.0a0a.0a00
set policy-options policy-statement prefix-sid term 1 from route-filter 192.168.0.0/32 exact
set policy-options policy-statement prefix-sid term 1 then prefix-segment index 100
set policy-options policy-statement prefix-sid term 1 then prefix-segment node-segment
set policy-options policy-statement prefix-sid term 1 then accept
set routing-options router-id 192.168.0.0
set routing-options autonomous-system 64512
set routing-options rib-groups test import-rib inet.0
set routing-options rib-groups test import-rib inet.3
set protocols isis interface et-0/0/5.0 level 2 metric 20
set protocols isis interface et-0/0/5.0 node-link-protection
set protocols isis interface et-0/0/5.0 point-to-point
set protocols isis interface et-1/0/3.0 level 2 metric 20
set protocols isis interface et-1/0/3.0 node-link-protection
set protocols isis interface et-1/0/3.0 point-to-point
set protocols isis interface all ldp-synchronization
set protocols isis interface all node-link-protection
set protocols isis interface fxp0.0 disable
```

```
set protocols isis interface lo0.0 disable
set protocols isis source-packet-routing adjacency-segment hold-time 240000
set protocols isis source-packet-routing srgb start-label 800000
set protocols isis source-packet-routing srgb index-range 80000
set protocols isis source-packet-routing explicit-null
set protocols isis source-packet-routing traffic-statistics statistics-granularity per-interface
set protocols isis level 1 disable
set protocols isis level 2 wide-metrics-only
set protocols isis level 2 preference 180
set protocols isis backup-spf-options remote-backup-calculation
set protocols isis backup-spf-options use-source-packet-routing
set protocols isis traffic-engineering l3-unicast-topology
set protocols isis traffic-engineering family inet shortcuts
set protocols isis export prefix-sid
set protocols isis ignore-attached-bit
set protocols isis rib-group inet test
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols source-packet-routing segment-list srpath1 auto-translate unprotected
set protocols source-packet-routing segment-list srpath1 hop1 ip-address 10.11.2.2
set protocols source-packet-routing segment-list srpath1 hop2 ip-address 192.168.0.2
set protocols source-packet-routing segment-list srpath1 hop2 label-type node
set protocols source-packet-routing segment-list srpath1 hop3 ip-address 192.168.0.3
set protocols source-packet-routing segment-list srpath1 hop3 label-type node
set protocols source-packet-routing segment-list srpath2 auto-translate unprotected
set protocols source-packet-routing segment-list srpath2 hop1 ip-address 10.14.1.2
set protocols source-packet-routing source-routing-path srlsp1 to 192.168.0.3
set protocols source-packet-routing source-routing-path srlsp1 primary srpath1 weight 23
set protocols source-packet-routing source-routing-path srlsp1 primary srpath1 bfd-liveness-
detection sbfd
set protocols source-packet-routing source-routing-path srlsp1 primary srpath1 bfd-liveness-
detection minimum-interval 1000
set protocols source-packet-routing source-routing-path srlsp1 primary srpath2 weight 45
set protocols source-packet-routing source-routing-path srlsp1 primary srpath2 bfd-liveness-
detection sbfd
set protocols source-packet-routing source-routing-path srlsp1 primary srpath2 bfd-liveness-
detection minimum-interval 1000
set protocols source-packet-routing inherit-label-nexthops
```

R1

```
set system host-name R1
```

```
set chassis network-services enhanced-ip
set interfaces xe-0/3/0 unit 0 description "CONNECTED TO R0"
set interfaces xe-0/3/0 unit 0 family inet address 10.11.2.2/24
set interfaces xe-0/3/0 unit 0 family iso
set interfaces xe-0/3/0 unit 0 family mpls
set interfaces xe-0/3/1 unit 0 description "CONNECTED TO R2"
set interfaces xe-0/3/1 unit 0 family inet address 11.12.1.1/24
set interfaces xe-0/3/1 unit 0 family iso
set interfaces xe-0/3/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.0.1/32
set interfaces lo0 unit 0 family iso address 49.0001.0001.0101.0100
set policy-options policy-statement prefix-sid term 1 from route-filter 192.168.0.1/32 exact
set policy-options policy-statement prefix-sid term 1 then prefix-segment index 1001
set policy-options policy-statement prefix-sid term 1 then prefix-segment node-segment
set policy-options policy-statement prefix-sid term 1 then accept
set routing-options router-id 192.168.0.1
set routing-options autonomous-system 64512
set routing-options rib-groups test import-rib inet.0
set routing-options rib-groups test import-rib inet.3
set protocols isis backup-spf-options remote-backup-calculation
set protocols isis backup-spf-options use-source-packet-routing
set protocols isis backup-spf-options per-prefix-calculation
set protocols isis traffic-engineering family inet shortcuts
set protocols isis source-packet-routing adjacency-segment hold-time 240000
set protocols isis source-packet-routing srgb start-label 800000
set protocols isis source-packet-routing srgb index-range 80000
set protocols isis source-packet-routing explicit-null
set protocols isis level 1 disable
set protocols isis level 1 wide-metrics-only
set protocols isis level 2 wide-metrics-only
set protocols isis interface xe-0/3/0.0 point-to-point
set protocols isis interface xe-0/3/0.0 node-link-protection
set protocols isis interface xe-0/3/1.0 point-to-point
set protocols isis interface xe-0/3/1.0 node-link-protection
set protocols isis interface all node-link-protection
set protocols isis interface em0.0 disable
set protocols isis interface fxp0.0 disable
set protocols isis export prefix-sid
set protocols isis ignore-attached-bit
set protocols isis rib-group inet test
```

```
set protocols mpls traffic-engineering
set protocols mpls interface all
```

R2

```
set system host-name R2
set chassis network-services enhanced-ip
set interfaces xe-0/2/1 unit 0 description "CONNECTED TO R1"
set interfaces xe-0/2/1 unit 0 family inet address 11.12.1.2/24
set interfaces xe-0/2/1 unit 0 family iso
set interfaces xe-0/2/1 unit 0 family mpls
set interfaces xe-0/3/0 unit 0 description "CONNECTED TO R3"
set interfaces xe-0/3/0 unit 0 family inet address 12.13.1.1/24
set interfaces xe-0/3/0 unit 0 family iso
set interfaces xe-0/3/0 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.0.2/32
set interfaces lo0 unit 0 family iso address 49.0001.0002.0202.0200
set policy-options policy-statement prefix-sid term 1 from route-filter 192.168.0.2/32 exact
set policy-options policy-statement prefix-sid term 1 then prefix-segment index 1002
set policy-options policy-statement prefix-sid term 1 then prefix-segment node-segment
set policy-options policy-statement prefix-sid term 1 then accept
set routing-options router-id 192.168.0.2
set routing-options autonomous-system 64512
set protocols isis backup-spf-options remote-backup-calculation
set protocols isis backup-spf-options use-source-packet-routing
set protocols isis traffic-engineering family inet shortcuts
set protocols isis source-packet-routing adjacency-segment hold-time 240000
set protocols isis source-packet-routing srgb start-label 800000
set protocols isis source-packet-routing srgb index-range 80000
set protocols isis source-packet-routing explicit-null
set protocols isis level 1 disable
set protocols isis level 1 wide-metrics-only
set protocols isis level 2 wide-metrics-only
set protocols isis interface xe-0/2/1.0 point-to-point
set protocols isis interface xe-0/2/1.0 node-link-protection
set protocols isis interface xe-0/3/0.0 point-to-point
set protocols isis interface xe-0/3/0.0 node-link-protection
set protocols isis interface all node-link-protection
set protocols isis interface em0.0 disable
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0 passive
set protocols isis export prefix-sid
```

```
set protocols isis ignore-attached-bit
set protocols mpls interface all
```

R3

```
set system host-name R3
set chassis network-services enhanced-mode
set interfaces et-0/0/1 unit 0 description "CONNECTED TO R2"
set interfaces et-0/0/1 unit 0 family inet address 12.13.1.2/24
set interfaces et-0/0/1 unit 0 family iso
set interfaces et-0/0/1 unit 0 family mpls
set interfaces et-0/0/3 unit 0 description "CONNECTED TO R5"
set interfaces et-0/0/3 unit 0 family inet address 10.15.1.1/24
set interfaces et-0/0/3 unit 0 family iso
set interfaces et-0/0/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.0.3/32
set interfaces lo0 unit 0 family iso address 49.0001.0003.0303.0300
set policy-options policy-statement prefix-sid term 1 from route-filter 192.168.0.3/32 exact
set policy-options policy-statement prefix-sid term 1 then prefix-segment index 1003
set policy-options policy-statement prefix-sid term 1 then prefix-segment node-segment
set policy-options policy-statement prefix-sid term 1 then accept
set routing-options router-id 192.168.0.3
set routing-options autonomous-system 64512
set protocols bfd sbfd local-discriminator-ip 192.168.0.3
set protocols isis interface et-0/0/1.0 node-link-protection
set protocols isis interface et-0/0/1.0 point-to-point
set protocols isis interface et-0/0/3.0 node-link-protection
set protocols isis interface et-0/0/3.0 point-to-point
set protocols isis interface all ldp-synchronization
set protocols isis interface all node-link-protection
set protocols isis interface em0.0 disable
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0 passive
set protocols isis source-packet-routing adjacency-segment hold-time 240000
set protocols isis source-packet-routing srgb start-label 800000
set protocols isis source-packet-routing srgb index-range 80000
set protocols isis source-packet-routing explicit-null
set protocols isis level 1 disable
set protocols isis level 1 wide-metrics-only
set protocols isis level 2 wide-metrics-only
set protocols isis backup-spf-options remote-backup-calculation
set protocols isis backup-spf-options use-source-packet-routing
```

```
set protocols isis export prefix-sid
set protocols isis ignore-attached-bit
set protocols mpls interface all
```

R4

```
set system host-name R4
set chassis network-services enhanced-ip
set interfaces xe-0/2/0 description "CONNECTED TO R0"
set interfaces xe-0/2/0 unit 0 family inet address 10.14.1.2/24
set interfaces xe-0/2/0 unit 0 family iso
set interfaces xe-0/2/0 unit 0 family mpls
set interfaces xe-0/3/0 unit 0 description "CONNECTED TO R5"
set interfaces xe-0/3/0 unit 0 family inet address 10.15.1.1/24
set interfaces xe-0/3/0 unit 0 family iso
set interfaces xe-0/3/0 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.0.4/32
set interfaces lo0 unit 0 family iso address 49.0001.0004.0404.0400
set policy-options policy-statement prefix-sid term 1 from route-filter 192.168.0.4/32 exact
set policy-options policy-statement prefix-sid term 1 then prefix-segment index 1004
set policy-options policy-statement prefix-sid term 1 then prefix-segment node-segment
set policy-options policy-statement prefix-sid term 1 then accept
set routing-options router-id 192.168.0.4
set routing-options autonomous-system 64512
set protocols isis backup-spf-options remote-backup-calculation
set protocols isis backup-spf-options use-source-packet-routing
set protocols isis traffic-engineering family inet-mpls shortcuts
set protocols isis traffic-engineering family inet shortcuts
set protocols isis source-packet-routing adjacency-segment hold-time 240000
set protocols isis source-packet-routing srgb start-label 800000
set protocols isis source-packet-routing srgb index-range 80000
set protocols isis source-packet-routing explicit-null
set protocols isis level 1 disable
set protocols isis level 1 wide-metrics-only
set protocols isis level 2 wide-metrics-only
set protocols isis interface xe-0/2/0.0 point-to-point
set protocols isis interface all node-link-protection
set protocols isis interface all ldp-synchronization
set protocols isis interface em0.0 disable
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0 passive
set protocols isis export prefix-sid
```

```
set protocols isis ignore-attached-bit
set protocols isis suppress-attached-bit
set protocols mpls interface all
```

R5

```
set system host-name R5
set chassis network-services enhanced-ip
set interfaces xe-0/2/0 unit 0 description "CONNECTED To R3"
set interfaces xe-0/2/0 unit 0 family inet address 10.15.1.2/24
set interfaces xe-0/2/0 unit 0 family iso
set interfaces xe-0/2/0 unit 0 family mpls
set interfaces xe-0/2/1 unit 0 description "CONNECTED To R4"
set interfaces xe-0/2/1 unit 0 family inet address 10.16.1.2/24
set interfaces xe-0/2/1 unit 0 family iso
set interfaces xe-0/2/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.0.5/32
set interfaces lo0 unit 0 family iso address 49.0001.0005.0505.0500
set policy-options policy-statement prefix-sid term 1 from route-filter 192.168.0.5/32 exact
set policy-options policy-statement prefix-sid term 1 then prefix-segment index 1005
set policy-options policy-statement prefix-sid term 1 then prefix-segment node-segment
set policy-options policy-statement prefix-sid term 1 then accept
set routing-options router-id 192.168.0.5
set routing-options autonomous-system 64512
set protocols isis backup-spf-options remote-backup-calculation
set protocols isis backup-spf-options use-source-packet-routing
set protocols isis traffic-engineering family inet shortcuts
set protocols isis source-packet-routing adjacency-segment hold-time 240000
set protocols isis source-packet-routing srgb start-label 800000
set protocols isis source-packet-routing srgb index-range 80000
set protocols isis source-packet-routing explicit-null
set protocols isis level 1 disable
set protocols isis level 1 wide-metrics-only
set protocols isis level 2 wide-metrics-only
set protocols isis interface xe-0/2/0.0 point-to-point
set protocols isis interface xe-0/2/1.0 point-to-point
set protocols isis interface all node-link-protection
set protocols isis interface all ldp-synchronization
set protocols isis interface em0.0 disable
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0 passive
set protocols isis export prefix-sid
```

```
set protocols isis ignore-attached-bit
set protocols mpls interface all
```

Appendix 2: Show Configuration Output on R0

Show configuration output on Device R0.

```
user@R0# show interfaces
et-0/0/5 {
    unit 0 {
        description "CONNECTED TO R4";
        family inet {
            address 10.14.1.1/24;
        }
        family iso;
        family mpls;
    }
}
et-1/0/3 {
    unit 0 {
        description "CONNECTED TO R1";
        family inet {
            address 10.11.2.1/24;
        }
        family iso;
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 192.168.0.0/32;
        }
        family iso {
            address 49.0001.000a.0a0a.0a00;
        }
    }
}
```

```
usre@R0# show policy-options
policy-statement prefix-sid {
    term 1 {
```

```
        from {
            route-filter 192.168.0.0/32 exact;
        }
        then {
            prefix-segment {
                index 100;
                node-segment;
            }
            accept;
        }
    }
}
```

```
user@R0# show routing-options
router-id 192.168.0.0;
autonomous-system 64512;
rib-groups {
    test {
        import-rib [ inet.0 inet.3 ];
    }
}
```

```
user@R0# show protocols
isis {
    interface et-0/0/5.0 {
        level 2 metric 20;
        node-link-protection;
        point-to-point;
    }
    interface et-1/0/3.0 {
        level 2 metric 20;
        node-link-protection;
        point-to-point;
    }
    interface all {
        ldp-synchronization;
        node-link-protection;
    }
    interface fxp0.0 {
        disable;
```

```
    }
    interface lo0.0 {
        disable;
    }
    source-packet-routing {
        adjacency-segment hold-time 240000;
        srgb start-label 800000 index-range 80000;
        explicit-null;
        traffic-statistics {
            statistics-granularity per-interface;
        }
    }
    level 1 disable;
    level 2 {
        wide-metrics-only;
        preference 180;
    }
    backup-spf-options {
        remote-backup-calculation;
        use-source-packet-routing;
    }
    traffic-engineering {
        l3-unicast-topology;
        family inet {
            shortcuts;
        }
    }
    export prefix-sid;
    ignore-attached-bit;
    rib-group inet test;
}
mpls {
    interface all;
    interface fxp0.0 {
        disable;
    }
}
source-packet-routing {
    segment-list srpath1 {
        auto-translate {
            unprotected;
        }
        hop1 ip-address 10.11.2.2;
```

```
        hop2 {
            ip-address 192.168.0.2;
            label-type {
                node;
            }
        }
        hop3 {
            ip-address 192.168.0.3;
            label-type {
                node;
            }
        }
    }
    segment-list srpath2 {
        auto-translate {
            unprotected;
        }
        hop1 ip-address 10.14.1.2;
    }
    source-routing-path srlsp1 {
        to 192.168.0.3;
        primary {
            srpath1 {
                weight 23;
                bfd-liveness-detection {
                    sbfd;
                    minimum-interval 1000;
                }
            }
            srpath2 {
                weight 45;
                bfd-liveness-detection {
                    sbfd;
                    minimum-interval 1000;
                }
            }
        }
    }
    inherit-label-nexthops;
}
```

Configuring Static Adjacency Segment Identifier for Aggregate Ethernet Member Links Using Single-Hop Static LSP

In a network where aggregate Ethernet (AE) bundles are in use, an aggregate link could be a bundle of any number of physical links. The traffic sent over these AE bundle interfaces are forwarded on any of the member links of an AE interface. The traffic can take any physical link based on the hash defined for load-balancing the traffic, which makes it difficult to isolate which links have gone bad or are dropping the traffic. One way to test the forwarding path available in the network is to add a single-hop static label switched path (LSP) with the next hop pointing to a specific member link of the AE interface.

The default label operation for the static LSPs must be pop and forward. When a packet hits these labeled routes, the packet is forwarded on to a specific member link. A unique label is used to identify the member link. These labels are referred to as adjacency segment identifiers (SID) and are statically provisioned.

You can control the flow of the packets in the network by constructing a label stack in controller, which includes the labels allocated by all transit static LSP. Operation, Administration, and Maintenance (OAM) packets are crafted and injected into the network with entire label-stack.

When a packet hits this label route the label is popped and traffic is forwarded on the member link specified in the configuration. A destination MAC is chosen while forwarding the packet, the destination Mac is the aggregate interface MAC address (selected based on nexthop address configured).

When the member link goes down and aggregate interface is up, then the route corresponding to that member link is deleted. When an aggregate interface goes down, then all the routes corresponding to member links of the aggregate interface are deleted. When the child physical interface is LACP detached but the child physical interface is up, the labeled route for the child link is deleted. In the case of LACP detach, if the member link is up and invalid forwarding state, then the OAM packets is dropped in the PFE when the child physical interface is detached.

Use the following example to configure single-hop static LSP for an AE member link.

1. Define a static label range.

```
user@host# set protocols mpls label-range static-label-range 1000000 1048575;
```

NOTE: We recommend configuring the default static label range of 1000000-1048575 for the static LSP. If you wish to configure a label range other than the default static label range, configure multiple ranges.

2. Create a static LSP for the AE member link from the segment routing local block (SRLB) pool of the static label range.

```
user@host# set protocols mpls static-label-switched-path statc-lsp transit 100001 pop next-hop 10.1.1.1
member-interface ge-0/0/0
```

In this configuration, a transit labelled router is installed in mpls.0, pops the label, and forwards the packet down the next hop. The next hop address is mandatory for broadcast interfaces (such as ge-, xe-, ae-) and the if-name is used for P2P interfaces (such as so-). The address is required for broadcast interfaces because the next hop IP address is used to pick the destination MAC address. The source MAC address for the packet is the AE's MAC address.

The sample outputs display the member link name in the next hop output:

show mpls static-lsp extensive

```
user@host> show mpls static-lsp extensive
Ingress LSPs:
Total 0, displayed 0, Up 0, Down 0

Transit LSPs:
LSPname: static-lsp1, Incoming-label: 1000001
  Description: verify-static-lsp-behavior
  State: Up, Sub State: Traffic via primary but unprotected
  Nexthop: 10.2.1.1 Via ae0.0 -> ge-0/0/0
  LabelOperation: Pop
  Created: Thu May 25 15:31:26 2017
  Bandwidth: 0 bps
  Statistics: Packets 0, Bytes 0
```

show route label label-name extensive

```
user@host> show route label 1000001 extensive

mpls.0: 14 destinations, 14 routes (14 active, 0 holddown, 0 hidden)
1000001 (1 entry, 1 announced)
TSI:
KRT in-kernel 1000001/52 -> {Pop      }
        *MPLS   Preference: 6
                Next hop type: Router, Next hop index: 611
                Address: 0xb7a17b0
```

```
                Next-hop reference count: 2
                Next hop: 10.2.1.1 via ae0.0 -> ge-0/0/0 weight 0x1, selected
                Label operation: Pop
                Load balance label: None;
                Label element ptr: 0xb7a1800
                Label parent element ptr: 0x0
                Label element references: 1
                Label element child references: 0
                Label element lsp id: 0
                Session Id: 0x15d
                State: <Active Int>
                Age: 3:13:15    Metric: 1
                Validation State: unverified
                Task: MPLS
                Announcement bits (1): 1-KRT
                AS path: I
                Label 188765184
```

Non-Router-ID Endpoints in Segment Routing Traffic Engineering

IN THIS SECTION

- Benefits of Non-Router-ID Endpoints for SR-TE | **914**
- Anycast as a Destination Address | **914**

Historically, RSVP-TE supported non-router-ID endpoints. Adding non-router-ID endpoints into Segment Routing Traffic Engineering (SR-TE) policies enhances your network's flexibility and efficiency across both IPv4 and IPv6 within SR-MPLS frameworks. The non-router-ID endpoint in SR-TE allows ingress routers to steer traffic to logical destinations (for example, anycast service address) rather than being constrained to node router-IDs. By enabling the specifications of anycast, secondary loopback (lo0), and interface prefix addresses as destination points, you can achieve robust load-balancing and redundancy.

Traditionally, SR-TE policies use router IDs, but you can specify anycast addresses to enhance redundancy and load balancing in SR-MPLS networks. Use IPv4 and IPv6 anycast addresses as IGP-learned destinations with or without SID stack compression. These anycast addresses are not redistributed (R-bit set). Use them as the to address for SR-TE policies with associated compute profiles.

NOTE: We currently support endpoint of type anycast as the destination address. We do not support endpoint types—interface prefixes and secondary loopback (lo0) address as destinations.

Benefits of Non-Router-ID Endpoints for SR-TE

- Enhances network flexibility by targeting multiple nodes advertising a shared prefix, facilitating effective load-balancing and redundancy.
- Expands SR-TE configurations with anycast addresses, providing more options for network design and optimization.
- Improves packet ingress control without additional traffic engineering constraints, allowing efficient network management.
- Optimizes routing paths for robust network performance, ensuring seamless traffic distribution and improved resilience.

Anycast as a Destination Address

Multiple nodes can advertise a single anycast IP address (IPv4 or IPv6) and corresponding anycast SIDs throughout an IGP domain.

An anycast SID is a type of prefix SID that identifies a set of nodes. The set of nodes (anycast group) is configured to advertise a shared prefix address and prefix SID. Anycast routing enables the steering of traffic toward multiple advertising nodes, providing load-balancing and redundancy. Packets addressed to an anycast address are forwarded to the topologically nearest nodes.

An ingress router is configured with an SR-TE policy using the anycast IP address (IPv4 or IPv6) as the endpoint.

Use the `show spring-traffic-engineering detail` command to view comprehensive details on non-router-ID endpoints, showing computed segments and anycast SID details. Use these commands to ensure SR-TE policies are correctly implemented, achieving seamless traffic distribution and optimized routing paths.

Computing Delay Optimized Intradomain and Interdomain Segment Routing Paths

IN THIS SECTION

- Delay-based Metrics for Traffic Engineered Paths Overview | **915**
- Benefits of Delay-based Metrics for Path Computation | **915**

- DCSPF-based Computation with Delay Metrics for SR Path Overview | **915**
- Delay Metrics for Interdomain and Intradomain Use Case Overview | **917**
- Delay Metrics in Optical Networks Use Case | **918**

Delay-based Metrics for Traffic Engineered Paths Overview

Leveraging dynamic delay-based metrics is becoming a desirable attribute in modern networks. Delay-based metrics is essential for a Path Computation Element (PCE) to compute traffic engineered paths. You can use delay-based metrics to steer packets on the least latency paths, or the least delay path. To do this, you need to measure the delay on every link and advertise it using a suitable routing protocol (IGP or BGP-LS), so that the ingress has the per link delay properties in its TED.

To understand how to advertise the delay on each link, or turn on link measurements, see How to Enable Link Delay Measurement and Advertising in ISIS.

The following sequence of events happen when you measure, advertise, and compute the detection of changes in the network and calculate traffic engineered path with shortest latency:

- Detect changes in the network by measuring the latency, with synthetic probes, router-to-router.
- Flood the results to ingress nodes through IGP extended TE-metric extensions.
- Advertise the results to centralized controllers with corresponding BGP-LS extensions.
- Compute IGP-based shortest cumulative latency metric paths (Flex-algo).
- Compute traffic-engineered explicit paths (source to destination) with shortest cumulative latency or delay metrics (SR-TE).

Benefits of Delay-based Metrics for Path Computation

- Deploy value-added services based on the lowest latency throughout the network.
- Measure per path latency (minimum, maximum, average) using delay-based metrics.
- Steer delay sensitive services (such as VPN or 5G services) on ultra-low latency SR optimized paths.

DCSPF-based Computation with Delay Metrics for SR Path Overview

Using the distributed Constrained Shortest Path First (CSPF) for segment routing LSP feature, you can compute a segment routing LSP locally on the ingress device according to the constraints you have

configured. With this feature, the LSPs are optimized based on the configured constraints and metric type (traffic-engineering or IGP). The LSPs are computed to utilize the available ECMP paths to the destination with segment routing label stack compression enabled or disabled.

You can configure distributed CSFP to run on multiple headends and do path computation independently on each headend. You can apply compute profile on the source path (source packet routing path). If you have configured compute profile optimized for delay average, you can also additionally apply a constraint called the `delay-variation-threshold`. When you configure a value for `delay-variation-threshold`, any link exceeding the threshold value would be excluded from path computation.

To configure delay metrics for DCSPF-based computation for SR path, you need to enable the `delay` configuration statement at the [`edit protocols source-packet-routing compute-profile` *`profile-name`* `metric-type delay`] hierarchy level. You can configure the delay metrics such as minimum, maximum, average, and delay variation threshold for path calculation.

- `minimum`—Minimum delay metric value from TED for cumulative lowest latency path calculation.
- `average`—Average delay metric value from TED for cumulative lowest latency path calculation.
- `maximum`—Maximum delay metric value from TED for cumulative lowest latency path calculation.
- `delay-variation-threshold`—Link delay variation threshold (1..16777215). Any link exceeding the delay variation threshold would be excluded from path calculation. The delay variation threshold is independent of whether you are doing optimization for minimum, maximum, or average. The `delay-variation-threshold` configuration statement appears at these hierarchy levels:
  - [`edit protocols source-packet-routing compute-profile` *`profile-name`* `metric-type delay`]
  - [`edit protocols source-packet-routing compute-profile` *`profile-name`* `metric-type delay minimum`]
  - [`edit protocols source-packet-routing compute-profile` *`profile-name`* `metric-type delay maximum`]
  - [`edit protocols source-packet-routing compute-profile` *`profile-name`* `metric-type delay average`]

You can configure delay metrics at the following CLI hierarchy:

```
[edit]
protocols {
    source-packet-routing {
        compute-profile <name> {
            metric-type delay {
                  minimum;
                  maximum;
                  average;
                  delay-variation-threshold <value>;
```

```
                }
            }
        }
}
```

Delay Metrics for Interdomain and Intradomain Use Case Overview

For the intra-domain case (path resides within a single domain), the link delay is taken into consideration when doing path computation. Delay metrics for segment routing path computation is supported on both compressed SIDs and uncompressed SIDs. If you have uncompressed SIDs, then adjacency segments for segment lists is used to produce multiple segment lists if there are equal costs. You can configure uncompressed SIDs using the `no-label-stack-compression` configuration statement at the [`edit protocols source-packet-routing compute-profile` *`profile-name`*] hierarchy level. This provides a fully expanded path using adjacency SIDs. See compute-profile for more information.

The following is a sample configuration for delay metrics:

```
[edit protocols source-packet-routing]
user@host# show
compute-profile profile1 {
    no-label-stack-compression;
    metric-type {
        delay {
            minimum;
            delay-variation-threshold 300;
        }
    }
}
```

NOTE: For optical path computation, it is recommended to use the delay metrics as minimum. Minimum is the default configuration.

For interdomain (multi-domain) use case, where there are multiple autonomous systems, you can use express segments to configure link delays for path computation. Express segments are links that connect border nodes or ASBRs. See Express Segment LSP Configuration to learn about express segments. You can configure the delay or inherit the delay of the underlying LSP in the express segment. You can configure `delay` at the [`edit protocols express-segments segment-template` *`template-name`* `metric`] hierarchy level and set the values for minimum, maximum, and average delays.

You can configure delay metrics in an express segment at the following CLI hierarchy:

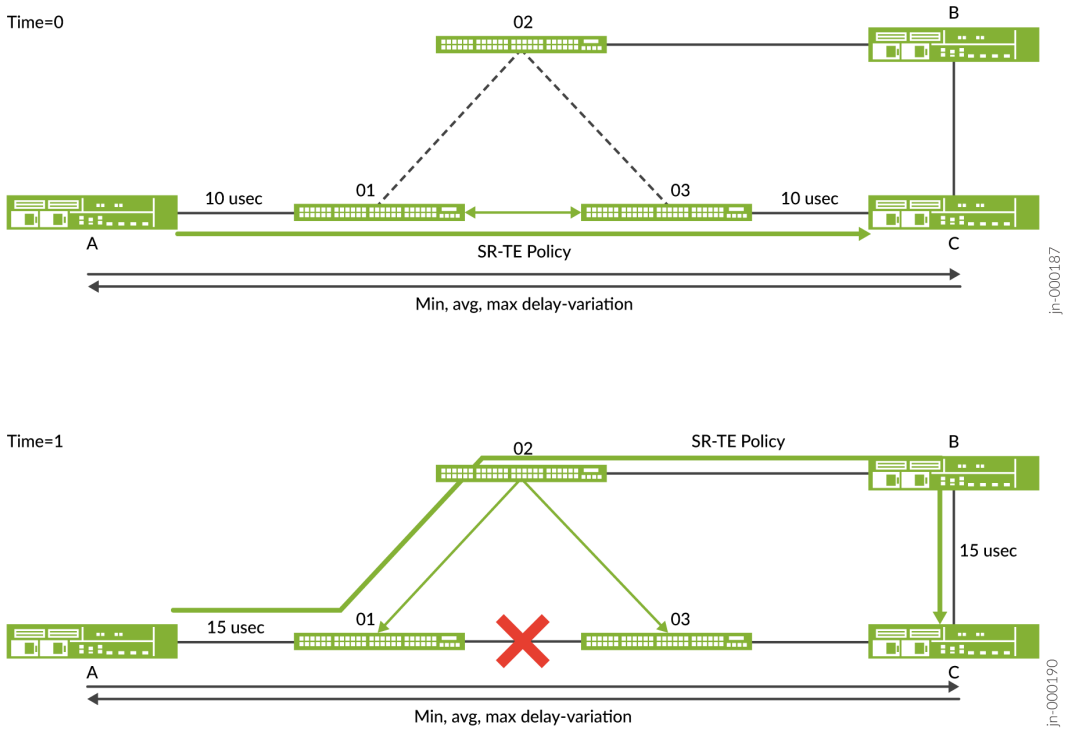
```
[edit]
protocols {
    express-segments {
        segment-template <name> {
           metric delay [ min <value> | avg <value> | max <value>
            }
        }
  }
```

For interdomain path computation, you can assign delay metrics on BGP EPE links. You can configure a value for `delay-metric` at the [`edit protocols bgp group` *`group-name`* `neighbor` *`ip address`* `egress-te-adj-segment` *`segment-name`* `egress-te-link attribute`] hierarchy level as shown below:

```
[edit]
protocols {
    bgp {
        group <name> {
            type external;
        }
        neighbor <ip_addr> {
            egress-te-adj-segment <name> {
                egress-te-link-attribute {
                  delay-metric <value>
                  }
                }
              }
          }
      }
```

Delay Metrics in Optical Networks Use Case

The following topologies depict an example of an optical use case. Optical protection and restoration solutions result in the underlying physical attributes, mainly path length, changing due to link failures and subsequent DWDM network optimization.



In figure on page 919, the link between A and C is through the optical nodes O1 and O3 and has a latency of 10 microseconds. In figure on page 919, you can see a link failure between optical nodes O1 and O3 and that the actual optical connection is rerouted through the optical node O2. This increases the latency of 15 microseconds. The metric for the link that connects A and B changes dynamically between time=0 and time=1.

When a change in the per link delay is detected by the ingress, the ingress triggers recomputation of the delay optimized path and the headend router reroutes the path over the next available least latency path. The change in the link delay may result in the ingress choosing another set of links that has the least latency path. In figure B, you can see there is a change in the link between the path A and C and the headend router reroutes and selects the path A-B-C as shown in the topology.

Understanding SRv6 TE Tunnel Local Path Computation

IN THIS SECTION

- Overview | **920**
- Benefits of SRv6 TE Tunnel Local Computation | **920**

Overview

The implementation of local computation for SRv6 TE tunnels, including classical and microsegmentation SIDs (uSIDs), significantly enhances path computation capabilities within your network. By enabling local path computation, you can dynamically and efficiently compute traffic engineering paths using SRv6 SIDs based on current network topology and constraints. Configuration options allow you to enable SRv6 TE path computation, attach compute profiles, and set IGP instance and topology constraints. Additionally, preferences for SID types and support for Seamless Bidirectional Forwarding Detection (sBFD) over SRv6 TE compute tunnels ensure optimized routing and high network efficiency, making this feature integral for scalable and adaptive network design.

Benefits of SRv6 TE Tunnel Local Computation

- Optimizes network efficiency by dynamically computing traffic engineering paths based on real-time topology and constraints, reducing unnecessary overhead and improving resource utilization.
- Enhances scalability by supporting both classical SRv6 SIDs and uSIDs, allowing for more efficient routing and management of segment lists.
- Facilitates adaptive routing through local computation, enabling quick adjustments to network changes without relying on external computation sources.
- Ensures high network efficiency and reliability with support for sBFD over SRv6 TE tunnels, providing robust failure detection and response mechanisms.

Configuring SRv6 TE Tunnel Local Path Computation

To configure SRv6 TE path computation, use the following CLI configurations:

1. Enable SRv6 TE path computation by allowing SRv6 TE tunnels with compute type primary and secondary:

```
user@host# set protocols source-packet-routing source-routing-path srv6-tunnel srv6
user@host# set protocols source-packet-routing source-routing-path srv6-tunnel primary name
compute
```

2. Attach the compute-profiles tot he SRv TE tunnel:

```
user@host# set protocols source-packet-routing source-routing-path srv6-tunnel primary name
compute compute-profile-name
```

3. Define the compute profile to be used in the computation:

```
user@host# set protocols source-packet-routing compute-profile compute-profile-name srv6
```

4. Specify whether to use classical SRv6 SIDs in your configuration:

```
user@host# set protocols source-packet-routing compute-profile name srv6 no-micro-sid
```

Commit checks validate SRv6-enabled compute profiles and segment lists, and confirm the existence of topology configurations in IGP. Diagnostic commands, such as `show spring-traffic-engineering lsp detail` and `show spring-traffic-engineering route detail`, help you troubleshoot and monitor path computation operations effectively. Trace options are also available to provide detailed logs for SRv6 TE activities, aiding in comprehensive diagnostics and troubleshooting.

Understand BGP VPN Option C MPLS over SRv6

SUMMARY

IN THIS SECTION

- Benefits of BGP VPN Option C with MPLS over SRv6 | **922**
- Overview | **922**
- BGP-LU LSP is Tunneled Across SRv6 Domain | **922**
- Enable SRv6 Signaling for BGP-LU Address Family | **923**
- Egress Route Behavior | **923**
- Ingress Route Behavior | **923**
- Transit Route Behavior | **923**
- Configure end-dtm-sid | **924**
- Configure advertise-srv6-service for BGP-LU Address Family | **924**
- Configure accept-srv6-service for BGP-LU address family | **924**
- Configure end-dtm-sid with Policy | **925**

BGP VPN Option C with MPLS over Segment Routing for IPv6 (SRv6) enhances multi-domain network architectures by facilitating seamless interworking between SR-MPLS for IPv4 (SR-MPLS-IPv4) and

SRv6 domains. This feature uses SRv6 as a transport layer, allowing you to advertise SRv6 Segment Identifiers (SIDs) alongside MPLS labels for optimal routing and forwarding.

Benefits of BGP VPN Option C with MPLS over SRv6

- Enhances network scalability by enabling seamless integration of MPLS and SRv6 domains, facilitating efficient routing in multi-domain environments.
- Improves routing flexibility through SRv6 Segment Identifiers (SIDs) alongside MPLS labels, allowing for dynamic transport scenarios and optimal packet delivery.
- Enables precise control over routing paths with support for ingress, egress, and transit route configurations, ensuring accurate label manipulation and efficient data forwarding.
- Increases network robustness and reliability by providing high availability support for SRv6 dynamic SIDs, minimizing service disruptions.
- Supports advanced network operations by allowing the advertisement and acceptance of SRv6 services within BGP-LU address families, enhancing interoperability and service management across complex network setups.

Overview

Consider a multi-domain network, where there is a central domain (C) with many leaf domains around it. Service flows from an ingress PE in an ingress leaf domain (LI), through the central (C) domain and up to an egress PE of the egress leaf domain (LE). Each domain runs its own IGP instance. The leaf domains (LI and LE) use MPLS data plane (SR-MPLS-IPv4), while the central domain (C) uses SRv6 data plane.

Provider edge devices run MPLS based BGP L3 (for example, VPN) services through service route reflectors. Service endpoint signaling through borders routers and corresponding forwarding state provide interworking over intermediate transport domain. The ingress PE encapsulates the payload in an MPLS service label and sends it through MPLS LSP to egress PE. Transport nodes forward encapsulated label stack to egress PE over SRv6 C domain.

BGP Label Unicast (BGP-LU) advertises transport reachability for PE IPv4 loopbacks across the multi-domain network. Next hop self on border routers provides independence of intra domain tunnel technology in different domains. MPLS based L3 BGP services are signaled with IPv4 next-hop of egress PE through service route reflectors (RRs) with no color extended community. Ingress PE need labelled reachability to remote PE IPv4 loopback address advertised as next hop with service routes. BGP-LU advertise IPv4 PE loopbacks. Next hop self is performed on the border routers.

BGP-LU LSP is Tunneled Across SRv6 Domain

- Existing BGP-LU label cross-connect on border routers for each PE IPv4 loopback address.

- The lookups at the ingress border router are based on BGP-LU label.
- SR-MPLS IGP label to next hop is replaced by an IPv6 tunnel with DA = SRv6 SID associated with DTM behavior in the SRv6 central domain.
- Ingress border router forwarding perform label swap and H.Encaps.M with DA = SRv6 SID associated with DTM behavior.

Enable SRv6 Signaling for BGP-LU Address Family

You can enable SRv6 signaling for BGP-LU address family by:

- Enabling `advertise-srv6-service`, `accept-srv6-service` CLI statements for advertising SRv6 service TLVs for BGP-LU address family.
- Supporting SRv6 tunnel for label route of the BGP prefix-SID attribute to signal SRv6 SIDs along with MPLS label bound to prefix in NLRI. SRv6 tunnel for label route TLV is encoded exactly like SRv6 service TLVs in prefix-SID attribute.
- Advertising end-dtm SID in BGP-LU address family without any transposition in SRv6 SID information sub-TLV of SRv6 tunnel for label route TLV in prefix-Sid attribute.

Egress Route Behavior

When end-dtm SID is configured, end-dtm SID route is added to inet6.0 table pointing to mpls.0 table nexthop with endpoint decapsulation and MPLS table lookup behavior. For end-dtm-sid, strict sid route is also added.

Ingress Route Behavior

The ingress border router performs label push and H.Encaps.M.red with IPv6 header destination address as SRv6 SID received in SRv6 tunnel for label route TLV of prefix-Sid attribute.

Transit Route Behavior

Transit border router performs label swap and H.Encaps.M.red with IPv6 header destination address as SRv6 SID received in SRv6 tunnel for label route TLV of prefix-Sid attribute.

Configure end-dtm-sid

Configure static or dynamic end-dtm-Sid. The locator can be flex algo locator.

```
user@host# set protocols bgp source-packet-routing srv6 locator locator-name end-dtm-sid [sid-
value | non-default]
```

The end-dtm-sid is the endpoint with decapsulation and MPLS table lookup behavior.

The non-default sids can be applied to BGP-LU prefixes only through a policy.

Configure advertise-srv6-service for BGP-LU Address Family

If the `advertise-srv6-service` CLI statement is configured, then only the SRv6 tunnel for label route TLV is advertised in BGP prefix SID attribute. The `advertise-srv6-service` statement is applicable for origination. The statement is not applicable for propagation.

```
user@host# set protocols bgp group internal-peers family inet labeled-unicast advertise-srv6-
service
OR
user@host# set protocols bgp family inet labeled-unicast advertise-srv6-service

user@host# set protocols bgp group internal-peers family inet6 labeled-unicast advertise-srv6-
service
OR
user@host# set protocols bgp family inet6 labeled-unicast advertise-srv6-service
```

Configure accept-srv6-service for BGP-LU address family

If the `accept-srv6-service` CLI statement is configured, then only SRv6 tunnel for label route TLV present in BGP prefix SID attribute is accepted.

```
user@host# set protocols bgp group internal-peers family inet labeled-unicast accept-srv6-service
OR
user@host# set protocols bgp family inet labeled-unicast accept-srv6-service

user@host# set protocols bgp group internal-peers family inet6 labeled-unicast accept-srv6-
```

```
service
OR
user@host# set protocols bgp family inet6 labeled-unicast accept-srv6-service
```

Configure end-dtm-sid with Policy

The SRv6 CLI statement is enhanced to configure the SRv6 SID type: end-dtm-sid.

```
user@host# set policy-options policy-statement policy-name term term-name then srv6 end-dtm-sid
```

Platform-Specific Segment Routing LSP Behavior

IN THIS SECTION

- Platform-Specific Segment Routing LSP Behavior | 925

Use Feature Explorer to confirm platform and release support for specific features.

Use the following table to review platform-specific behaviors for your platform:

Platform-Specific Segment Routing LSP Behavior

Platform	Difference
ACX Series	• The `set routing-options resolution preserve-nexthop-hierarchy` configuration statement is mandatory on ACX7000 Series devices for the S-BFD feature to work. • On ACX7000 Series devices, the `lo0` interface has to be configured with local host address of 127.0.0.1 for S-BFD sessions to properly form the adjacency.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
19.4R1	You can configure a tunnel template for PCE-initiated segment routing LSPs to pass down two additional parameters for these LSPs - Bidirectional forwarding detection (BFD) and LDP tunneling.

RELATED DOCUMENTATION

MPLS Overview | 2

Express Segment LSP Configuration

IN THIS SECTION

- Establish End-to-End Segment Routing Path Using Express Segments | 926
- Example: Inter-domain SR-TE Connectivity Using Express Segments Through RSVP-TE Underlay | 935
- Example: Inter-domain SR-TE Connectivity Using Express Segments Through SR-TE Underlay | 1048

Establish End-to-End Segment Routing Path Using Express Segments

IN THIS SECTION

- Benefits of Express Segments | 927
- Use Cases | 928
- How does Express Segment Work? | 929
- How are Express Segments Advertised? | 932
- How are Express Segments Used by a Path Computing Element? | 933

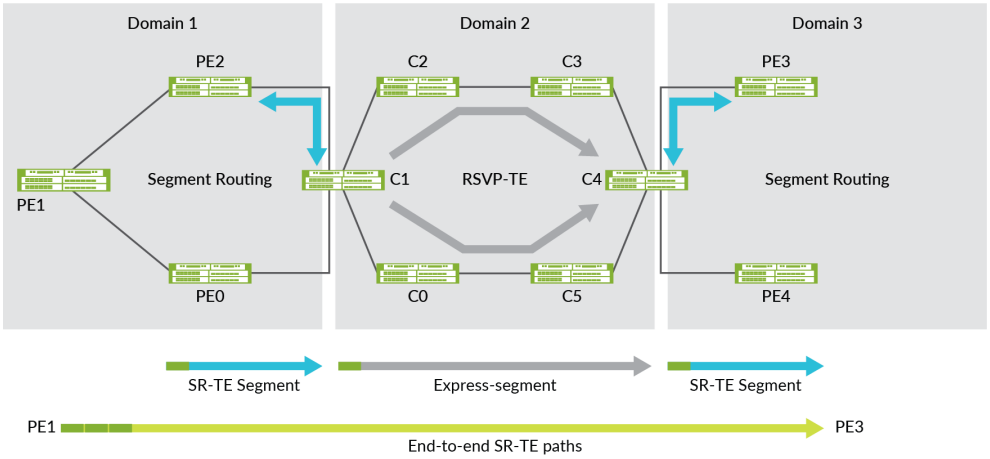
Learn about the benefits, use cases, and overview of how express segments work to establish an end-to-end segment routing path in a multi-domain network.

Benefits of Express Segments

- Express segments are a segment routing (SR) abstraction of an underlay path. Express segments facilitate the establishment of end-to-end SR paths using any underlay technology. The underlay technology currently supported are RSVP-TE and SR-TE. Express segment through RSVP-TE underlay is explained below.

  In Figure 59 on page 927, Domain 2 leverages its RSVP-TE underlay LSPs for traffic engineering management and presents those underlay RSVP-TE LSPs as express segments to the adjacent domains (Domain 1 and Domain 3), therefore enabling end-to-end SR-TE path establishment.

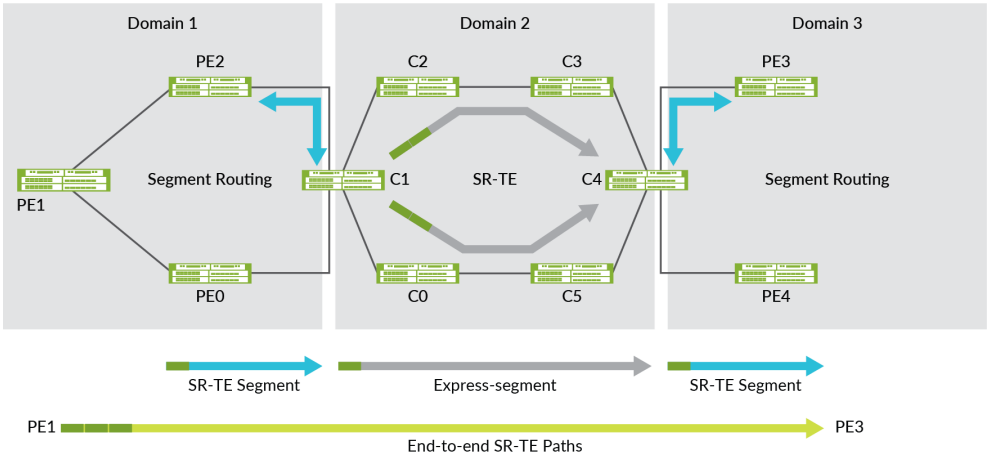
Figure 59: Multi-Domain End-to-End SR-TE with RSVP Underlay



- Express segments implicitly reduce the size of the SR segment list by compressing them (segment lists) to, at a minimum, one segment ID (SID)/label per domain. This becomes useful when end-to-end traffic engineered constraints would otherwise result in a segment list that exceeds the ingress router's label imposition capabilities. This also becomes beneficial when one or more domains are already implementing SR-TE for traffic engineered path management.

  In Figure 60 on page 928, you can see Domain 2 is using SR-TE and how the use of express segments enables PE1 device to use three labels to traverse the multi-domain network instead of five.

Figure 60: Multi-Domain End-to-End SR-TE with Reduced Label Stack

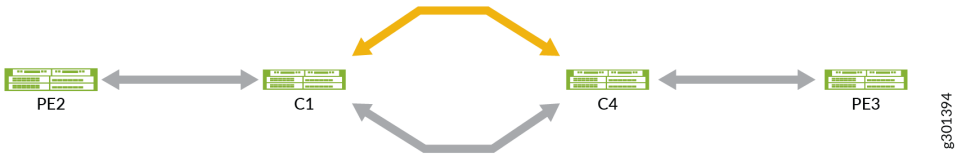


- Express segments allow operators to present an abstraction of the network to adjacent domains and/or higher layer systems.

  To establish a traffic engineered path through a series of interconnected domains or multi-domain network, it is necessary to have a certain amount of traffic engineering information about each network domain. Topology abstraction allows the use of policies to connect across domains. Topology abstraction does not necessarily offer all possible connectivity options but presents a view of potential connectivity according to the policies that determine how the domain resources need to be used. The domain could be constructed as a mesh of border node to border node express segments.

  Using Figure 60 on page 928, PE2's view of an end-to-end traffic engineered system is represented in its local traffic engineering database as shown in Figure 61 on page 928.

Figure 61: Abstracted Traffic Engineered Domain



Use Cases

This section describes a few use cases for establishing end-to-end SR-TE connectivity. *RFC7926* introduces a comprehensive set of terminology and use cases along with an architecture to facilitate

traffic engineering link and node information exchange between domains. As Service providers' networks are expanding because of continued growth, multi-domain networks are becoming more prevalent. In these multi-domain networks, it is required to establish an end-to-end traffic engineered path between one or more domains from a source to a destination

Intra and Inter-domain SR-TE Connectivity Using Express Segments

Express segments have the capability to abstract traffic engineering information when the routing information exchange happens between domains. The traffic engineering information used as a criterion for path selection is the data relating to traffic engineered nodes and links. Traffic engineering information may be link metrics such as IGP, traffic engineering, latency, or administrative link attributes such as affinities. Express segments are best described as virtual traffic engineered Links that facilitate the abstraction of underlay LSPs.

Enhanced On-demand Next-hop

Enhanced On-demand Next-hop (EODN) (also known as BGP-triggered SR policies) facilitates the dynamic provisioning of end-to-end SR-TE policies, with constraints, upon the arrival of services routes. In large networks having hundreds of PE devices creating and maintaining traffic engineering policies on any ingress PE for every egress PE is challenging. Considering colors specific services (per VPN or per group of prefixes) makes things even more complicated and harder to maintain and troubleshoot. BGP triggered SR-TE addresses the task by automatically creating dynamic SR tunnels based on pre-configured templates. There is no need to provision ingress PEs with configuration for every egress PE.

How does Express Segment Work?

Express segments can be used to establish end-to-end traffic engineered paths between interconnected traffic engineered networks. Express segments (also known as virtual traffic engineering links) are generated dynamically through policies matching the underlay LSPs. Express segments and the corresponding abstracted topology (required by *RFC7926*) is generated with policies.

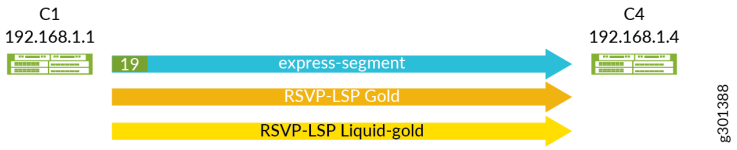
To apply a policy, include the `policy` *`policy-name`* configuration statement at the [`edit protocols express-segment traffic-engineering`] hierarchy level.

NOTE: The *`policy-name`* is optional. If a policy name is not defined, then the policy implicitly imports all the express segments into the local traffic engineering database. An express segment template automatically creates a one-on-one mapping of express links.

To configure express segment, include the `express-segment` configuration statement under the [`edit protocols`] hierarchy level.

Let us refer to Figure 59 on page 927 and use the pair of RSVP-TE LSPs shown between C1 and C4 border nodes and how express segments are generated representing the underlay LSPs. In Figure 62 on page 930, a policy is created to represent two RSVP-TE (gold and liquid-gold) LSPs as a single express segment.

Figure 62: A Pair of RSVP-TE LSPs Represented as an Express Segment



The following is a sample policy where the policy name is matched through a regular expression and the end-point of the RSVP-TE LSPs:

```
protocols {
    express-segment-set gold-exp-seg {
        policy gold;
        }
    }
policy-options {
    policy-statement gold {
        from {
            route-filter 10/8 {
                install-next-hop lsp-regex *gold;
            }
        }
        then accept;
    }
}
```

In the following sample output, you can see the newly created express segment (**Gold-Exp-Set-192.168.1.4**) along with the traffic engineering attributes are inherited from the underlay RSVP-TE tunnels:

```
user@C1#show express-segments name gold-exp-seg-192.168.1.4 detail
Gold-Exp-Set-192.168.1.4
  To: 192.168.1.4, Set: gold-exp-set
  Status: Up (since 4d 11:09:05)
  Label: 19 (Route installed in mpls.0, TED entry added)
```

```
  LinkAttributes:
    ID: 2147483655
    TE-Metric: 10*, IGP-Metric: 30
    AdminGroups: gold, liquid-gold
    SRLGs: fiber-span-101
    BW: 1000Mbps
  UnderlayPaths:
    RSVP-LSP C1_to_C4_gold
      TE-Metric: 30, IGP-Metric: 30
      AdminGroups: gold
      SRLGs: fiber-span-101
      BW: 500Mbps
    RSVP-LSP C1_to_C4_liquid_gold
      TE-Metric: 30, IGP-Metric: 30
      AdminGroups: liquid-gold
      SRLGs: None
      BW: 500Mbps
```

You can observe the following in the output:

- Automatic naming of the express segment (`Gold-Exp-Set-192.168.1.4`).
- Traffic engineering attributes (bandwidth, metrics, admin groups, SRLGs) of the underlay RSVP-LSPs are inherited by the express segment.
- The express segment is an unnumbered traffic engineered link and has been added to the traffic engineering database.
- Label 19 has been assigned and installed in the `mpls.0` forwarding table as the adjacency SID for the SR virtual traffic engineering link.

The following is an example where SR-TE LSP destination is matched:

```
protocols {
      express-segments {
        segment-set set1sr {
            membership-policy expresspolsr1;
        }
        traffic-engineering;
    }
policy-options {
    policy-statement expresspolsr1 {
        from {
            protocol spring-te;
```

```
            route-filter 3.3.3.3/32 exact;
        }
        then accept;
```

In the following sample output, you can see the newly created express segment (**set1sr-3.3.3.3**) from the uncolored SR-TE underlay tunnels:

```
user@C1show express-segments detail

Name: set1sr-3.3.3.3
  To: 3.3.3.3, Type: Dynamic (Set: set1sr)
  Label: 16 (Route installed in mpls.0, TED entry added)
  Status: Up (ElapsedTime: 5d 20:37:08)
  LinkAttributes:
    LocalID: 2147483649
    TE-Metric: 20, IGP-Metric: 20
    BW: 0bps
  UnderlayPaths: 1
    SRTE LSP: lsp1to3_sr
      TE-Metric: 0, IGP-Metric: 0
      BW: 0bps
```

How are Express Segments Advertised?

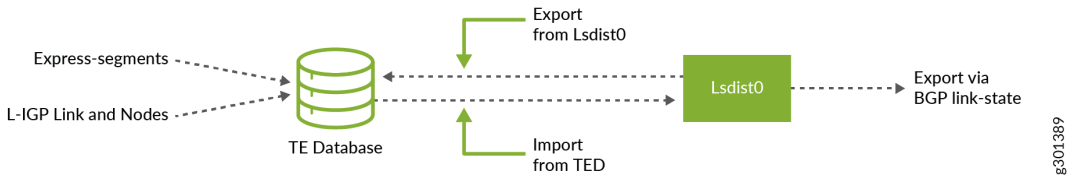
Express segments are advertised across domain boundaries or to higher-level controllers and Path Computing Elements (PCEs) using the BGP link state. When exchanging information through the BGP link state, the extensions for the BGP link state are used to advertise express segments as traffic engineered links. The express segment traffic engineered links and other normal traffic engineering links appear in the traffic engineering link-state database of any LSR in the network and are used for computing end-to-end traffic engineered paths. Express segment traffic engineering database entries are imported and exported from the **lsdist.0** table for advertisement through the BGP link state with the following traffic-engineering database import and export configuration:

```
protocols {
    mpls {
        traffic-engineering {
            database {
                import {
                    l3-unicast-topology {
                        bgp-link-state;
                        }
```

```
                    policy es_2_bgpls;
                }
                export {
                    policy bgpls_2_ted;
                }
            }
        }
    }
    bgp {
        group te-peers {
            family traffic-engineering {
                unicast;
                }
            export abstract-topo;
        }
    }
}
```

Figure 63 on page 933 provides a visual representation of how traffic engineering links and nodes are mirrored between the local traffic engineering database and the **lsdist.0** RIB that BGP-LS uses for advertisement. As illustrated, there are several policy attachment points.

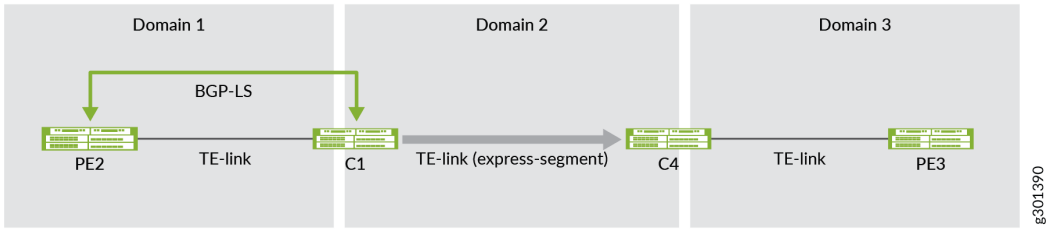
Figure 63: Advertising Express Segments



How are Express Segments Used by a Path Computing Element?

The BGP link state export policy is an effective place to create an abstract or customized topology that is advertised to a traffic engineered peer. For example, you may want to advertise only the express segment and Domain 3's TE links and nodes to PE2 such that the traffic engineered topology is abstracted as shown in Figure 64 on page 934. The abstracted view is then used by PE2 for end-to-end path computation.

Figure 64: Abstracting Traffic Engineered Domain 2 with Express Segment



The following is a sample configuration of a BGP link state export policy on C1:

```
policy-options {
    policy-statement abstract-topo {
        from {
            traffic-engineering {
                protocol express-segment;
                ipv4-prefix {
                    as 3;
                }
            }
        }
        then accept;
    }
}
```

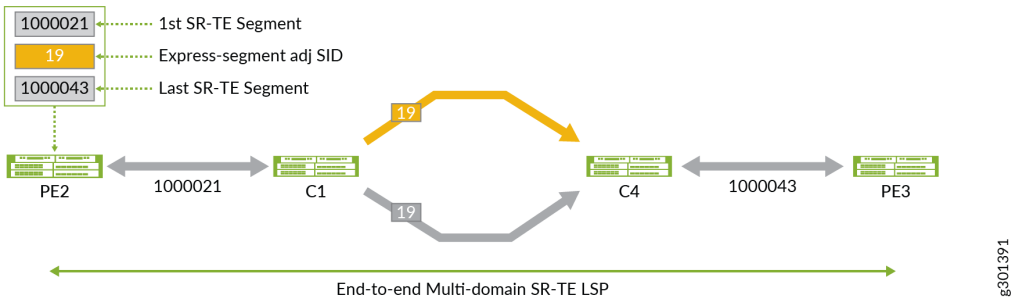
The following is a sample SR policy configuration on PE2 router to establish an end-to-end multi-domain path from PE2 to PE3:

```
protocols {
    source-packet-routing {
        source-routing-path pe2-to-pe3 {
            to 192.168.70.1;
            color 10;
            primary {
                sl1 {
                    compute {
                        profile_any-path;
                    }
                }
            }
```

```
        }
    }
}
```

The resulting end-to-end path is represented in Figure 65 on page 935. You can see the express segment's adjacency SID (label 19) is used in the SR segment-list resulting in traffic being load-balanced over both the gold and liquid-gold RSVP-TE LSPs within Domain 2.

Figure 65: Multi-domain End-to-End SR-TE LSP



Example: Inter-domain SR-TE Connectivity Using Express Segments Through RSVP-TE Underlay

IN THIS SECTION

- Requirements | 935
- Overview | 936
- Configuration | 940
- Verification | 1031

Use this example to learn how to establish an end-to-end inter-domain SR-TE connectivity using express segments.

Requirements

This example uses the following hardware and software components:

- MX Series routers as provider edge, border nodes, and intermediate routers.

- Junos OS Release 20.4R1 or later running on all devices.

Overview

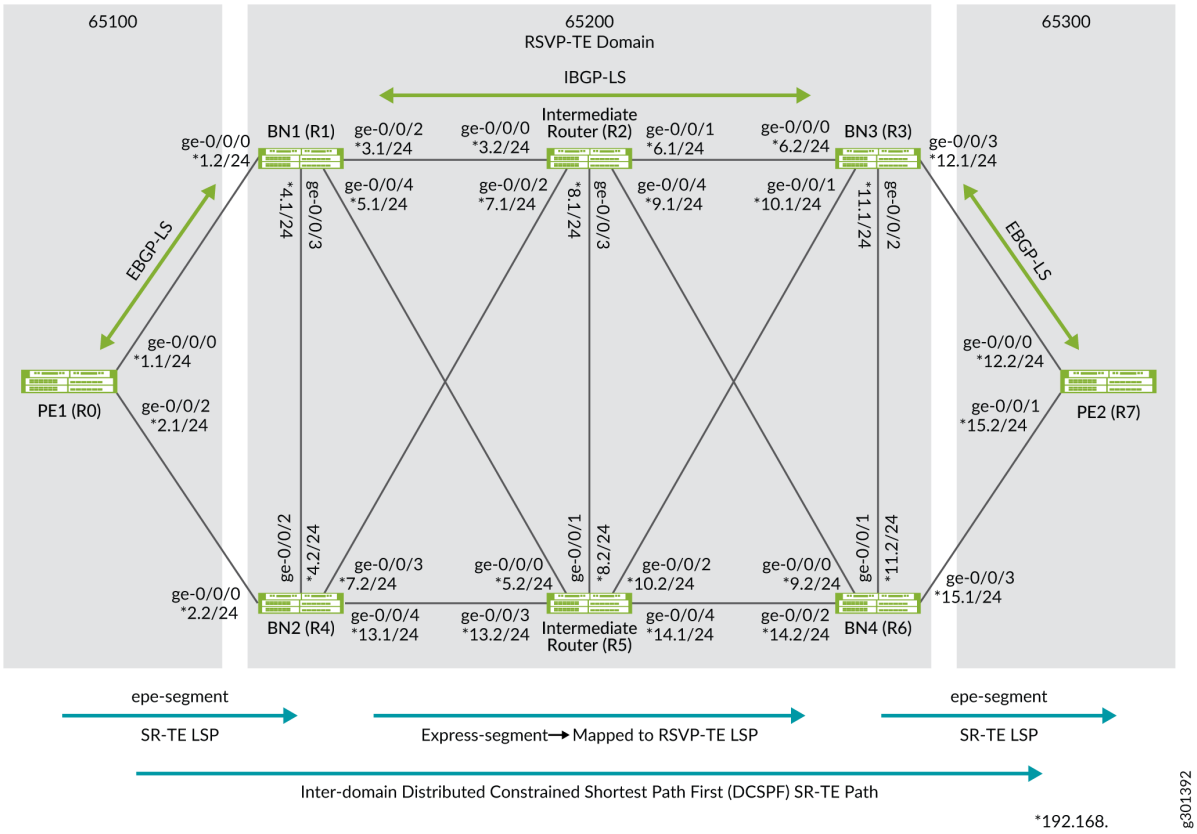
IN THIS SECTION

- Topology | 936

The following topology (Figure 66 on page 936) shows two SR-TE domains (AS100 and AS300) running EBGP-LS inter-connected through an RSVP-TE (AS200) domain:

Topology

Figure 66: Inter-domain SR-TE Connectivity Using Express Segments



In this topology, an end-to-end SR-TE path between PE1 router to PE2 router is established. Egress peer engineering (EPE) segments are defined on PE1 and PE2 routers to steer traffic towards their directly connected border nodes BN1/BN2 and BN3/BN4, respectively. EPE segments defined on the border nodes are advertised internally through the BGP link state. These two SR-TE domains are interconnected through the domain (AS200) that is leveraging RSVP-TE LSPs for internal path establishment.

The border nodes of the AS200 domain facilitate the abstraction of SR-TE information between domains. Express segments are created on border nodes (BN1, BN2, BN3, and BN4). Express segments are created in a one-on-one relationship with the underlying RSVP-TE LSPs and all express segments are inserted into the border node's local TE database for subsequent BGP link-state advertisement. The AS200 domain leverages RSVP-TE LSP underlays for TE management and presents those underlay RSVP-TE LSPs as express segments to the AS100 and the AS300 domains, enabling the domains to have end-to-end SR-TE LSP connectivity.

The following table describes the domains, routers, and connections in the topology:

Table 29: Describes the domains, routers, and connections in the Topology

Domain	Devices	Router ID/Lo) Address	Connection Details
AS65100 (EBGP-LS/ SR-TE LSP)	R0 (PE1 router)	10.100.100.100 10.100.100.101	Connected to R1 (BN1 router) through interface ge-0/0/0, assigned IP address 192.168.1.1/24. Connected to R4 (BN2 router) through interface ge-0/0/2, assigned IP address 192.168.2.1/24.
AS65200 (RSVP-TE LSP)	R1 (BN1 router)	10.1.1.1	Connected to R0 (PE1 router) through interface ge-0/0/0, assigned IP address 192.168.1.2/24. Connected to R4 (BN2 router) through interface ge-0/0/3, assigned IP address 192.168.4.1/24. Connected to R2 (Intermediate router) through interface ge-0/0/2, assigned IP address 192.168.3.1/24. Connected to R5 (Intermediate router) through interface ge-0/0/4, assigned IP address 192.168.5.1/24.

Table 29: Describes the domains, routers, and connections in the Topology *(Continued)*

Domain	Devices	Router ID/Lo) Address	Connection Details
	R4(BN2 router)	10.4.4.4	Connected to R0 (PE1 router) through interface ge-0/0/0, assigned IP address 192.168.2.2/24. Connected to R1 (BN1 router) through interface ge-0/0/2, assigned IP address 192.168.4.2/24. Connected to R2 (Intermediate router) through interface ge-0/0/3, assigned IP address 192.168.7.1/24. Connected to R5 (Intermediate router) through interface ge-0/0/4, assigned IP address 192.168.13.1/24.
	R2(Intermediate router)	10.2.2.2	Connected to R1 (BN1 router) through interface ge-0/0/0, assigned IP address 192.168.3.2/24. Connected to R4 (BN2 router) through interface ge-0/0/2, assigned IP address 192.168.7.1/24. Connected to R5 (Intermediate router) through interface ge-0/0/3, assigned IP address 192.168.8.1/24. Connected to R3 (BN3 router) through interface ge-0/0/1, assigned IP address 192.168.6.1/24. Connected to R6 (BN4 router) through interface ge-0/0/4, assigned IP address 192.168.9.1/24.

Table 29: Describes the domains, routers, and connections in the Topology *(Continued)*

Domain	Devices	Router ID/Lo) Address	Connection Details
	R5 (Intermediate router)	10.5.5.5	Connected to R1 (BN1 router) through interface ge-0/0/0, assigned IP address 192.168.5.2/24. Connected to R4 (BN2 router) through interface ge-0/0/3, assigned IP address 192.168.13.2/24. Connected to R2 (Intermediate router) through interface ge-0/0/1, assigned IP address 192.168.8.2/24. Connected to R3 (BN3 router) through interface ge-0/0/2, assigned IP address 192.168.10.2/24. Connected to R6 (BN4 router) through interface ge-0/0/4, assigned IP address 192.168.14.1/24.
	R3 (BN3 router)	10.3.3.3	Connected to R7 (PE2 router) through interface ge-0/0/3, assigned IP address 192.168.12.1/24. Connected to R6 (BN4 router) through interface ge-0/0/2, assigned IP address 192.168.11.1/24. Connected to R2 (Intermediate router) through interface ge-0/0/0, assigned IP address 192.168.6.2/24. Connected to R5 (Intermediate router) through interface ge-0/0/1, assigned IP address 192.168.10.1/24.

Table 29: Describes the domains, routers, and connections in the Topology *(Continued)*

Domain	Devices	Router ID/Lo) Address	Connection Details
	R6 (BN4 router)	10.6.6.6	Connected to R7 (PE2 router) through interface ge-0/0/3, assigned IP address 192.168.15.1/24. Connected to R3 (BN3 router) through interface ge-0/0/1, assigned IP address 192.168.11.2/24. Connected to R2 (Intermediate router) through interface ge-0/0/0, assigned IP address 192.168.9.2/24. Connected to R5 (Intermediate router) through interface ge-0/0/2, assigned IP address 192.168.14.2/24.
AS65300 (EBGP-LS/SR-TE LSP)	R7 (PE2 router)	10.7.7.7	Connected to R3 (BN3 router) through interface ge-0/0/0, assigned IP address 192.168.12.2/24. Connected to R6 (BN4 router) through interface ge-0/0/1, assigned IP address 192.168.15.2/24.

Configuration

IN THIS SECTION

- CLI Quick Configuration | **941**
- Configure R0 (PE1 router) | **957**
- Configure R1 (BN1 router) | **966**
- Configure R4 (BN2 router) | **977**
- Configure R2 (Intermediate router) | **987**
- Configure R5 (Intermediate router) | **995**
- Configure R3 (BN3 router) | **1003**
- Configure R6 (BN4 router) | **1013**
- Configure R7 (PE2 router) | **1023**

To inter-connect a multi-domain network and establish an end-to-end SR path using express segments, perform these tasks:

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

Device R0 (PE1 router)

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/0 description To_R1_1
set interfaces ge-0/0/0 unit 0 family inet address 192.168.1.1/24
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/2 description To_R4_1
set interfaces ge-0/0/2 unit 0 family inet address 192.168.2.1/24
set interfaces ge-0/0/2 unit 0 family iso
set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
set interfaces lo0 unit 0 family inet address 10.100.100.100/32
set interfaces lo0 unit 0 family inet address 10.100.100.101/32
set interfaces lo0 unit 0 family iso address 49.0001.000a.0a0a.0a00
set policy-options policy-statement nlri2bgp_epe term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp_epe term 1 from protocol bgp-ls-epe
set policy-options policy-statement nlri2bgp_epe term 1 then next-hop self
set policy-options policy-statement nlri2bgp_epe term 1 then accept
set policy-options policy-statement nlri2ted_bgp term 1 from protocol bgp
set policy-options policy-statement nlri2ted_bgp term 1 then accept
set policy-options policy-statement pplb then load-balance per-packet
set policy-options policy-statement ted2nlri term 1 from protocol bgp-ls-epe
set policy-options policy-statement ted2nlri term 1 then accept
set routing-options static route 10.7.7.71/32 next-hop 10.7.7.7
set routing-options static route 10.7.7.71/32 resolve
set routing-options router-id 10.100.100.100
set routing-options autonomous-system 65100
set routing-options forwarding-table ecmp-fast-reroute
set protocols bgp group ebgp1 type external
set protocols bgp group ebgp1 family inet unicast
set protocols bgp group ebgp1 family traffic-engineering unicast
set protocols bgp group ebgp1 export nlri2bgp_epe
set protocols bgp group ebgp1 neighbor 192.168.1.2 peer-as 65200
```

```
set protocols bgp group ebgp1 neighbor 192.168.1.2 egress-te-adj-segment epe_adj1_toR1 label 7101
set protocols bgp group ebgp1 neighbor 192.168.1.2 egress-te-adj-segment epe_adj1_toR1 next-hop
192.168.1.2
set protocols bgp group ebgp1 neighbor 192.168.1.2 egress-te-adj-segment epe_adj1_toR1 te-link-
attribute te-metric 20
set protocols bgp group ebgp1 neighbor 192.168.1.2 egress-te-adj-segment epe_adj1_toR1 te-link-
attribute igp-metric 10
set protocols bgp group ebgp1 neighbor 192.168.1.2 egress-te-adj-segment epe_adj1_toR1 te-link-
attribute admin-group red
set protocols bgp group ebgp1 neighbor 192.168.1.2 egress-te-adj-segment epe_adj1_toR1 te-link-
attribute admin-group brown
set protocols bgp group ebgp1 neighbor 192.168.2.2 peer-as 200
set protocols bgp group ebgp1 neighbor 192.168.2.2 egress-te-adj-segment epe_adj1_toR4 label 7104
set protocols bgp group ebgp1 neighbor 192.168.2.2 egress-te-adj-segment epe_adj1_toR4 next-hop
192.168.2.2
set protocols bgp group ebgp1 neighbor 192.168.2.2 egress-te-adj-segment epe_adj1_toR4 te-link-
attribute te-metric 20
set protocols bgp group ebgp1 neighbor 192.168.2.2 egress-te-adj-segment epe_adj1_toR4 te-link-
attribute igp-metric 10
set protocols bgp group ebgp1 neighbor 192.168.2.2 egress-te-adj-segment epe_adj1_toR4 te-link-
attribute admin-group red
set protocols bgp group ebgp1 neighbor 192.168.2.2 egress-te-adj-segment epe_adj1_toR4 te-link-
attribute admin-group brown
set protocols mpls traffic-engineering database import l3-unicast-topology bgp-link-state
set protocols mpls traffic-engineering database import policy ted2nlri
set protocols mpls traffic-engineering database export policy nlri2ted_bgp
set protocols mpls traffic-engineering database export l3-unicast-topology
set protocols mpls admin-groups red 0
set protocols mpls admin-groups blue 1
set protocols mpls admin-groups brown 5
set protocols mpls label-range static-label-range 7000 70000
set protocols mpls interface all
set protocols source-packet-routing compute-profile compute1 no-label-stack-compression
set protocols source-packet-routing compute-profile ecompute1 admin-group include-any red
set protocols source-packet-routing compute-profile ecompute1 admin-group include-any brown
set protocols source-packet-routing compute-profile ecompute1 no-label-stack-compression
set protocols source-packet-routing compute-profile ecompute2 admin-group include-any red
set protocols source-packet-routing compute-profile ecompute2 admin-group include-any blue
set protocols source-packet-routing compute-profile ecompute2 no-label-stack-compression
set protocols source-packet-routing source-routing-path computelsp1 to 10.7.7.7
set protocols source-packet-routing source-routing-path computelsp1 install 10.7.7.71
set protocols source-packet-routing source-routing-path computelsp1 primary p1 compute compute1
set protocols source-packet-routing source-routing-path ecomputelsp1 to 10.7.7.7
```

```
set protocols source-packet-routing source-routing-path ecomputelsp1 color 7000
set protocols source-packet-routing source-routing-path ecomputelsp1 primary p1 compute ecompute1
set protocols source-packet-routing source-routing-path ecomputelsp2 to 10.7.7.7
set protocols source-packet-routing source-routing-path ecomputelsp2 color 7001
set protocols source-packet-routing source-routing-path ecomputelsp2 primary p1 compute ecompute2
```

Device R1 (BN1 router)

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/0 description To_R0_1
set interfaces ge-0/0/0 unit 0 family inet address 192.168.1.2/24
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/2 description To_R2
set interfaces ge-0/0/2 unit 0 family inet address 192.168.3.1/24
set interfaces ge-0/0/2 unit 0 family iso
set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/3 description to-R4
set interfaces ge-0/0/3 unit 0 family inet address 192.168.4.1/24
set interfaces ge-0/0/3 unit 0 family iso
set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/4 description to-R5
set interfaces ge-0/0/4 unit 0 family inet address 192.168.5.1/24
set interfaces ge-0/0/4 unit 0 family iso
set interfaces ge-0/0/4 unit 0 family mpls maximum-labels 8
set interfaces lo0 unit 0 family inet address 10.1.1.1/32
set interfaces lo0 unit 0 family iso address 49.0001.0001.0101.0100
set policy-options policy-statement expresspol1 from route-filter 10.6.6.6/32 exact install-
nexthop lsp lsp1to6_a
set policy-options policy-statement expresspol1 then accept
set policy-options policy-statement expresspol2 from route-filter 10.3.3.3/32 exact install-
nexthop lsp lsp1to3_a
set policy-options policy-statement expresspol2 then accept
set policy-options policy-statement nlri2bgp_epe term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp_epe term 1 from protocol bgp-ls-epe
set policy-options policy-statement nlri2bgp_epe term 1 then next-hop self
set policy-options policy-statement nlri2bgp_epe term 1 then accept
set policy-options policy-statement nlri2bgp_stat term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp_stat term 1 from protocol express-segments
set policy-options policy-statement nlri2bgp_stat term 1 then accept
set policy-options policy-statement pplb then load-balance per-packet
set policy-options policy-statement ted2nlri_epe_stat term 1 from family traffic-engineering
```

```
set policy-options policy-statement ted2nlri_epe_stat term 1 from protocol express-segments
set policy-options policy-statement ted2nlri_epe_stat term 1 then accept
set policy-options policy-statement ted2nlri_epe_stat term 2 from family traffic-engineering
set policy-options policy-statement ted2nlri_epe_stat term 2 from protocol bgp-ls-epe
set policy-options policy-statement ted2nlri_epe_stat term 2 then accept
set policy-options policy-statement ted2nlri_epe_stat term 3 from protocol isis
set policy-options policy-statement ted2nlri_epe_stat term 3 then reject
set routing-options router-id 10.1.1.1
set routing-options autonomous-system 65200
set protocols bgp group ebgp1 type external
set protocols bgp group ebgp1 family inet-vpn unicast
set protocols bgp group ebgp1 family traffic-engineering unicast
set protocols bgp group ebgp1 export nlri2bgp_stat
set protocols bgp group ebgp1 neighbor 192.168.1.1 peer-as 65100
set protocols bgp group ebgp1 neighbor 192.168.1.1 egress-te-adj-segment epe_adj1_toR0 label 8110
set protocols bgp group ebgp1 neighbor 192.168.1.1 egress-te-adj-segment epe_adj1_toR0 next-hop
192.168.1.1
set protocols bgp group ebgp1 neighbor 192.168.1.1 egress-te-adj-segment epe_adj1_toR0 te-link-
attribute te-metric 20
set protocols bgp group ebgp1 neighbor 192.168.1.1 egress-te-adj-segment epe_adj1_toR0 te-link-
attribute igp-metric 10
set protocols bgp group ebgp1 neighbor 192.168.1.1 egress-te-adj-segment epe_adj1_toR0 te-link-
attribute admin-group red
set protocols bgp group ebgp1 neighbor 192.168.1.1 egress-te-adj-segment epe_adj1_toR0 te-link-
attribute admin-group brown
set protocols bgp group ibgp1 type internal
set protocols bgp group ibgp1 local-address 10.1.1.1
set protocols bgp group ibgp1 family traffic-engineering unicast
set protocols bgp group ibgp1 export nlri2bgp_epe
set protocols bgp group ibgp1 neighbor 10.2.2.2
set protocols bgp group ibgp1 neighbor 10.5.5.5
set protocols express-segments segment-template template1 admin-group red
set protocols express-segments segment-template template1 metric te 200
set protocols express-segments segment-template template1 metric igp 100
set protocols express-segments segment-set r1-exp-set1 membership-policy expresspol1
set protocols express-segments segment-set r1-exp-set1 template template1
set protocols express-segments segment-set r1-exp-set2 membership-policy expresspol2
set protocols express-segments traffic-engineering
set protocols isis interface ge-0/0/2.0
set protocols isis interface ge-0/0/3.0
set protocols isis interface ge-0/0/4.0
set protocols isis interface lo0.0 passive
set protocols isis level 1 disable
```

```
set protocols isis level 2 wide-metrics-only
set protocols mpls traffic-engineering database import l3-unicast-topology bgp-link-state
set protocols mpls traffic-engineering database import policy ted2nlri_epe_stat
set protocols mpls traffic-engineering database export l3-unicast-topology
set protocols mpls admin-groups red 0
set protocols mpls admin-groups blue 1
set protocols mpls admin-groups brown 5
set protocols mpls label-switched-path lsp1to6_a to 10.6.6.6
set protocols mpls label-switched-path lsp1to6_a admin-group include-any brown
set protocols mpls label-switched-path lsp1to6_a admin-group include-any red
set protocols mpls label-switched-path lsp1to6_b to 10.6.6.6
set protocols mpls label-switched-path lsp1to6_b admin-group include-any brown
set protocols mpls label-switched-path lsp1to6_b admin-group include-any blue
set protocols mpls label-switched-path lsp1to6_c to 10.6.6.6
set protocols mpls label-switched-path lsp1to6_c admin-group include-any blue
set protocols mpls label-switched-path lsp1to3_a to 10.3.3.3
set protocols mpls label-switched-path lsp1to3_a admin-group include-any brown
set protocols mpls label-switched-path lsp1to3_a admin-group include-any red
set protocols mpls label-switched-path lsp1to3_b to 10.3.3.3
set protocols mpls label-switched-path lsp1to3_b admin-group include-any blue
set protocols mpls label-range static-label-range 7000 70000
set protocols mpls interface ge-0/0/3.0 admin-group red
set protocols mpls interface ge-0/0/2.0 admin-group brown
set protocols mpls interface ge-0/0/4.0 admin-group blue
set protocols mpls interface all
set protocols rsvp interface all link-protection
```

Device R4 (BN2 router)

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/0 description To_R0
set interfaces ge-0/0/0 unit 0 family inet address 192.168.2.2/24
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/2 description To_R1
set interfaces ge-0/0/2 unit 0 family inet address 192.168.4.2/24
set interfaces ge-0/0/2 unit 0 family iso
set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/3 description To_R2
set interfaces ge-0/0/3 unit 0 family inet address 192.168.7.2/24
set interfaces ge-0/0/3 unit 0 family iso
set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
```

```
set interfaces ge-0/0/4 description To_R5
set interfaces ge-0/0/4 unit 0 family inet address 192.168.13.1/24
set interfaces ge-0/0/4 unit 0 family iso
set interfaces ge-0/0/4 unit 0 family mpls maximum-labels 8
set interfaces lo0 unit 0 family inet address 10.4.4.4/32
set interfaces lo0 unit 0 family iso address 49.0001.0004.0404.0400
set policy-options policy-statement expresspol1 from route-filter 10.6.6.6/32 exact install-
nexthop lsp lsp4to6_a
set policy-options policy-statement expresspol1 then accept
set policy-options policy-statement expresspol2 from route-filter 10.3.3.3/32 exact install-
nexthop lsp lsp4to3_a
set policy-options policy-statement expresspol2 then accept
set policy-options policy-statement nlri2bgp_epe term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp_epe term 1 from protocol bgp-ls-epe
set policy-options policy-statement nlri2bgp_epe term 1 then next-hop self
set policy-options policy-statement nlri2bgp_epe term 1 then accept
set policy-options policy-statement nlri2bgp_stat term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp_stat term 1 from protocol express-segments
set policy-options policy-statement nlri2bgp_stat term 1 then accept
set policy-options policy-statement pplb then load-balance per-packet
set policy-options policy-statement ted2nlri_epe_stat term 1 from family traffic-engineering
set policy-options policy-statement ted2nlri_epe_stat term 1 from protocol express-segments
set policy-options policy-statement ted2nlri_epe_stat term 1 then accept
set policy-options policy-statement ted2nlri_epe_stat term 2 from family traffic-engineering
set policy-options policy-statement ted2nlri_epe_stat term 2 from protocol bgp-ls-epe
set policy-options policy-statement ted2nlri_epe_stat term 2 then accept
set policy-options policy-statement ted2nlri_epe_stat term 3 from protocol isis
set policy-options policy-statement ted2nlri_epe_stat term 3 then reject
set routing-options router-id 10.4.4.4
set routing-options autonomous-system 65200
set protocols bgp group ibgp1 type internal
set protocols bgp group ibgp1 local-address 10.4.4.4
set protocols bgp group ibgp1 family traffic-engineering unicast
set protocols bgp group ibgp1 export nlri2bgp_epe
set protocols bgp group ibgp1 neighbor 10.2.2.2
set protocols bgp group ibgp1 neighbor 10.5.5.5
set protocols bgp group ebgp1 type external
set protocols bgp group ebgp1 family inet-vpn unicast
set protocols bgp group ebgp1 family traffic-engineering unicast
set protocols bgp group ebgp1 export nlri2bgp_stat
set protocols bgp group ebgp1 neighbor 192.168.2.1 peer-as 65100
set protocols bgp group ebgp1 neighbor 192.168.2.1 egress-te-adj-segment epe_adj1_toR0 label 8140
set protocols bgp group ebgp1 neighbor 192.168.2.1 egress-te-adj-segment epe_adj1_toR0 next-hop
```

```
192.168.2.1
set protocols bgp group ebgp1 neighbor 192.168.2.1 egress-te-adj-segment epe_adj1_toR0 te-link-
attribute te-metric 20
set protocols bgp group ebgp1 neighbor 192.168.2.1 egress-te-adj-segment epe_adj1_toR0 te-link-
attribute igp-metric 10
set protocols bgp group ebgp1 neighbor 192.168.2.1 egress-te-adj-segment epe_adj1_toR0 te-link-
attribute admin-group red
set protocols bgp group ebgp1 neighbor 192.168.2.1 egress-te-adj-segment epe_adj1_toR0 te-link-
attribute admin-group brown
set protocols express-segments segment-set r4-exp-set1 membership-policy expresspol1
set protocols express-segments segment-set r4-exp-set2 membership-policy expresspol2
set protocols express-segments traffic-engineering
set protocols isis interface ge-0/0/0.0
set protocols isis interface ge-0/0/2.0
set protocols isis interface ge-0/0/3.0
set protocols isis interface ge-0/0/4.0
set protocols isis interface lo0.0 passive
set protocols isis level 1 disable
set protocols isis level 2 wide-metrics-only
set protocols mpls traffic-engineering database import l3-unicast-topology bgp-link-state
set protocols mpls traffic-engineering database import policy ted2nlri_epe_stat
set protocols mpls traffic-engineering database export l3-unicast-topology
set protocols mpls admin-groups red 0
set protocols mpls admin-groups blue 1
set protocols mpls admin-groups brown 5
set protocols mpls label-switched-path lsp4to6_a to 10.6.6.6
set protocols mpls label-switched-path lsp4to6_a admin-group include-any brown
set protocols mpls label-switched-path lsp4to6_a admin-group include-any red
set protocols mpls label-switched-path lsp4to6_b to 10.6.6.6
set protocols mpls label-switched-path lsp4to6_b admin-group include-any blue
set protocols mpls label-switched-path lsp4to3_a to 10.3.3.3
set protocols mpls label-switched-path lsp4to3_a admin-group include-any brown
set protocols mpls label-switched-path lsp4to3_a admin-group include-any red
set protocols mpls label-switched-path lsp4to3_b to 10.3.3.3
set protocols mpls label-switched-path lsp4to3_b admin-group include-any brown
set protocols mpls label-switched-path lsp4to3_c to 10.3.3.3
set protocols mpls label-switched-path lsp4to3_c admin-group include-any brown
set protocols mpls label-range static-label-range 7000 70000
set protocols mpls interface ge-0/0/2.0 admin-group red
set protocols mpls interface ge-0/0/4.0 admin-group brown
set protocols mpls interface all
set protocols rsvp interface all link-protection
```

Device R2 (Intermediate router)

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/0 description To_R1
set interfaces ge-0/0/0 unit 0 family inet address 192.168.3.2/24
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/1 description To_R3
set interfaces ge-0/0/1 unit 0 family inet address 192.168.6.1/24
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/2 description To_R4
set interfaces ge-0/0/2 unit 0 family inet address 192.168.7.1/24
set interfaces ge-0/0/2 unit 0 family iso
set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/3 description To_R5
set interfaces ge-0/0/3 unit 0 family inet address 192.168.8.1/24
set interfaces ge-0/0/3 unit 0 family iso
set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/4 description To_R6
set interfaces ge-0/0/4 unit 0 family inet address 192.168.9.1/24
set interfaces ge-0/0/4 unit 0 family iso
set interfaces ge-0/0/4 unit 0 family mpls maximum-labels 8
set interfaces lo0 unit 0 family inet address 10.2.2.2/32
set interfaces lo0 unit 0 family iso address 49.0001.0002.0202.0200
set policy-options policy-statement bgplsepe_rt_2_ted term 1 from protocol bgp
set policy-options policy-statement bgplsepe_rt_2_ted term 1 then accept
set policy-options policy-statement nlri2bgp term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp term 1 then next-hop self
set policy-options policy-statement nlri2bgp term 1 then accept
set policy-options policy-statement nlri2bgp_igp term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp_igp term 1 from protocol isis
set policy-options policy-statement nlri2bgp_igp term 1 then accept
set policy-options policy-statement nlri2ted_igp term 1 from traffic-engineering protocol isis-
level-2
set policy-options policy-statement nlri2ted_igp term 1 then accept
set policy-options policy-statement pplb then load-balance per-packet
set policy-options policy-statement ted2nlri term 1 from protocol bgp-ls-epe
set policy-options policy-statement ted2nlri term 1 then accept
set policy-options policy-statement ted2nlri_1 term 1 from traffic-engineering
set policy-options policy-statement ted2nlri_1 term 1 then accept
set policy-options policy-statement ted2nlri_igp term 1 from family traffic-engineering
```

```
set policy-options policy-statement ted2nlri_igp term 1 from protocol isis
set policy-options policy-statement ted2nlri_igp term 1 then accept
set routing-options router-id 10.2.2.2
set routing-options autonomous-system 65200
set protocols bgp group RR1 type internal
set protocols bgp group RR1 local-address 10.2.2.2
set protocols bgp group RR1 family traffic-engineering unicast
set protocols bgp group RR1 neighbor 10.1.1.1
set protocols bgp group RR1 neighbor 10.3.3.3
set protocols bgp group RR1 neighbor 10.6.6.6
set protocols bgp group RR1 neighbor 10.4.4.4
set protocols bgp cluster 10.2.2.2
set protocols isis interface ge-0/0/0.0
set protocols isis interface ge-0/0/1.0
set protocols isis interface ge-0/0/2.0
set protocols isis interface ge-0/0/3.0
set protocols isis interface ge-0/0/4.0
set protocols isis interface lo0.0 passive
set protocols isis level 1 disable
set protocols isis level 2 wide-metrics-only
set protocols mpls admin-groups red 0
set protocols mpls admin-groups blue 1
set protocols mpls admin-groups brown 5
set protocols mpls label-range static-label-range 7000 70000
set protocols mpls interface ge-0/0/0.0 admin-group brown
set protocols mpls interface ge-0/0/3.0 admin-group red
set protocols mpls interface ge-0/0/4.0 admin-group blue
set protocols mpls interface ge-0/0/1.0 admin-group brown
set protocols mpls interface all
set protocols rsvp interface all link-protection
```

Device R5 (Intermediate router)

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/0 description To_R1
set interfaces ge-0/0/0 unit 0 family inet address 192.168.5.2/24
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/1 description To_R2
set interfaces ge-0/0/1 unit 0 family inet address 192.168.8.2/24
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family mpls maximum-labels 8
```

```
set interfaces ge-0/0/2 description To_R3
set interfaces ge-0/0/2 unit 0 family inet address 192.168.10.2/24
set interfaces ge-0/0/2 unit 0 family iso
set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/3 description To_R4
set interfaces ge-0/0/3 unit 0 family inet address 192.168.13.2/24
set interfaces ge-0/0/3 unit 0 family iso
set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/4 description To_R6
set interfaces ge-0/0/4 unit 0 family inet address 192.168.14.1/24
set interfaces ge-0/0/4 unit 0 family iso
set interfaces ge-0/0/4 unit 0 family mpls maximum-labels 8
set interfaces lo0 unit 0 family inet address 10.5.5.5/32
set interfaces lo0 unit 0 family iso address 49.0001.0005.0505.0500
set policy-options policy-statement nlri2bgp term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp term 1 then next-hop self
set policy-options policy-statement nlri2bgp term 1 then accept
set policy-options policy-statement nlri2ted_igp term 1 from traffic-engineering protocol isis-
level-2
set policy-options policy-statement nlri2ted_igp term 1 then accept
set policy-options policy-statement pplb then load-balance per-packet
set policy-options policy-statement ted2nlri term 1 from protocol bgp-ls-epe
set policy-options policy-statement ted2nlri term 1 then accept
set policy-options policy-statement ted2nlri_igp term 1 from family traffic-engineering
set policy-options policy-statement ted2nlri_igp term 1 from protocol isis
set policy-options policy-statement ted2nlri_igp term 1 then accept
set routing-options router-id 10.5.5.5
set routing-options autonomous-system 65200
set protocols bgp group RR2 type internal
set protocols bgp group RR2 family inet unicast
set protocols bgp group RR2 family traffic-engineering unicast
set protocols bgp group RR2 neighbor 10.1.1.1
set protocols bgp group RR2 neighbor 10.3.3.3
set protocols bgp group RR2 neighbor 10.6.6.6
set protocols bgp group RR2 neighbor 10.4.4.4
set protocols bgp cluster 10.5.5.5
set protocols isis interface ge-0/0/0.0
set protocols isis interface ge-0/0/1.0
set protocols isis interface ge-0/0/2.0
set protocols isis interface ge-0/0/3.0
set protocols isis interface ge-0/0/4.0
set protocols isis interface lo0.0 passive
set protocols isis level 1 disable
```

```
set protocols isis level 2 wide-metrics-only
set protocols mpls admin-groups red 0
set protocols mpls admin-groups blue 1
set protocols mpls admin-groups brown 5
set protocols mpls label-range static-label-range 7000 70000
set protocols mpls interface ge-0/0/0.0 admin-group blue
set protocols mpls interface ge-0/0/1.0 admin-group red
set protocols mpls interface ge-0/0/3.0 admin-group brown
set protocols mpls interface ge-0/0/4.0 admin-group brown
set protocols mpls interface all
set protocols rsvp interface all link-protection
```

Device R3 (BN3 router)

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/0 description To_R2
set interfaces ge-0/0/0 unit 0 family inet address 192.168.6.2/24
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/1 description To_R5
set interfaces ge-0/0/1 unit 0 family inet address 192.168.10.1/24
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/2 description To_R6
set interfaces ge-0/0/2 unit 0 family inet address 192.168.11.1/24
set interfaces ge-0/0/2 unit 0 family iso
set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/3 description To_R7
set interfaces ge-0/0/3 unit 0 family inet address 192.168.12.1/24
set interfaces ge-0/0/3 unit 0 family iso
set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
set interfaces lo0 unit 0 family inet address 10.3.3.3/32
set interfaces lo0 unit 0 family iso address 49.0001.0003.0303.0300
set policy-options policy-statement expresspol1 from route-filter 10.1.1.1/32 exact install-
nexthop lsp lsp3to1_a
set policy-options policy-statement expresspol1 then accept
set policy-options policy-statement expresspol2 from route-filter 10.4.4.4/32 exact install-
nexthop lsp lsp3to4_a
set policy-options policy-statement expresspol2 then accept
set policy-options policy-statement nlri2bgp_epe term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp_epe term 1 from protocol bgp-ls-epe
set policy-options policy-statement nlri2bgp_epe term 1 then next-hop self
```

```
set policy-options policy-statement nlri2bgp_epe term 1 then accept
set policy-options policy-statement nlri2bgp_stat term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp_stat term 1 from protocol express-segments
set policy-options policy-statement nlri2bgp_stat term 1 then accept
set policy-options policy-statement pplb then load-balance per-packet
set policy-options policy-statement ted2nlri_epe_stat term 1 from family traffic-engineering
set policy-options policy-statement ted2nlri_epe_stat term 1 from protocol static
set policy-options policy-statement ted2nlri_epe_stat term 1 then accept
set policy-options policy-statement ted2nlri_epe_stat term 2 from family traffic-engineering
set policy-options policy-statement ted2nlri_epe_stat term 2 from protocol bgp-ls-epe
set policy-options policy-statement ted2nlri_epe_stat term 2 then accept
set policy-options policy-statement ted2nlri_epe_stat term 3 from protocol isis
set policy-options policy-statement ted2nlri_epe_stat term 3 then reject
set routing-options router-id 10.3.3.3
set routing-options autonomous-system 65200
set protocols bgp group ibgp1 type internal
set protocols bgp group ibgp1 local-address 10.3.3.3
set protocols bgp group ibgp1 family traffic-engineering unicast
set protocols bgp group ibgp1 export nlri2bgp_epe
set protocols bgp group ibgp1 neighbor 10.2.2.2
set protocols bgp group ibgp1 neighbor 10.5.5.5
set protocols bgp group ebgp1 type external
set protocols bgp group ebgp1 family traffic-engineering unicast
set protocols bgp group ebgp1 export nlri2bgp_stat
set protocols bgp group ebgp1 neighbor 192.168.12.2 peer-as 65300
set protocols bgp group ebgp1 neighbor 192.168.12.2 egress-te-adj-segment epe_adj1_toR7 label
7137
set protocols bgp group ebgp1 neighbor 192.168.12.2 egress-te-adj-segment epe_adj1_toR7 next-hop
192.168.12.2
set protocols bgp group ebgp1 neighbor 192.168.12.2 egress-te-adj-segment epe_adj1_toR7 te-link-
attribute te-metric 20
set protocols bgp group ebgp1 neighbor 192.168.12.2 egress-te-adj-segment epe_adj1_toR7 te-link-
attribute igp-metric 10
set protocols bgp group ebgp1 neighbor 192.168.12.2 egress-te-adj-segment epe_adj1_toR7 te-link-
attribute admin-group red
set protocols bgp group ebgp1 neighbor 192.168.12.2 egress-te-adj-segment epe_adj1_toR7 te-link-
attribute admin-group brown
set protocols bgp group ebgp1 vpn-apply-export
set protocols express-segments segment-set set1 membership-policy expresspol1
set protocols express-segments segment-set set2 membership-policy expresspol2
set protocols express-segments traffic-engineering
set protocols isis interface ge-0/0/0.0
set protocols isis interface ge-0/0/1.0
```

```
set protocols isis interface ge-0/0/2.0
set protocols isis interface ge-0/0/3.0 passive
set protocols isis interface lo0.0 passive
set protocols isis level 1 disable
set protocols isis level 2 wide-metrics-only
set protocols mpls traffic-engineering database import l3-unicast-topology bgp-link-state
set protocols mpls traffic-engineering database import policy ted2nlri_epe_stat
set protocols mpls traffic-engineering database export l3-unicast-topology
set protocols mpls admin-groups red 0
set protocols mpls admin-groups blue 1
set protocols mpls admin-groups brown 5
set protocols mpls label-switched-path lsp3to1_a to 10.1.1.1
set protocols mpls label-switched-path lsp3to1_a admin-group include-any red
set protocols mpls label-switched-path lsp3to1_a admin-group include-any brown
set protocols mpls label-switched-path lsp3to4_a to 10.4.4.4
set protocols mpls label-switched-path lsp3to4_a admin-group include-any red
set protocols mpls label-switched-path lsp3to4_a admin-group include-any brown
set protocols mpls label-range static-label-range 7000 70000
set protocols mpls interface ge-0/0/0.0 admin-group brown
set protocols mpls interface ge-0/0/2.0 admin-group red
set protocols mpls interface ge-0/0/3.0 admin-group red
set protocols mpls interface ge-0/0/3.0 admin-group brown
set protocols mpls interface all
set protocols rsvp interface all link-protection
```

Device R6 (BN4 router)

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/0 description To_R2
set interfaces ge-0/0/0 unit 0 family inet address 192.168.9.2/24
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/1 description To_R3
set interfaces ge-0/0/1 unit 0 family inet address 192.168.11.2/24
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/2 description To_R5
set interfaces ge-0/0/2 unit 0 family inet address 192.168.14.2/24
set interfaces ge-0/0/2 unit 0 family iso
set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/3 description To_R7
set interfaces ge-0/0/3 unit 0 family inet address 192.168.15.1/24
```

```
set interfaces ge-0/0/3 unit 0 family iso
set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
set interfaces lo0 unit 0 family inet address 10.6.6.6/32
set interfaces lo0 unit 0 family iso address 49.0001.0006.0606.0600
set policy-options policy-statement expresspol1 from route-filter 10.1.1.1/32 exact install-
nexthop lsp lsp6to1_a
set policy-options policy-statement expresspol1 then accept
set policy-options policy-statement expresspol2 from route-filter 10.4.4.4/32 exact install-
nexthop lsp lsp6to4_a
set policy-options policy-statement expresspol2 then accept
set policy-options policy-statement nlri2bgp_epe term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp_epe term 1 from protocol bgp-ls-epe
set policy-options policy-statement nlri2bgp_epe term 1 then next-hop self
set policy-options policy-statement nlri2bgp_epe term 1 then accept
set policy-options policy-statement nlri2bgp_stat term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp_stat term 1 from protocol express-segments
set policy-options policy-statement nlri2bgp_stat term 1 then accept
set policy-options policy-statement pplb then load-balance per-packet
set policy-options policy-statement ted2nlri_epe_stat term 1 from family traffic-engineering
set policy-options policy-statement ted2nlri_epe_stat term 1 from protocol static
set policy-options policy-statement ted2nlri_epe_stat term 1 then accept
set policy-options policy-statement ted2nlri_epe_stat term 2 from family traffic-engineering
set policy-options policy-statement ted2nlri_epe_stat term 2 from protocol bgp-ls-epe
set policy-options policy-statement ted2nlri_epe_stat term 2 then accept
set policy-options policy-statement ted2nlri_epe_stat term 3 from protocol isis
set policy-options policy-statement ted2nlri_epe_stat term 3 then reject
set routing-options router-id 10.6.6.6
set routing-options autonomous-system 65200
set protocols bgp group ibgp1 type internal
set protocols bgp group ibgp1 local-address 10.6.6.6
set protocols bgp group ibgp1 family traffic-engineering unicast
set protocols bgp group ibgp1 export nlri2bgp_epe
set protocols bgp group ibgp1 neighbor 10.2.2.2
set protocols bgp group ibgp1 neighbor 10.5.5.5
set protocols bgp group ebgp1 type external
set protocols bgp group ebgp1 family traffic-engineering unicast
set protocols bgp group ebgp1 export nlri2bgp_stat
set protocols bgp group ebgp1 neighbor 192.168.15.2 peer-as 65300
set protocols bgp group ebgp1 neighbor 192.168.15.2 egress-te-adj-segment epe_adj1_toR7 label
7167
set protocols bgp group ebgp1 neighbor 192.168.15.2 egress-te-adj-segment epe_adj1_toR7 next-hop
192.168.15.2
set protocols bgp group ebgp1 neighbor 192.168.15.2 egress-te-adj-segment epe_adj1_toR7 te-link-
```

```
attribute te-metric 20
set protocols bgp group ebgp1 neighbor 192.168.15.2 egress-te-adj-segment epe_adj1_toR7 te-link-
attribute igp-metric 10
set protocols bgp group ebgp1 neighbor 192.168.15.2 egress-te-adj-segment epe_adj1_toR7 te-link-
attribute admin-group red
set protocols bgp group ebgp1 neighbor 192.168.15.2 egress-te-adj-segment epe_adj1_toR7 te-link-
attribute admin-group brown
set protocols express-segments segment-set set1 membership-policy expresspol1
set protocols express-segments segment-set set2 membership-policy expresspol2
set protocols express-segments traffic-engineering
set protocols isis interface ge-0/0/0.0
set protocols isis interface ge-0/0/1.0
set protocols isis interface ge-0/0/2.0
set protocols isis interface lo0.0 passive
set protocols isis level 1 disable
set protocols isis level 2 wide-metrics-only
set protocols mpls traffic-engineering database import l3-unicast-topology bgp-link-state
set protocols mpls traffic-engineering database import policy ted2nlri_epe_stat
set protocols mpls traffic-engineering database export l3-unicast-topology
set protocols mpls admin-groups red 0
set protocols mpls admin-groups blue 1
set protocols mpls admin-groups brown 5
set protocols mpls label-switched-path lsp6to1_a to 10.1.1.1
set protocols mpls label-switched-path lsp6to1_a admin-group include-any red
set protocols mpls label-switched-path lsp6to1_a admin-group include-any brown
set protocols mpls label-switched-path lsp6to4_a to 10.4.4.4
set protocols mpls label-switched-path lsp6to4_a admin-group include-any red
set protocols mpls label-switched-path lsp6to4_a admin-group include-any brown
set protocols mpls label-range static-label-range 7000 70000
set protocols mpls interface ge-0/0/0.0 admin-group blue
set protocols mpls interface ge-0/0/1.0 admin-group red
set protocols mpls interface ge-0/0/2.0 admin-group brown
set protocols mpls interface ge-0/0/3.0 admin-group red
set protocols mpls interface ge-0/0/3.0 admin-group brown
set protocols mpls interface all
set protocols rsvp interface all link-protection
```

Device R7 (PE2 router)

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/0 description To_R3
set interfaces ge-0/0/0 unit 0 family inet address 192.168.12.2/24
```

```
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/1 description To_R6
set interfaces ge-0/0/1 unit 0 family inet address 192.168.15.2/24
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family mpls maximum-labels 8
set interfaces lo0 unit 0 family inet address 10.7.7.7/32
set interfaces lo0 unit 0 family inet address 10.7.7.71/32
set interfaces lo0 unit 0 family iso address 49.0001.0007.0707.0700
set policy-options policy-statement nlri2bgp_epe term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp_epe term 1 from protocol bgp-ls-epe
set policy-options policy-statement nlri2bgp_epe term 1 then next-hop self
set policy-options policy-statement nlri2bgp_epe term 1 then accept
set policy-options policy-statement nlri2ted_bgp term 1 from protocol bgp
set policy-options policy-statement nlri2ted_bgp term 1 then accept
set policy-options policy-statement pplb then load-balance per-packet
set policy-options policy-statement ted2nlri term 1 from protocol bgp-ls-epe
set policy-options policy-statement ted2nlri term 1 then accept
set policy-options resolution-map map1 mode ip-color
set routing-options static route 10.100.100.101/32 next-hop 10.100.100.100
set routing-options static route 10.100.100.101/32 resolve
set routing-options router-id 10.7.7.7
set routing-options autonomous-system 65300
set protocols bgp group ebgp1 type external
set protocols bgp group ebgp1 family inet unicast
set protocols bgp group ebgp1 family traffic-engineering unicast
set protocols bgp group ebgp1 export nlri2bgp_epe
set protocols bgp group ebgp1 neighbor 192.168.12.1 peer-as 65200
set protocols bgp group ebgp1 neighbor 192.168.12.1 egress-te-adj-segment epe_adj1_toR3 label
8173
set protocols bgp group ebgp1 neighbor 192.168.12.1 egress-te-adj-segment epe_adj1_toR3 next-hop
192.168.12.1
set protocols bgp group ebgp1 neighbor 192.168.12.1 egress-te-adj-segment epe_adj1_toR3 te-link-
attribute te-metric 20
set protocols bgp group ebgp1 neighbor 192.168.12.1 egress-te-adj-segment epe_adj1_toR3 te-link-
attribute igp-metric 10
set protocols bgp group ebgp1 neighbor 192.168.12.1 egress-te-adj-segment epe_adj1_toR3 te-link-
attribute admin-group red
set protocols bgp group ebgp1 neighbor 192.168.12.1 egress-te-adj-segment epe_adj1_toR3 te-link-
attribute admin-group brown
set protocols bgp group ebgp1 neighbor 192.168.15.1 peer-as 200
set protocols bgp group ebgp1 neighbor 192.168.15.1 egress-te-adj-segment epe_adj1_toR6 label
8176
```

```
set protocols bgp group ebgp1 neighbor 192.168.15.1 egress-te-adj-segment epe_adj1_toR6 next-hop
192.168.15.1
set protocols bgp group ebgp1 neighbor 192.168.15.1 egress-te-adj-segment epe_adj1_toR6 te-link-
attribute te-metric 20
set protocols bgp group ebgp1 neighbor 192.168.15.1 egress-te-adj-segment epe_adj1_toR6 te-link-
attribute igp-metric 10
set protocols bgp group ebgp1 neighbor 192.168.15.1 egress-te-adj-segment epe_adj1_toR6 te-link-
attribute admin-group red
set protocols bgp group ebgp1 neighbor 192.168.15.1 egress-te-adj-segment epe_adj1_toR6 te-link-
attribute admin-group brown
set protocols mpls traffic-engineering database import l3-unicast-topology bgp-link-state
set protocols mpls traffic-engineering database import policy ted2nlri
set protocols mpls traffic-engineering database export policy nlri2ted_bgp
set protocols mpls traffic-engineering database export l3-unicast-topology
set protocols mpls admin-groups red 0
set protocols mpls admin-groups blue 1
set protocols mpls admin-groups brown 5
set protocols mpls label-range static-label-range 7000 70000
set protocols mpls interface all
set protocols source-packet-routing compute-profile compute1 no-label-stack-compression
set protocols source-packet-routing source-routing-path computelsp1 to 10.100.100.100
set protocols source-packet-routing source-routing-path computelsp1 install 10.100.100.101
set protocols source-packet-routing source-routing-path computelsp1 primary p1 compute compute1
```

Configure R0 (PE1 router)

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure device R0:

1. Configure the network services mode as Enhanced IP. Enhanced IP sets the router's network services to enhanced Internet Protocol and uses enhanced mode capabilities.

```
[edit]
user@R0#set chassis network-services enhanced-ip
```

After you configure the `enhanced-ip` statement and commit the configuration, the following warning message appears prompting you to reboot the router:

```
'chassis'
WARNING: Chassis configuration for network services has been changed. A system reboot is
mandatory. Please reboot the system NOW. Continuing without a reboot might result in
unexpected system behavior.
commit complete
```

The reboot brings up the FPCs on the router.

2. Configure the interfaces to enable IP, MPLS, and ISO transport.

```
[edit]
user@R0#set interfaces ge-0/0/0 description To_R1_1
user@R0#set interfaces ge-0/0/0 unit 0 family inet address 192.168.1.1/24
user@R0#set interfaces ge-0/0/0 unit 0 family iso
user@R0#set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 8
user@R0#set interfaces ge-0/0/2 description To_R4_1
user@R0#set interfaces ge-0/0/2 unit 0 family inet address 192.168.2.1/24
user@R0#set interfaces ge-0/0/2 unit 0 family iso
user@R0#set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
```

3. Configure the loopback interface to enable tunnel endpoints and service endpoints.

```
[edit]
user@R0#set interfaces lo0 unit 0 family inet address 10.100.100.100/32
user@R0#set interfaces lo0 unit 0 family inet address 10.100.100.101/32
user@R0#set interfaces lo0 unit 0 family iso address 49.0001.000a.0a0a.0a00
```

4. Configure routing options to identify the router in the domain.

```
[edit]
user@R0#set routing-options router-id 10.100.100.100
user@R0#set routing-options autonomous-system 65100
user@R0#set routing-options static route 10.7.7.71/32 next-hop 10.7.7.7
user@R0#set routing-options static route 10.7.7.71/32 resolve
```

5. Define import and export policies. For example, configure policies that export EPE TE links from the local TE database to **lsdist.0** and policies to import from **lsdist.0** into the local TE database. You can configure policies to advertise the BGP routes to a peer.

```
[edit]
user@R0#set policy-options policy-statement nlri2bgp_epe term 1 from family traffic-
engineering
user@R0#set policy-options policy-statement nlri2bgp_epe term 1 from protocol bgp-ls-epe
user@R0#set policy-options policy-statement nlri2bgp_epe term 1 then next-hop self
user@R0#set policy-options policy-statement nlri2bgp_epe term 1 then accept
user@R0#set policy-options policy-statement nlri2ted_bgp term 1 from protocol bgp
user@R0#set policy-options policy-statement nlri2ted_bgp term 1 then accept
user@R0#set policy-options policy-statement pplb then load-balance per-packet
user@R0#set policy-options policy-statement ted2nlri term 1 from protocol bgp-ls-epe
user@R0#set policy-options policy-statement ted2nlri term 1 then accept
```

6. Configure BGP to enable BGP-LS route advertisement to the connected peers and define the EPE links. Since express segment is an internal TE link, this configuration creates an external TE link.

```
[edit]
user@R0#set protocols bgp group ebgp1 type external
user@R0#set protocols bgp group ebgp1 family inet unicast
user@R0#set protocols bgp group ebgp1 family traffic-engineering unicast
user@R0#set protocols bgp group ebgp1 export nlri2bgp_epe
user@R0#set protocols bgp group ebgp1 neighbor 192.168.1.2 peer-as 65200
user@R0#set protocols bgp group ebgp1 neighbor 192.168.1.2 egress-te-adj-segment
epe_adj1_toR1 label 7101
user@R0#set protocols bgp group ebgp1 neighbor 192.168.1.2 egress-te-adj-segment
epe_adj1_toR1 next-hop 192.168.1.2
user@R0#set protocols bgp group ebgp1 neighbor 192.168.1.2 egress-te-adj-segment
epe_adj1_toR1 te-link-attribute te-metric 20
user@R0#set protocols bgp group ebgp1 neighbor 192.168.1.2 egress-te-adj-segment
epe_adj1_toR1 te-link-attribute igp-metric 10
user@R0#set protocols bgp group ebgp1 neighbor 192.168.1.2 egress-te-adj-segment
epe_adj1_toR1 te-link-attribute admin-group red
user@R0#set protocols bgp group ebgp1 neighbor 192.168.1.2 egress-te-adj-segment
epe_adj1_toR1 te-link-attribute admin-group brown
user@R0#set protocols bgp group ebgp1 neighbor 192.168.2.2 peer-as 200
user@R0#set protocols bgp group ebgp1 neighbor 192.168.2.2 egress-te-adj-segment
epe_adj1_toR4 label 7104
user@R0#set protocols bgp group ebgp1 neighbor 192.168.2.2 egress-te-adj-segment
```

```
epe_adj1_toR4 next-hop 192.168.2.2
user@R0#set protocols bgp group ebgp1 neighbor 192.168.2.2 egress-te-adj-segment
epe_adj1_toR4 te-link-attribute te-metric 20
user@R0#set protocols bgp group ebgp1 neighbor 192.168.2.2 egress-te-adj-segment
epe_adj1_toR4 te-link-attribute igp-metric 10
user@R0#set protocols bgp group ebgp1 neighbor 192.168.2.2 egress-te-adj-segment
epe_adj1_toR4 te-link-attribute admin-group red
user@R0#set protocols bgp group ebgp1 neighbor 192.168.2.2 egress-te-adj-segment
epe_adj1_toR4 te-link-attribute admin-group brown
```

7. Enable import and export of traffic engineering database parameters using policies.

```
[edit]
user@R0#set protocols mpls traffic-engineering database import l3-unicast-topology bgp-link-
state
user@R0#set protocols mpls traffic-engineering database import policy ted2nlri
user@R0#set protocols mpls traffic-engineering database export policy nlri2ted_bgp
user@R0#set protocols mpls traffic-engineering database export l3-unicast-topology
```

8. Configure MPLS administrative group policies for LSP path computation.

```
[edit]
user@R0#set protocols mpls admin-groups red 0
user@R0#set protocols mpls admin-groups blue 1
user@R0#set protocols mpls admin-groups brown 5
```

9. Configure the MPLS label range to assign static labels for the EPE links.

```
[edit]
user@R0#set protocols mpls label-range static-label-range 7000 70000
```

10. Configure MPLS on the interfaces.

```
[edit]
user@R0#set protocols mpls interface all
```

11. Configure SR-TE policies on the ingress router to enable end-to-end SR-TE policy.

```
[edit]
user@R0#set protocols source-packet-routing compute-profile compute1 no-label-stack-
compression
user@R0#set protocols source-packet-routing compute-profile ecompute1 admin-group include-
any red
user@R0#set protocols source-packet-routing compute-profile ecompute1 admin-group include-
any brown
user@R0#set protocols source-packet-routing compute-profile ecompute1 no-label-stack-
compression
user@R0#set protocols source-packet-routing compute-profile ecompute2 admin-group include-
any red
user@R0#set protocols source-packet-routing compute-profile ecompute2 admin-group include-
any blue
user@R0#set protocols source-packet-routing compute-profile ecompute2 no-label-stack-
compression
user@R0#set protocols source-packet-routing source-routing-path computelsp1 to 10.7.7.7
user@R0#set protocols source-packet-routing source-routing-path computelsp1 install
10.7.7.71
user@R0#set protocols source-packet-routing source-routing-path computelsp1 primary p1
compute compute1
user@R0#set protocols source-packet-routing source-routing-path ecomputelsp1 to 10.7.7.7
user@R0#set protocols source-packet-routing source-routing-path ecomputelsp1 color 7000
user@R0#set protocols source-packet-routing source-routing-path ecomputelsp1 primary p1
compute ecompute1
user@R0#set protocols source-packet-routing source-routing-path ecomputelsp2 to 10.7.7.7
user@R0#set protocols source-packet-routing source-routing-path ecomputelsp2 color 7001
user@R0#set protocols source-packet-routing source-routing-path ecomputelsp2 primary p1
compute ecompute2
```

Results

From configuration mode, confirm your configuration by entering the `show chassis`, `show interfaces`, `show policy-optionsshow routing-options`, and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
chassis {
    network-services enhanced-ip;
}
```

```
interfaces {
    ge-0/0/0 {
        description To_R1_1;
        unit 0 {
            family inet {
                address 192.168.1.1/24;
            }
            family iso;
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/2 {
        description To_R4_1;
        unit 0 {
            family inet {
                address 192.168.2.1/24;
            }
            family iso;
            family mpls {
                maximum-labels 8;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.100.100.100/32;
            }
            family iso {
                address 49.0001.000a.0a0a.0a00;
            }
        }
    }
}
policy-options {
    policy-statement nlri2bgp_epe {
        term 1 {
            from {
                family traffic-engineering;
                protocol bgp-ls-epe;
            }
```

```
            then {
                next-hop self;
                accept;
            }
        }
    }
    policy-statement nlri2ted_bgp {
        term 1 {
            from protocol bgp;
            then accept;
        }
    }
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
    }
    policy-statement ted2nlri {
        term 1 {
            from protocol bgp-ls-epe;
            then accept;
        }
    }
}
routing-options {
    static {
        route 10.7.7.71/32 {
            next-hop 10.7.7.7;
            resolve;
        }
    }
    router-id 10.100.100.100;
    autonomous-system 65100;
    forwarding-table {
        ecmp-fast-reroute;
    }
}
protocols {
    bgp {
        group ebgp1 {
            type external;
            family inet {
                unicast;
```

```
            }
            family traffic-engineering {
                unicast;
            }
            export nlri2bgp_epe;
            neighbor 192.168.1.2 {
                peer-as 65200;
                egress-te-adj-segment epe_adj1_toR1 {
                    label 7101;
                    next-hop 192.168.1.2;
                    te-link-attribute {
                        te-metric 20;
                        igp-metric 10;
                        admin-group [ red brown ];
                    }
                }
            }
            neighbor 192.168.2.2 {
                peer-as 65200;
                egress-te-adj-segment epe_adj1_toR4 {
                    label 7104;
                    next-hop 192.168.2.2;
                    te-link-attribute {
                        te-metric 20;
                        igp-metric 10;
                        admin-group [ red brown ];
                    }
                }
            }
        }
    }
    mpls {
        traffic-engineering {
            database {
                import {
                    l3-unicast-topology {
                        bgp-link-state;
                    }
                    policy ted2nlri;
                }
                export {
                    policy nlri2ted_bgp;
                    l3-unicast-topology;
```

```
                }
            }
        }
        admin-groups {
            red 0;
            blue 1;
            brown 5;
        }
        label-range {
            static-label-range 7000 70000;
        }
        interface all;
    }
    source-packet-routing {
        compute-profile compute1 {
            no-label-stack-compression;
        }
        compute-profile ecompute1 {
            admin-group include-any [ red brown ];
            no-label-stack-compression;
        }
        compute-profile ecompute2 {
            admin-group include-any [ red blue ];
            no-label-stack-compression;
        }
        source-routing-path computelsp1 {
            to 10.7.7.7;
            install 10.7.7.71;
            primary {
                p1 {
                    compute {
                        compute1;
                    }
                }
            }
        }
        source-routing-path ecomputelsp1 {
            to 10.7.7.7;
            color 7000;
            primary {
                p1 {
                    compute {
                        ecompute1;
```

```
                    }
                }
            }
        }
        source-routing-path ecomputelsp2 {
            to 10.7.7.7;
            color 7001;
            primary {
                p1 {
                    compute {
                        ecompute2;
                    }
                }
            }
        }
    }
}
```

Configure R1 (BN1 router)

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure device R1:

1. Configure the network services mode as Enhanced IP. Enhanced IP sets the router's network services to enhanced Internet Protocol and uses enhanced mode capabilities.

   ```
   [edit]
   user@R1#set chassis network-services enhanced-ip
   ```

   After you configure the `enhanced-ip` statement and commit the configuration, the following warning message appears prompting you to reboot the router:

   ```
   'chassis'
   WARNING: Chassis configuration for network services has been changed. A system reboot is
   mandatory. Please reboot the system NOW. Continuing without a reboot might result in
   ```

```
unexpected system behavior.
commit complete
```

The reboot brings up the FPCs on the router.

2. Configure the interfaces to enable IP, MPLS, and ISO transport.

```
[edit]
user@R1#set interfaces ge-0/0/0 description To_R0_1
user@R1#set interfaces ge-0/0/0 unit 0 family inet address 192.168.1.2/24
user@R1#set interfaces ge-0/0/0 unit 0 family iso
user@R1#set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 8
user@R1#set interfaces ge-0/0/2 description To_R2
user@R1#set interfaces ge-0/0/2 unit 0 family inet address 192.168.3.1/24
user@R1#set interfaces ge-0/0/2 unit 0 family iso
user@R1#set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
user@R1#set interfaces ge-0/0/3 description to-R4
user@R1#set interfaces ge-0/0/3 unit 0 family inet address 192.168.4.1/24
user@R1#set interfaces ge-0/0/3 unit 0 family iso
user@R1#set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
user@R1#set interfaces ge-0/0/4 description to-R5
user@R1#set interfaces ge-0/0/4 unit 0 family inet address 192.168.5.1/24
user@R1#set interfaces ge-0/0/4 unit 0 family iso
user@R1#set interfaces ge-0/0/4 unit 0 family mpls maximum-labels 8
```

3. Configure the loopback interface to enable tunnel endpoints and service endpoints.

```
[edit]
user@R1#set interfaces lo0 unit 0 family inet address 10.1.1.1/32
user@R1#set interfaces lo0 unit 0 family iso address 49.0001.0001.0101.0100
```

4. Configure routing options to identify the router in the domain.

```
[edit]
user@R1#set routing-options router-id 10.1.1.1
user@R1#set routing-options autonomous-system 65200
```

5. Define import and export policies. For example, configure policies that export EPE TE links from the local TE database to lsdist.0 and policies to import from lsdist.0 into the local TE database. You can configure policies to advertise the BGP routes to a peer.

```
[edit]
user@R1#set policy-options policy-statement expresspol1 from route-filter 10.6.6.6/32 exact
install-nexthop lsp lsp1to6_a
user@R1#set policy-options policy-statement expresspol1 then accept
user@R1#set policy-options policy-statement expresspol2 from route-filter 10.3.3.3/32 exact
install-nexthop lsp lsp1to3_a
user@R1#set policy-options policy-statement expresspol2 then accept
user@R1#set policy-options policy-statement nlri2bgp_epe term 1 from family traffic-
engineering
user@R1#set policy-options policy-statement nlri2bgp_epe term 1 from protocol bgp-ls-epe
user@R1#set policy-options policy-statement nlri2bgp_epe term 1 then next-hop self
user@R1#set policy-options policy-statement nlri2bgp_epe term 1 then accept
user@R1#set policy-options policy-statement nlri2bgp_stat term 1 from family traffic-
engineering
user@R1#set policy-options policy-statement nlri2bgp_stat term 1 from protocol express-
segments
user@R1#set policy-options policy-statement nlri2bgp_stat term 1 then accept
user@R1#set policy-options policy-statement pplb then load-balance per-packet
user@R1#set policy-options policy-statement ted2nlri_epe_stat term 1 from family traffic-
engineering
user@R1#set policy-options policy-statement ted2nlri_epe_stat term 1 from protocol express-
segments
user@R1#set policy-options policy-statement ted2nlri_epe_stat term 1 then accept
user@R1#set policy-options policy-statement ted2nlri_epe_stat term 2 from family traffic-
engineering
user@R1#set policy-options policy-statement ted2nlri_epe_stat term 2 from protocol bgp-ls-
epe
user@R1#set policy-options policy-statement ted2nlri_epe_stat term 2 then accept
user@R1#set policy-options policy-statement ted2nlri_epe_stat term 3 from protocol isis
user@R1#set policy-options policy-statement ted2nlri_epe_stat term 3 then reject
```

6. Configure BGP to enable BGP-LS route advertisement to the connected peers and define the EPE links. Since express segment is an internal TE link, this configuration creates an external TE link.

```
[edit]
user@R1#set protocols bgp group ebgp1 type external
user@R1#set protocols bgp group ebgp1 family inet-vpn unicast
```

```
user@R1#set protocols bgp group ebgp1 family traffic-engineering unicast
user@R1#set protocols bgp group ebgp1 export nlri2bgp_stat
user@R1#set protocols bgp group ebgp1 neighbor 192.168.1.1 peer-as 65100
user@R1#set protocols bgp group ebgp1 neighbor 192.168.1.1 egress-te-adj-segment
epe_adj1_toR0 label 8110
user@R1#set protocols bgp group ebgp1 neighbor 192.168.1.1 egress-te-adj-segment
epe_adj1_toR0 next-hop 192.168.1.1
user@R1#set protocols bgp group ebgp1 neighbor 192.168.1.1 egress-te-adj-segment
epe_adj1_toR0 te-link-attribute te-metric 20
user@R1#set protocols bgp group ebgp1 neighbor 192.168.1.1 egress-te-adj-segment
epe_adj1_toR0 te-link-attribute igp-metric 10
user@R1#set protocols bgp group ebgp1 neighbor 192.168.1.1 egress-te-adj-segment
epe_adj1_toR0 te-link-attribute admin-group red
user@R1#set protocols bgp group ebgp1 neighbor 192.168.1.1 egress-te-adj-segment
epe_adj1_toR0 te-link-attribute admin-group brown
user@R1#set protocols bgp group ibgp1 type internal
user@R1#set protocols bgp group ibgp1 local-address 10.1.1.1
user@R1#set protocols bgp group ibgp1 family traffic-engineering unicast
user@R1#set protocols bgp group ibgp1 export nlri2bgp_epe
user@R1#set protocols bgp group ibgp1 neighbor 10.2.2.2
user@R1#set protocols bgp group ibgp1 neighbor 10.5.5.5
```

7. Configure the express segment set and express segment templates. What the express segment template does is it manually assigns or overrides inherited attributes to the express segments regardless of what the underlay attributes are. The express segment name `r1-exp-set1` is prefixed to the underlay end point for automatic naming.

```
[edit]
user@R1#set protocols express-segments segment-template template1 admin-group red
user@R1#set protocols express-segments segment-template template1 metric te 200
user@R1#set protocols express-segments segment-template template1 metric igp 100
user@R1#set protocols express-segments segment-set r1-exp-set1 membership-policy expresspol1
user@R1#set protocols express-segments segment-set r1-exp-set1 template template1
user@R1#set protocols express-segments segment-set r1-exp-set2 membership-policy expresspol2
user@R1#set protocols express-segments traffic-engineering
```

8. Configure IS-IS protocol on the interfaces and apply MPLS administrative groups to those interfaces.

```
[edit]
user@R1#set protocols isis interface ge-0/0/2.0
```

```
user@R1#set protocols isis interface ge-0/0/3.0
user@R1#set protocols isis interface ge-0/0/4.0
user@R1#set protocols isis interface lo0.0 passive
user@R1#set protocols isis level 1 disable
user@R1#set protocols isis level 2 wide-metrics-only
user@R1#set protocols mpls interface ge-0/0/3.0 admin-group red
user@R1#set protocols mpls interface ge-0/0/2.0 admin-group brown
user@R1#user@R1#set protocols mpls interface ge-0/0/4.0 admin-group blue
user@R1#set protocols mpls interface all
```

9. Enable link protection on all the RSVP interfaces. Using link protection, you can configure a network to reroute traffic quickly around broken links.

```
[edit]
user@R1#set protocols rsvp interface all link-protection
```

10. Enable import and export of traffic engineering database parameters using the policies.

```
[edit]
user@R1#set protocols mpls traffic-engineering database import l3-unicast-topology bgp-link-
state
user@R1#set protocols mpls traffic-engineering database import policy ted2nlri_epe_stat
user@R1#set protocols mpls traffic-engineering database export l3-unicast-topology
```

11. Configure MPLS administrative group policies for LSP path computation.

```
[edit]
user@R1#set protocols mpls admin-groups red 0
user@R1#set protocols mpls admin-groups blue 1
user@R1#set protocols mpls admin-groups brown 5
```

12. Configure MPLS with a label-switched path (LSP) and include administrative groups.

```
[edit]
user@R1#set protocols mpls label-switched-path lsp1to6_a to 10.6.6.6
user@R1#set protocols mpls label-switched-path lsp1to6_a admin-group include-any brown
user@R1#set protocols mpls label-switched-path lsp1to6_a admin-group include-any red
user@R1#set protocols mpls label-switched-path lsp1to6_b to 10.6.6.6
user@R1#set protocols mpls label-switched-path lsp1to6_b admin-group include-any brown
```

```
user@R1#set protocols mpls label-switched-path lsp1to6_b admin-group include-any blue
user@R1#set protocols mpls label-switched-path lsp1to6_c to 10.6.6.6
user@R1#set protocols mpls label-switched-path lsp1to6_c admin-group include-any blue
user@R1#set protocols mpls label-switched-path lsp1to3_a to 10.3.3.3
user@R1#set protocols mpls label-switched-path lsp1to3_a admin-group include-any brown
user@R1#set protocols mpls label-switched-path lsp1to3_a admin-group include-any red
user@R1#set protocols mpls label-switched-path lsp1to3_b to 10.3.3.3
user@R1#set protocols mpls label-switched-path lsp1to3_b admin-group include-any blue
user@R1#set protocols mpls label-range static-label-range 7000 70000
```

Results

From configuration mode, confirm your configuration by entering the `show chassis`, `show interfaces`, `show policy-options`, `show routing-options`, and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
chassis {
    network-services enhanced-ip;
}
interfaces {
    ge-0/0/0 {
        description To_R0_1;
        unit 0 {
            family inet {
                address 192.168.1.2/24;
            }
            family iso;
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/2 {
        description To_R2;
        unit 0 {
            family inet {
                address 192.168.3.1/24;
            }
            family iso;
            family mpls {
                maximum-labels 8;
```

```
            }
        }
    }
    ge-0/0/3 {
        description to-R4;
        unit 0 {
            family inet {
                address 192.168.4.1/24;
            }
            family iso;
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/4 {
        description to-R5;
        unit 0 {
            family inet {
                address 192.168.5.1/24;
            }
            family iso;
            family mpls {
                maximum-labels 8;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.1.1.1/32;
            }
            family iso {
                address 49.0001.0001.0101.0100;
            }
        }
    }
}
policy-options {
    policy-statement expresspol1 {
        from {
            route-filter 10.6.6.6/32 exact {
                install-nexthop lsp lsp1to6_a;
```

```
            }
        }
        then accept;
    }
    policy-statement expresspol2 {
        from {
            route-filter 10.3.3.3/32 exact {
                install-nexthop lsp lsp1to3_a;
            }
        }
        then accept;
    }
    policy-statement nlri2bgp_epe {
        term 1 {
            from {
                family traffic-engineering;
                protocol bgp-ls-epe;
            }
            then {
                next-hop self;
                accept;
            }
        }
    }
    policy-statement nlri2bgp_stat {
        term 1 {
            from {
                family traffic-engineering;
                protocol express-segments;
            }
            then accept;
        }
    }
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
    }
    policy-statement ted2nlri_epe_stat {
        term 1 {
            from {
                family traffic-engineering;
                protocol express-segments;
```

```
            }
            then accept;
        }
        term 2 {
            from {
                family traffic-engineering;
                protocol bgp-ls-epe;
            }
            then accept;
        }
        term 3 {
            from protocol isis;
            then reject;
        }
    }
}
routing-options {
    router-id 10.1.1.1;
    autonomous-system 65200;
}
protocols {
    bgp {
        group ebgp1 {
            type external;
            family inet-vpn {
                unicast;
            }
            family traffic-engineering {
                unicast;
            }
            export nlri2bgp_stat;
            neighbor 192.168.1.1 {
                peer-as 65100;
                egress-te-adj-segment epe_adj1_toR0 {
                    label 8110;
                    next-hop 192.168.1.1;
                    te-link-attribute {
                        te-metric 20;
                        igp-metric 10;
                        admin-group [ red brown ];
                    }
                }
            }
```

```
        }
        group ibgp1 {
            type internal;
            local-address 10.1.1.1;
            family traffic-engineering {
                unicast;
            }
            export nlri2bgp_epe;
            neighbor 10.2.2.2;
            neighbor 10.5.5.5;
        }
    }
    express-segments {
        segment-template template1 {
            admin-group red;
            metric {
                te 200;
                igp 100;
            }
        }
        segment-set r1-exp-set1 {
            membership-policy expresspol1;
            template {
                template1;
            }
        }
        segment-set r1-exp-set2 {
            membership-policy expresspol2;
        }
        traffic-engineering;
    }
    isis {
        interface ge-0/0/2.0;
        interface ge-0/0/3.0;
        interface ge-0/0/4.0;
        interface lo0.0 {
            passive;
        }
        level 1 disable;
        level 2 wide-metrics-only;
    }
    mpls {
        traffic-engineering {
```

```
            database {
                import {
                    l3-unicast-topology {
                        bgp-link-state;
                    }
                    policy ted2nlri_epe_stat;
                }
                export {
                    l3-unicast-topology;
                }
            }
        }
        admin-groups {
            red 0;
            blue 1;
            brown 5;
        }
        label-switched-path lsp1to6_a {
            to 10.6.6.6;
            admin-group include-any [ brown red ];
        }
        label-switched-path lsp1to6_b {
            to 10.6.6.6;
            admin-group include-any [ brown blue ];
        }
        label-switched-path lsp1to3_a {
            to 10.3.3.3;
            admin-group include-any [ brown red ];
        }
        label-switched-path lsp1to3_b {
            to 10.3.3.3;
            admin-group include-any [ blue ];
        }
        label-range {
            static-label-range 7000 70000;
        }
        interface ge-0/0/3.0 {
            admin-group red;
        }
        interface ge-0/0/2.0 {
            admin-group brown;
        }
        interface ge-0/0/4.0 {
```

```
            admin-group blue;
        }
        interface all;
    }
    rsvp {
        interface all {
            link-protection;
        }
    }
}
```

Configure R4 (BN2 router)

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure device R4:

1. Configure the network services mode as Enhanced IP. Enhanced IP sets the router's network services to enhanced Internet Protocol and uses enhanced mode capabilities.

   ```
   [edit]
   user@R4#set chassis network-services enhanced-ip
   ```

   After you configure the `enhanced-ip` statement and commit the configuration, the following warning message appears prompting you to reboot the router:

   ```
   'chassis'
   WARNING: Chassis configuration for network services has been changed. A system reboot is
   mandatory. Please reboot the system NOW. Continuing without a reboot might result in
   unexpected system behavior.
   commit complete
   ```

   The reboot brings up the FPCs on the router.

2. Configure the interfaces to enable IP, MPLS, and ISO transport.

```
user@R4#set interfaces ge-0/0/0 description To_R0
user@R4#set interfaces ge-0/0/0 unit 0 family inet address 192.168.2.2/24
user@R4#set interfaces ge-0/0/0 unit 0 family iso
user@R4#set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 8
user@R4#set interfaces ge-0/0/2 description To_R1
user@R4#set interfaces ge-0/0/2 unit 0 family inet address 192.168.4.2/24
user@R4#set interfaces ge-0/0/2 unit 0 family iso
user@R4#set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
user@R4#set interfaces ge-0/0/3 description To_R2
user@R4#set interfaces ge-0/0/3 unit 0 family inet address 192.168.7.2/24
user@R4#set interfaces ge-0/0/3 unit 0 family iso
user@R4#set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
user@R4#set interfaces ge-0/0/4 description To_R5
user@R4#set interfaces ge-0/0/4 unit 0 family inet address 192.168.13.1/24
user@R4#set interfaces ge-0/0/4 unit 0 family iso
user@R4#set interfaces ge-0/0/4 unit 0 family mpls maximum-labels 8
```

3. Configure the loopback interface to enable tunnel endpoints and service endpoints.

```
[edit]
user@R4#set interfaces lo0 unit 0 family inet address 10.4.4.4/32
user@R4#set interfaces lo0 unit 0 family iso address 49.0001.0004.0404.0400
```

4. Configure routing options to identify the router in the domain.

```
[edit]
user@R4#set routing-options router-id 10.4.4.4
user@R4#set routing-options autonomous-system 65200
```

5. Define import and export policies. For example, configure policies that export EPE TE links from the local TE database to **lsdist.0** and policies to import from **lsdist.0** into the local TE database. You can configure policies to advertise the BGP routes to a peer.

```
[edit]
user@R4#set policy-options policy-statement expresspol1 from route-filter 10.6.6.6/32 exact
install-nexthop lsp lsp4to6_a
user@R4#set policy-options policy-statement expresspol1 then accept
```

```
user@R4#set policy-options policy-statement expresspol2 from route-filter 10.3.3.3/32 exact
install-nexthop lsp lsp4to3_a
user@R4#set policy-options policy-statement expresspol2 then accept
user@R4#set policy-options policy-statement nlri2bgp_epe term 1 from family traffic-
engineering
user@R4#set policy-options policy-statement nlri2bgp_epe term 1 from protocol bgp-ls-epe
user@R4#set policy-options policy-statement nlri2bgp_epe term 1 then next-hop self
user@R4#set policy-options policy-statement nlri2bgp_epe term 1 then accept
user@R4#set policy-options policy-statement nlri2bgp_stat term 1 from family traffic-
engineering
user@R4#set policy-options policy-statement nlri2bgp_stat term 1 from protocol express-
segments
user@R4#set policy-options policy-statement nlri2bgp_stat term 1 then accept
user@R4#set policy-options policy-statement pplb then load-balance per-packet
user@R4#set policy-options policy-statement ted2nlri_epe_stat term 1 from family traffic-
engineering
user@R4#set policy-options policy-statement ted2nlri_epe_stat term 1 from protocol express-
segments
user@R4#set policy-options policy-statement ted2nlri_epe_stat term 1 then accept
user@R4#set policy-options policy-statement ted2nlri_epe_stat term 2 from family traffic-
engineering
user@R4#set policy-options policy-statement ted2nlri_epe_stat term 2 from protocol bgp-ls-
epe
user@R4#set policy-options policy-statement ted2nlri_epe_stat term 2 then accept
user@R4#set policy-options policy-statement ted2nlri_epe_stat term 3 from protocol isis
user@R4#set policy-options policy-statement ted2nlri_epe_stat term 3 then reject
```

6. Configure the express segment set and express segment templates. What the express segment template does is it manually assigns or overrides inherited attributes to the express segments regardless of what the underlay attributes are. The express segment name `r4-exp-set1` is prefixed to the underlay end point for automatic naming.

```
[edit]
user@R4#set protocols express-segments segment-set r4-exp-set1 membership-policy expresspol1
user@R4#set protocols express-segments segment-set r4-exp-set2 membership-policy expresspol2
user@R4#set protocols express-segments traffic-engineering
```

7. Configure IS-IS and MPLS protocol on the interfaces.

```
[edit]
user@R4#set protocols isis interface ge-0/0/0.0
```

```
user@R4#set protocols isis interface ge-0/0/2.0
user@R4#set protocols isis interface ge-0/0/3.0
user@R4#set protocols isis interface ge-0/0/4.0
user@R4#set protocols isis interface lo0.0 passive
user@R4#set protocols isis level 1 disable
user@R4#set protocols isis level 2 wide-metrics-only
user@R4#set protocols mpls interface ge-0/0/2.0 admin-group red
user@R4#set protocols mpls interface ge-0/0/4.0 admin-group brown
user@R4#set protocols mpls interface all
```

8. Enable import and export of traffic engineering database parameters using policies.

```
[edit]
user@R4#set protocols mpls traffic-engineering database import l3-unicast-topology bgp-link-
state
user@R4#set protocols mpls traffic-engineering database import policy ted2nlri_epe_stat
user@R4#set protocols mpls traffic-engineering database export l3-unicast-topology
```

9. Configure MPLS administrative group policies for LSP path computation.

```
[edit]
user@R4#set protocols mpls admin-groups red 0
user@R4#set protocols mpls admin-groups blue 1
user@R4#set protocols mpls admin-groups brown 5
```

10. Configure MPLS with a label-switched path (LSP) and include administrative groups.

```
[edit]
user@R4#set protocols mpls label-switched-path lsp4to6_a to 10.6.6.6
user@R4#set protocols mpls label-switched-path lsp4to6_a admin-group include-any brown
user@R4#set protocols mpls label-switched-path lsp4to6_a admin-group include-any red
user@R4#set protocols mpls label-switched-path lsp4to6_b to 10.6.6.6
user@R4#set protocols mpls label-switched-path lsp4to6_b admin-group include-any blue
user@R4#set protocols mpls label-switched-path lsp4to3_a to 10.3.3.3
user@R4#set protocols mpls label-switched-path lsp4to3_a admin-group include-any brown
user@R4#set protocols mpls label-switched-path lsp4to3_a admin-group include-any red
user@R4#set protocols mpls label-switched-path lsp4to3_b to 10.3.3.3
user@R4#set protocols mpls label-switched-path lsp4to3_b admin-group include-any brown
```

```
user@R4#set protocols mpls label-switched-path lsp4to3_c to 10.3.3.3
user@R4#set protocols mpls label-switched-path lsp4to3_c admin-group include-any brown
```

11. Configure the MPLS label range to assign static labels for the EPE links.

```
[edit]
user@R4#set protocols mpls label-range static-label-range 7000 70000
```

12. Enable link protection on all the RSVP interfaces. Using link protection, you can configure a network to reroute traffic quickly around broken links.

```
[edit]
user@R4#set protocols rsvp interface all link-protection
```

Results

From configuration mode, confirm your configuration by entering the `show chassis`, `show interfaces`, `show policy-options`, `show routing-options`, and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
chassis {
    network-services enhanced-ip;
}
interfaces {
    ge-0/0/0 {
        description To_R0;
        unit 0 {
            family inet {
                address 192.168.2.2/24;
            }
            family iso;
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/2 {
        description To_R1;
        unit 0 {
```

```
            family inet {
                address 192.168.4.2/24;
            }
            family iso;
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/3 {
        description To_R2;
        unit 0 {
            family inet {
                address 192.168.7.2/24;
            }
            family iso;
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/4 {
        description To_R5;
        unit 0 {
            family inet {
                address 192.168.13.1/24;
            }
            family iso;
            family mpls {
                maximum-labels 8;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.4.4.4/32;
            }
            family iso {
                address 49.0001.0004.0404.0400;
            }
        }
    }
```

```
}
policy-options {
    policy-statement expresspol1 {
        from {
            route-filter 10.6.6.6/32 exact {
                install-nexthop lsp lsp4to6_a;
            }
        }
        then accept;
    }
    policy-statement expresspol2 {
        from {
            route-filter 10.3.3.3/32 exact {
                install-nexthop lsp lsp4to3_a;
            }
        }
        then accept;
    }
    policy-statement nlri2bgp_epe {
        term 1 {
            from {
                family traffic-engineering;
                protocol bgp-ls-epe;
            }
            then {
                next-hop self;
                accept;
            }
        }
    }
    policy-statement nlri2bgp_stat {
        term 1 {
            from {
                family traffic-engineering;
                protocol express-segments;
            }
            then accept;
        }
    }
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
```

```
    }
    policy-statement ted2nlri_epe_stat {
        term 1 {
            from {
                family traffic-engineering;
                protocol express-segments;
            }
            then accept;
        }
        term 2 {
            from {
                family traffic-engineering;
                protocol bgp-ls-epe;
            }
            then accept;
        }
        term 3 {
            from protocol isis;
            then reject;
        }
    }
}
routing-options {
    router-id 10.4.4.4;
    autonomous-system 65200;
}
protocols {
    bgp {
        group ibgp1 {
            type internal;
            local-address 10.4.4.4;
            family traffic-engineering {
                unicast;
            }
            export nlri2bgp_epe;
            neighbor 10.2.2.2;
            neighbor 10.5.5.5;
        }
        group ebgp1 {
            type external;
            family inet-vpn {
                unicast;
            }
```

```
            family traffic-engineering {
                unicast;
            }
            export nlri2bgp_stat;
            neighbor 192.168.2.1 {
                peer-as 65100;
                egress-te-adj-segment epe_adj1_toR0 {
                    label 8140;
                    next-hop 192.168.2.1;
                    te-link-attribute {
                        te-metric 20;
                        igp-metric 10;
                        admin-group [ red brown ];
                    }
                }
            }
        }
    }
    express-segments {
        segment-set r4-exp-set1 {
            membership-policy expresspol1;
        }
        segment-set r4-exp-set2 {
            membership-policy expresspol2;
        }
        traffic-engineering;
    }
    isis {
        interface ge-0/0/0.0;
        interface ge-0/0/2.0;
        interface ge-0/0/3.0;
        interface ge-0/0/4.0;
        interface lo0.0 {
            passive;
        }
        level 1 disable;
        level 2 wide-metrics-only;
    }
    mpls {
        traffic-engineering {
            database {
                import {
                    l3-unicast-topology {
```

```
                bgp-link-state;
            }
            policy ted2nlri_epe_stat;
        }
        export {
            l3-unicast-topology;
        }
    }
}
admin-groups {
    red 0;
    blue 1;
    brown 5;
}
label-switched-path lsp4to6_a {
    to 10.6.6.6;
    admin-group include-any [ brown red ];
}
label-switched-path lsp4to6_b {
    to 10.6.6.6;
    admin-group include-any [ blue ];
}
label-switched-path lsp4to3_a {
    to 10.3.3.3;
    admin-group include-any [ brown red ];
}
label-switched-path lsp4to3_b {
    to 10.3.3.3;
    admin-group include-any [ brown ];
}
label-switched-path lsp4to3_c {
    to 10.3.3.3;
    admin-group include-any [ brown ];
}
label-range {
    static-label-range 7000 70000;
}
interface ge-0/0/2.0 {
    admin-group red;
}
interface ge-0/0/4.0 {
    admin-group brown;
}
```

```
        interface all;
    }
    rsvp {
        interface all {
            link-protection;
        }
    }
}
```

Configure R2 (Intermediate router)

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure device R2:

1. Configure the network services mode as Enhanced IP. Enhanced IP sets the router's network services to enhanced Internet Protocol and uses enhanced mode capabilities.

```
[edit]
user@R2#set chassis network-services enhanced-ip
```

After you configure the `enhanced-ip` statement and commit the configuration, the following warning message appears prompting you to reboot the router:

```
'chassis'
WARNING: Chassis configuration for network services has been changed. A system reboot is
mandatory. Please reboot the system NOW. Continuing without a reboot might result in
unexpected system behavior.
commit complete
```

The reboot brings up the FPCs on the router.

2. Configure the interfaces to enable IP, MPLS, and ISO transport.

```
user@R2#set interfaces ge-0/0/0 description To_R1
user@R2#set interfaces ge-0/0/0 unit 0 family inet address 192.168.3.2/24
user@R2#set interfaces ge-0/0/0 unit 0 family iso
```

```
user@R2#set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 8
user@R2#set interfaces ge-0/0/1 description To_R3
user@R2#set interfaces ge-0/0/1 unit 0 family inet address 192.168.6.1/24
user@R2#set interfaces ge-0/0/1 unit 0 family iso
user@R2#set interfaces ge-0/0/1 unit 0 family mpls maximum-labels 8
user@R2#set interfaces ge-0/0/2 description To_R4
user@R2#set interfaces ge-0/0/2 unit 0 family inet address 192.168.7.1/24
user@R2#set interfaces ge-0/0/2 unit 0 family iso
user@R2#set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
user@R2#set interfaces ge-0/0/3 description To_R5
user@R2#set interfaces ge-0/0/3 unit 0 family inet address 192.168.8.1/24
user@R2#set interfaces ge-0/0/3 unit 0 family iso
user@R2#set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
user@R2#set interfaces ge-0/0/4 description To_R6
user@R2#set interfaces ge-0/0/4 unit 0 family inet address 192.168.9.1/24
user@R2#set interfaces ge-0/0/4 unit 0 family iso
user@R2#set interfaces ge-0/0/4 unit 0 family mpls maximum-labels 8
```

3. Configure the loopback interface to enable tunnel endpoints and service endpoints.

```
[edit]
user@R2#set interfaces lo0 unit 0 family inet address 10.2.2.2/32
user@R2#set interfaces lo0 unit 0 family iso address 49.0001.0002.0202.0200
```

4. Configure routing options to identify the router in the domain.

```
[edit]
user@R2#set routing-options router-id 10.2.2.2
user@R2#set routing-options autonomous-system 65200
```

5. Define import and export policies. For example, configure policies that export EPE TE links from the local TE database to **lsdist.0** and policies to import from **lsdist.0** into the local TE database. You can configure policies to advertise the BGP routes to a peer.

```
[edit]
user@R2#set policy-options policy-statement bgplsepe_rt_2_ted term 1 from protocol bgp
user@R2#set policy-options policy-statement bgplsepe_rt_2_ted term 1 then accept
user@R2#set policy-options policy-statement nlri2bgp term 1 from family traffic-engineering
user@R2#set policy-options policy-statement nlri2bgp term 1 then next-hop self
user@R2#set policy-options policy-statement nlri2bgp term 1 then accept
```

```
user@R2#set policy-options policy-statement nlri2bgp_igp term 1 from family traffic-
engineering
user@R2#set policy-options policy-statement nlri2bgp_igp term 1 from protocol isis
user@R2#set policy-options policy-statement nlri2bgp_igp term 1 then accept
user@R2#set policy-options policy-statement nlri2ted_igp term 1 from traffic-engineering
protocol isis-level-2
user@R2#set policy-options policy-statement nlri2ted_igp term 1 then accept
user@R2#set policy-options policy-statement pplb then load-balance per-packet
user@R2#set policy-options policy-statement ted2nlri term 1 from protocol bgp-ls-epe
user@R2#set policy-options policy-statement ted2nlri term 1 then accept
user@R2#set policy-options policy-statement ted2nlri_1 term 1 from traffic-engineering
user@R2#set policy-options policy-statement ted2nlri_1 term 1 then accept
user@R2#set policy-options policy-statement ted2nlri_igp term 1 from family traffic-
engineering
user@R2#set policy-options policy-statement ted2nlri_igp term 1 from protocol isis
user@R2#set policy-options policy-statement ted2nlri_igp term 1 then accept
```

6. Configure BGP to enable BGP-LS route advertisement to the connected peers.

```
[edit]
user@R2#set protocols bgp group RR1 type internal
user@R2#set protocols bgp group RR1 local-address 10.2.2.2
user@R2#set protocols bgp group RR1 family traffic-engineering unicast
user@R2#set protocols bgp group RR1 neighbor 10.1.1.1
user@R2#set protocols bgp group RR1 neighbor 10.3.3.3
user@R2#set protocols bgp group RR1 neighbor 10.6.6.6
user@R2#set protocols bgp group RR1 neighbor 10.4.4.4
user@R2#set protocols bgp cluster 10.2.2.2
```

7. Configure IS-IS and MPLS protocol on the interfaces.

```
[edit]
user@R2#set protocols isis interface ge-0/0/0.0
user@R2#set protocols isis interface ge-0/0/1.0
user@R2#set protocols isis interface ge-0/0/2.0
user@R2#set protocols isis interface ge-0/0/3.0
user@R2#set protocols isis interface ge-0/0/4.0
user@R2#set protocols isis interface lo0.0 passive
user@R2#set protocols isis level 1 disable
user@R2#set protocols isis level 2 wide-metrics-only
user@R2#set protocols mpls interface ge-0/0/0.0 admin-group brown
```

```
user@R2#set protocols mpls interface ge-0/0/3.0 admin-group red
user@R2#set protocols mpls interface ge-0/0/4.0 admin-group blue
user@R2#set protocols mpls interface ge-0/0/1.0 admin-group brown
user@R2#set protocols mpls interface all
```

8. Configure MPLS administrative group policies for LSP path computation.

```
[edit]
user@R2#set protocols mpls admin-groups red 0
user@R2#set protocols mpls admin-groups blue 1
user@R2#set protocols mpls admin-groups brown 5
```

9. Configure the MPLS label range to assign static labels for the EPE links.

```
[edit]
user@R2#set protocols mpls label-range static-label-range 7000 70000
```

10. Enable link protection on all the RSVP interfaces. Using link protection, you can configure a network to reroute traffic quickly around broken links.

```
[edit]
user@R2#set protocols rsvp interface all link-protection
```

Results

From configuration mode, confirm your configuration by entering the `show chassis`, `show interfaces`, `show policy-options`, `show routing-options`, and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
chassis {
    network-services enhanced-ip;
}
interfaces {
    ge-0/0/0 {
        description To_R1;
        unit 0 {
            family inet {
                address 192.168.3.2/24;
```

```
            }
            family iso;
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/1 {
        description To_R3;
        unit 0 {
            family inet {
                address 192.168.6.1/24;
            }
            family iso;
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/2 {
        description To_R4;
        unit 0 {
            family inet {
                address 192.168.7.1/24;
            }
            family iso;
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/3 {
        description To_R5;
        unit 0 {
            family inet {
                address 192.168.8.1/24;
            }
            family iso;
            family mpls {
                maximum-labels 8;
            }
        }
    }
```

```
    ge-0/0/4 {
        description To_R6;
        unit 0 {
            family inet {
                address 192.168.9.1/24;
            }
            family iso;
            family mpls {
                maximum-labels 8;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.2.2.2/32;
            }
            family iso {
                address 49.0001.0002.0202.0200;
            }
        }
    }
}
policy-options {
    policy-statement bgplsepe_rt_2_ted {
        term 1 {
            from protocol bgp;
            then accept;
        }
    }
    policy-statement nlri2bgp {
        term 1 {
            from family traffic-engineering;
            then {
                next-hop self;
                accept;
            }
        }
    }
    policy-statement nlri2bgp_igp {
        term 1 {
            from {
                family traffic-engineering;
```

```
                protocol isis;
            }
            then accept;
        }
    }
    policy-statement nlri2ted_igp {
        term 1 {
            from {
                traffic-engineering {
                    protocol isis-level-2;
                }
            }
            then accept;
        }
    }
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
    }
    policy-statement ted2nlri {
        term 1 {
            from protocol bgp-ls-epe;
            then accept;
        }
    }
    policy-statement ted2nlri_1 {
        term 1 {
            from {
                traffic-engineering;
            }
            then accept;
        }
    }
    policy-statement ted2nlri_igp {
        term 1 {
            from {
                family traffic-engineering;
                protocol isis;
            }
            then accept;
        }
    }
```

```
}
routing-options {
    router-id 10.2.2.2;
    autonomous-system 65200;
}
protocols {
    bgp {
        group RR1 {
            type internal;
            local-address 10.2.2.2;
            family traffic-engineering {
                unicast;
            }
            neighbor 10.1.1.1;
            neighbor 10.3.3.3;
            neighbor 10.6.6.6;
            neighbor 10.4.4.4;
        }
        cluster 10.2.2.2;
    }
    isis {
        interface ge-0/0/0.0;
        interface ge-0/0/1.0;
        interface ge-0/0/2.0;
        interface ge-0/0/3.0;
        interface ge-0/0/4.0;
        interface lo0.0 {
            passive;
        }
        level 1 disable;
        level 2 wide-metrics-only;
    }
    mpls {
        admin-groups {
            red 0;
            blue 1;
            brown 5;
        }
        label-range {
            static-label-range 7000 70000;
        }
        interface ge-0/0/0.0 {
            admin-group brown;
```

```
        }
        interface ge-0/0/3.0 {
            admin-group red;
        }
        interface ge-0/0/4.0 {
            admin-group blue;
        }
        interface ge-0/0/1.0 {
            admin-group brown;
        }
        interface all;
    }
    rsvp {
        interface all {
            link-protection;
        }
    }
}
```

Configure R5 (Intermediate router)

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure device R5:

1. Configure the network services mode as Enhanced IP. Enhanced IP sets the router's network services to enhanced Internet Protocol and uses enhanced mode capabilities.

```
[edit]
user@R5#set chassis network-services enhanced-ip
```

After you configure the `enhanced-ip` statement and commit the configuration, the following warning message appears prompting you to reboot the router:

```
'chassis'
WARNING: Chassis configuration for network services has been changed. A system reboot is
mandatory. Please reboot the system NOW. Continuing without a reboot might result in
```

```
unexpected system behavior.
commit complete
```

The reboot brings up the FPCs on the router.

2. Configure the interfaces to enable IP, MPLS, and ISO transport.

```
user@R5#set interfaces ge-0/0/0 description To_R1
user@R5#set interfaces ge-0/0/0 unit 0 family inet address 192.168.5.2/24
user@R5#set interfaces ge-0/0/0 unit 0 family iso
user@R5#set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 8
user@R5#set interfaces ge-0/0/1 description To_R2
user@R5#set interfaces ge-0/0/1 unit 0 family inet address 192.168.8.2/24
user@R5#set interfaces ge-0/0/1 unit 0 family iso
user@R5#set interfaces ge-0/0/1 unit 0 family mpls maximum-labels 8
user@R5#set interfaces ge-0/0/2 description To_R3
user@R5#set interfaces ge-0/0/2 unit 0 family inet address 192.168.10.2/24
user@R5#set interfaces ge-0/0/2 unit 0 family iso
user@R5#set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
user@R5#set interfaces ge-0/0/3 description To_R4
user@R5#set interfaces ge-0/0/3 unit 0 family inet address 192.168.13.2/24
user@R5#set interfaces ge-0/0/3 unit 0 family iso
user@R5#set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
user@R5#set interfaces ge-0/0/4 description To_R6
user@R5#set interfaces ge-0/0/4 unit 0 family inet address 192.168.14.1/24
user@R5#set interfaces ge-0/0/4 unit 0 family iso
user@R5#set interfaces ge-0/0/4 unit 0 family mpls maximum-labels 8
```

3. Configure the loopback interface to enable tunnel endpoints and service endpoints.

```
[edit]
user@R5#set interfaces lo0 unit 0 family inet address 10.5.5.5/32
user@R5#set interfaces lo0 unit 0 family iso address 49.0001.0005.0505.0500
```

4. Configure routing options to identify the router in the domain.

```
[edit]
user@R5#set routing-options router-id 10.5.5.5
user@R5#set routing-options autonomous-system 65200
```

5. Define import and export policies. For example, configure policies that export EPE TE links from the local TE database to **lsdist.0** and policies to import from **lsdist.0** into the local TE database. You can configure policies to advertise the BGP routes to a peer.

```
[edit]
user@R5#set policy-options policy-statement nlri2bgp term 1 from family traffic-engineering
user@R5#set policy-options policy-statement nlri2bgp term 1 then next-hop self
user@R5#set policy-options policy-statement nlri2bgp term 1 then accept
user@R5#set policy-options policy-statement nlri2ted_igp term 1 from traffic-engineering
protocol isis-level-2
user@R5#set policy-options policy-statement nlri2ted_igp term 1 then accept
user@R5#set policy-options policy-statement pplb then load-balance per-packet
user@R5#set policy-options policy-statement ted2nlri term 1 from protocol bgp-ls-epe
user@R5#set policy-options policy-statement ted2nlri term 1 then accept
user@R5#set policy-options policy-statement ted2nlri_igp term 1 from family traffic-
engineering
user@R5#set policy-options policy-statement ted2nlri_igp term 1 from protocol isis
user@R5#set policy-options policy-statement ted2nlri_igp term 1 then accept
```

6. Configure IS-IS and MPLS protocol on the interfaces.

```
[edit]
user@R5#set protocols isis interface ge-0/0/0.0
user@R5#set protocols isis interface ge-0/0/1.0
user@R5#set protocols isis interface ge-0/0/2.0
user@R5#set protocols isis interface ge-0/0/3.0
user@R5#set protocols isis interface ge-0/0/4.0
user@R5#set protocols isis interface lo0.0 passive
user@R5#set protocols isis level 1 disable
user@R5#set protocols isis level 2 wide-metrics-only
user@R5#set protocols mpls interface ge-0/0/0.0 admin-group blue
user@R5#set protocols mpls interface ge-0/0/1.0 admin-group red
user@R5#set protocols mpls interface ge-0/0/3.0 admin-group brown
user@R5#set protocols mpls interface ge-0/0/4.0 admin-group brown
user@R5#set protocols mpls interface all
```

7. Configure BGP to enable BGP-LS route advertisement to the connected peers.

```
[edit]
user@R5#set protocols bgp group RR2 type internal
```

```
user@R5#set protocols bgp group RR2 family inet unicast
user@R5#set protocols bgp group RR2 family traffic-engineering unicast
user@R5#set protocols bgp group RR2 neighbor 10.1.1.1
user@R5#set protocols bgp group RR2 neighbor 10.3.3.3
user@R5#set protocols bgp group RR2 neighbor 10.6.6.6
user@R5#set protocols bgp group RR2 neighbor 10.4.4.4
user@R5#set protocols bgp cluster 10.5.5.5
```

8. Configure MPLS administrative group policies for LSP path computation.

```
[edit]
user@R5#set protocols mpls admin-groups red 0
user@R5#set protocols mpls admin-groups blue 1
user@R5#set protocols mpls admin-groups brown 5
```

9. Configure the MPLS label range to assign static labels for the EPE links.

```
[edit]
user@R5#set protocols mpls label-range static-label-range 7000 70000
```

10. Enable link protection on all the RSVP interfaces. Using link protection, you can configure a network to reroute traffic quickly around broken links.

```
[edit]
user@R5#set protocols rsvp interface all link-protection
```

Results

From configuration mode, confirm your configuration by entering the `show chassis`, `show interfaces`, `show policy-options`, `show routing-options`, and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
chassis {
    network-services enhanced-ip;
}
interfaces {
    ge-0/0/0 {
        description To_R1;
```

```
        unit 0 {
            family inet {
                address 192.168.5.2/24;
            }
            family iso;
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/1 {
        description To_R2;
        unit 0 {
            family inet {
                address 192.168.8.2/24;
            }
            family iso;
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/2 {
        description To_R3;
        unit 0 {
            family inet {
                address 192.168.10.2/24;
            }
            family iso;
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/3 {
        description To_R4;
        unit 0 {
            family inet {
                address 192.168.13.2/24;
            }
            family iso;
            family mpls {
                maximum-labels 8;
```

```
            }
        }
    }
    ge-0/0/4 {
        description To_R6;
        unit 0 {
            family inet {
                address 192.168.14.1/24;
            }
            family iso;
            family mpls {
                maximum-labels 8;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.5.5.5/32;
            }
            family iso {
                address 49.0001.0005.0505.0500;
            }

        }
    }
}
policy-options {
    policy-statement nlri2bgp {
        term 1 {
            from family traffic-engineering;
            then {
                next-hop self;
                accept;
            }
        }
    }
    policy-statement nlri2ted_igp {
        term 1 {
            from {
                traffic-engineering {
                    protocol isis-level-2;
                }
```

```
            }
            then accept;
        }
    }
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
    }
    policy-statement ted2nlri {
        term 1 {
            from protocol bgp-ls-epe;
            then accept;
        }
    }
    policy-statement ted2nlri_igp {
        term 1 {
            from {
                family traffic-engineering;
                protocol isis;
            }
            then accept;
        }
    }
}
routing-options {
    router-id 10.5.5.5;
    autonomous-system 65200;

}
protocols {
    bgp {
        group RR2 {
            type internal;
            family inet {
                unicast;
            }
            family traffic-engineering {
                unicast;
            }
            neighbor 10.1.1.1;
            neighbor 10.3.3.3;
            neighbor 10.6.6.6;
```

```
            neighbor 10.4.4.4;
        }
        cluster 10.5.5.5;
    }
    isis {
        interface ge-0/0/0.0;
        interface ge-0/0/1.0;
        interface ge-0/0/2.0;
        interface ge-0/0/3.0;
        interface ge-0/0/4.0;
        interface lo0.0 {
            passive;
        }
        level 1 disable;
        level 2 wide-metrics-only;
    }
    mpls {
        admin-groups {
            red 0;
            blue 1;
            brown 5;
        }
        label-range {
            static-label-range 7000 70000;
        }
        interface ge-0/0/0.0 {
            admin-group blue;
        }
        interface ge-0/0/1.0 {
            admin-group red;
        }
        interface ge-0/0/3.0 {
            admin-group brown;
        }
        interface ge-0/0/4.0 {
            admin-group brown;
        }
        interface all;
    }
    rsvp {
        interface all {
            link-protection;
        }
```

```
    }
}
```

Configure R3 (BN3 router)

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure device R3:

1. Configure the network services mode as Enhanced IP. Enhanced IP sets the router's network services to enhanced Internet Protocol and uses enhanced mode capabilities.

   ```
   [edit]
   user@R3#set chassis network-services enhanced-ip
   ```

   After you configure the `enhanced-ip` statement and commit the configuration, the following warning message appears prompting you to reboot the router:

   ```
   'chassis'
   WARNING: Chassis configuration for network services has been changed. A system reboot is
   mandatory. Please reboot the system NOW. Continuing without a reboot might result in
   unexpected system behavior.
   commit complete
   ```

   The reboot brings up the FPCs on the router.

2. Configure the interfaces to enable IP, MPLS, and ISO transport.

   ```
   user@R3#set interfaces ge-0/0/0 description To_R2
   user@R3#set interfaces ge-0/0/0 unit 0 family inet address 192.168.6.2/24
   user@R3#set interfaces ge-0/0/0 unit 0 family iso
   user@R3#set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 8
   user@R3#set interfaces ge-0/0/1 description To_R5
   user@R3#set interfaces ge-0/0/1 unit 0 family inet address 192.168.10.1/24
   user@R3#set interfaces ge-0/0/1 unit 0 family iso
   user@R3#set interfaces ge-0/0/1 unit 0 family mpls maximum-labels 8
   user@R3#set interfaces ge-0/0/2 description To_R6
   ```

```
user@R3#set interfaces ge-0/0/2 unit 0 family inet address 192.168.11.1/24
user@R3#set interfaces ge-0/0/2 unit 0 family iso
user@R3#set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
user@R3#set interfaces ge-0/0/3 description To_R7
user@R3#set interfaces ge-0/0/3 unit 0 family inet address 192.168.12.1/24
user@R3#set interfaces ge-0/0/3 unit 0 family iso
user@R3#set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
```

3. Configure the loopback interface to enable tunnel endpoints and service endpoints.

```
[edit]
user@R3#set interfaces lo0 unit 0 family inet address 10.3.3.3/32
user@R3#set interfaces lo0 unit 0 family iso address 49.0001.0003.0303.0300
```

4. Configure routing options to identify the router in the domain.

```
[edit]
user@R3#set routing-options router-id 10.3.3.3
user@R3#set routing-options autonomous-system 65200
```

5. Define import and export policies. For example, configure policies that export EPE TE links from the local TE database to **lsdist.0** and policies to import from **lsdist.0** into the local TE database. You can configure policies to advertise the BGP routes to a peer.

```
[edit]
user@R3#set policy-options policy-statement expresspol1 from route-filter 10.1.1.1/32 exact
install-nexthop lsp lsp3to1_a
user@R3#set policy-options policy-statement expresspol1 then accept
user@R3#set policy-options policy-statement expresspol2 from route-filter 10.4.4.4/32 exact
install-nexthop lsp lsp3to4_a
user@R3#set policy-options policy-statement expresspol2 then accept
user@R3#set policy-options policy-statement nlri2bgp_epe term 1 from family traffic-
engineering
user@R3#set policy-options policy-statement nlri2bgp_epe term 1 from protocol bgp-ls-epe
user@R3#set policy-options policy-statement nlri2bgp_epe term 1 then next-hop self
user@R3#set policy-options policy-statement nlri2bgp_epe term 1 then accept
user@R3#set policy-options policy-statement nlri2bgp_stat term 1 from family traffic-
engineering
user@R3#set policy-options policy-statement nlri2bgp_stat term 1 from protocol express-
segments
```

```
user@R3#set policy-options policy-statement nlri2bgp_stat term 1 then accept
user@R3#set policy-options policy-statement pplb then load-balance per-packet
user@R3#set policy-options policy-statement ted2nlri_epe_stat term 1 from family traffic-
engineering
user@R3#set policy-options policy-statement ted2nlri_epe_stat term 1 from protocol static
user@R3#set policy-options policy-statement ted2nlri_epe_stat term 1 then accept
user@R3#set policy-options policy-statement ted2nlri_epe_stat term 2 from family traffic-
engineering
user@R3#set policy-options policy-statement ted2nlri_epe_stat term 2 from protocol bgp-ls-
epe
user@R3#set policy-options policy-statement ted2nlri_epe_stat term 2 then accept
user@R3#set policy-options policy-statement ted2nlri_epe_stat term 3 from protocol isis
user@R3#set policy-options policy-statement ted2nlri_epe_stat term 3 then reject
```

6. Configure BGP to enable BGP-LS route advertisement for peer and define the EPE links. Since express segment is an internal TE link, this configuration creates an external TE link.

```
[edit]
user@R3#set protocols bgp group ibgp1 type internal
user@R3#set protocols bgp group ibgp1 local-address 10.3.3.3
user@R3#set protocols bgp group ibgp1 family traffic-engineering unicast
user@R3#set protocols bgp group ibgp1 export nlri2bgp_epe
user@R3#set protocols bgp group ibgp1 neighbor 10.2.2.2
user@R3#set protocols bgp group ibgp1 neighbor 10.5.5.5
user@R3#set protocols bgp group ebgp1 type external
user@R3#set protocols bgp group ebgp1 family traffic-engineering unicast
user@R3#set protocols bgp group ebgp1 export nlri2bgp_stat
user@R3#set protocols bgp group ebgp1 neighbor 192.168.12.2 peer-as 65300
user@R3#set protocols bgp group ebgp1 neighbor 192.168.12.2 egress-te-adj-segment
epe_adj1_toR7 label 7137
user@R3#set protocols bgp group ebgp1 neighbor 192.168.12.2 egress-te-adj-segment
epe_adj1_toR7 next-hop 192.168.12.2
user@R3#set protocols bgp group ebgp1 neighbor 192.168.12.2 egress-te-adj-segment
epe_adj1_toR7 te-link-attribute te-metric 20
user@R3#set protocols bgp group ebgp1 neighbor 192.168.12.2 egress-te-adj-segment
epe_adj1_toR7 te-link-attribute igp-metric 10
user@R3#set protocols bgp group ebgp1 neighbor 192.168.12.2 egress-te-adj-segment
epe_adj1_toR7 te-link-attribute admin-group red
user@R3#set protocols bgp group ebgp1 neighbor 192.168.12.2 egress-te-adj-segment
epe_adj1_toR7 te-link-attribute admin-group brown
user@R3#set protocols bgp group ebgp1 vpn-apply-export
```

7. Define a mechanism to automatically (dynamic) create express segments and insert them in to the TE database so that they can be advertised through BGP-LS. In this example, express segments are created for all the underlay RSVP tunnels automatically. This is done by configuring a template with a policy and then express segments are automatically created based on the policies.

```
[edit]
user@R3#set protocols express-segments segment-set set1 membership-policy expresspol1
user@R3#set protocols express-segments segment-set set2 membership-policy expresspol2
user@R3#set protocols express-segments traffic-engineering
```

8. Configure IS-IS and MPLS protocol on the interfaces.

```
[edit]
user@R3#set protocols isis interface ge-0/0/0.0
user@R3#set protocols isis interface ge-0/0/1.0
user@R3#set protocols isis interface ge-0/0/2.0
user@R3#set protocols isis interface ge-0/0/3.0 passive
user@R3#set protocols isis interface lo0.0 passive
user@R3#set protocols isis level 1 disable
user@R3#set protocols isis level 2 wide-metrics-only
user@R3#set protocols mpls interface ge-0/0/0.0 admin-group brown
user@R3#set protocols mpls interface ge-0/0/2.0 admin-group red
user@R3#set protocols mpls interface ge-0/0/3.0 admin-group red
user@R3#set protocols mpls interface ge-0/0/3.0 admin-group brown
user@R3#set protocols mpls interface all
```

9. Enable import and export of traffic engineering database parameters using policies.

```
[edit]
user@R3#set protocols mpls traffic-engineering database import l3-unicast-topology bgp-link-
state
user@R3#set protocols mpls traffic-engineering database import policy ted2nlri_epe_stat
user@R3#set protocols mpls traffic-engineering database export l3-unicast-topology
```

10. Configure MPLS administrative group policies for LSP path computation.

```
[edit]
user@R3#set protocols mpls admin-groups red 0
```

```
user@R3#set protocols mpls admin-groups blue 1
user@R3#set protocols mpls admin-groups brown 5
```

11. Configure MPLS with a label-switched path (LSP) and include administrative groups.

```
[edit]
user@R3#set protocols mpls label-switched-path lsp3to1_a to 10.1.1.1
user@R3#set protocols mpls label-switched-path lsp3to1_a admin-group include-any red
user@R3#set protocols mpls label-switched-path lsp3to1_a admin-group include-any brown
user@R3#set protocols mpls label-switched-path lsp3to4_a to 10.4.4.4
user@R3#set protocols mpls label-switched-path lsp3to4_a admin-group include-any red
user@R3#set protocols mpls label-switched-path lsp3to4_a admin-group include-any brown
```

12. Configure the MPLS label range to assign static labels for the EPE links.

```
[edit]
user@R3#set protocols mpls label-range static-label-range 7000 70000
```

13. Enable link protection on all the RSVP interfaces. Using link protection, you can configure a network to reroute traffic quickly around broken links.

```
[edit]
user@R3#set protocols rsvp interface all link-protection
```

Results

From configuration mode, confirm your configuration by entering the `show chassis`, `show interfaces`, `show policy-options`, `show routing-options`, and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
chassis {
    network-services enhanced-ip;
}
interfaces {
    ge-0/0/0 {
        description To_R2;
        unit 0 {
            family inet {
```

```
                address 192.168.6.2/24;
            }
            family iso;
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/1 {
        description To_R5;
        unit 0 {
            family inet {
                address 192.168.10.1/24;
            }
            family iso;
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/2 {
        description To_R6;
        unit 0 {
            family inet {
                address 192.168.11.1/24;
            }
            family iso;
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/3 {
        description To_R7;
        unit 0 {
            family inet {
                address 192.168.12.1/24;
            }
            family iso;
            family mpls {
                maximum-labels 8;
            }
        }
```

```
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.3.3.3/32;
            }
            family iso {
                address 49.0001.0003.0303.0300;
            }
        }
    }
}
policy-options {
    policy-statement expresspol1 {
        from {
            route-filter 10.1.1.1/32 exact {
                install-nexthop lsp lsp3to1_a;
            }
        }
        then accept;
    }
    policy-statement expresspol2 {
        from {
            route-filter 10.4.4.4/32 exact {
                install-nexthop lsp lsp3to4_a;
            }
        }
        then accept;
    }
    policy-statement nlri2bgp_epe {
        term 1 {
            from {
                family traffic-engineering;
                protocol bgp-ls-epe;
            }
            then {
                next-hop self;
                accept;
            }
        }
    }
    policy-statement nlri2bgp_stat {
        term 1 {
```

```
            from {
                family traffic-engineering;
                protocol express-segments;
            }
            then accept;
        }
    }
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
    }
    policy-statement ted2nlri_epe_stat {
        term 1 {
            from {
                family traffic-engineering;
                protocol static;
            }
            then accept;
        }
        term 2 {
            from {
                family traffic-engineering;
                protocol bgp-ls-epe;
            }
            then accept;
        }
        term 3 {
            from protocol isis;
            then reject;
        }
    }
}
routing-options {
    router-id 10.3.3.3;
    autonomous-system 65200;
}
protocols {
    bgp {
        group ibgp1 {
            type internal;
            local-address 10.3.3.3;
            family traffic-engineering {
```

```
                unicast;
            }
            export nlri2bgp_epe;
            neighbor 10.2.2.2;
            neighbor 10.5.5.5;
        }
        group ebgp1 {
            type external;
            family traffic-engineering {
                unicast;
            }
            export nlri2bgp_stat;
            neighbor 192.168.12.2 {
                peer-as 65300;
                egress-te-adj-segment epe_adj1_toR7 {
                    label 7137;
                    next-hop 192.168.12.2;
                    te-link-attribute {
                        te-metric 20;
                        igp-metric 10;
                        admin-group [ red brown ];
                    }
                }
            }
            vpn-apply-export;
        }
    }
    express-segments {
        segment-set set1 {
            membership-policy expresspol1;
        }
        segment-set set2 {
            membership-policy expresspol2;
        }
        traffic-engineering;
    }
    isis {
        interface ge-0/0/0.0;
        interface ge-0/0/1.0;
        interface ge-0/0/2.0;
        interface ge-0/0/3.0 {
            passive;
        }
```

```
        interface lo0.0 {
            passive;
        }
        level 1 disable;
        level 2 wide-metrics-only;
    }
    mpls {
        traffic-engineering {
            database {
                import {
                    l3-unicast-topology {
                        bgp-link-state;
                    }
                    policy ted2nlri_epe_stat;
                }
                export {
                    l3-unicast-topology;
                }
            }
        }
        admin-groups {
            red 0;
            blue 1;
            brown 5;
        }
        label-switched-path lsp3to1_a {
            to 10.1.1.1;
            admin-group include-any [ red brown ];
        }
        label-switched-path lsp3to4_a {
            to 10.4.4.4;
            admin-group include-any [ red brown ];
        }
        label-range {
            static-label-range 7000 70000;
        }
        interface ge-0/0/0.0 {
            admin-group brown;
        }
        interface ge-0/0/2.0 {
            admin-group red;
        }
        interface ge-0/0/3.0 {
```

```
            admin-group [ red brown ];
        }
        interface all;
    }
    rsvp {
        interface all {
            link-protection;
        }
    }
}
```

Configure R6 (BN4 router)

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure device R6:

1. Configure the network services mode as Enhanced IP. Enhanced IP sets the router's network services to enhanced Internet Protocol and uses enhanced mode capabilities.

   ```
   [edit]
   user@R6#set chassis network-services enhanced-ip
   ```

   After you configure the `enhanced-ip` statement and commit the configuration, the following warning message appears prompting you to reboot the router:

   ```
   'chassis'
   WARNING: Chassis configuration for network services has been changed. A system reboot is
   mandatory. Please reboot the system NOW. Continuing without a reboot might result in
   unexpected system behavior.
   commit complete
   ```

   The reboot brings up the FPCs on the router.

2. Configure the interfaces to enable IP, MPLS, and ISO transport.

```
user@R6#set interfaces ge-0/0/0 description To_R2
user@R6#set interfaces ge-0/0/0 unit 0 family inet address 192.168.9.2/24
user@R6#set interfaces ge-0/0/0 unit 0 family iso
user@R6#set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 8
user@R6#set interfaces ge-0/0/1 description To_R3
user@R6#set interfaces ge-0/0/1 unit 0 family inet address 192.168.11.2/24
user@R6#set interfaces ge-0/0/1 unit 0 family iso
user@R6#set interfaces ge-0/0/1 unit 0 family mpls maximum-labels 8
user@R6#set interfaces ge-0/0/2 description To_R5
user@R6#set interfaces ge-0/0/2 unit 0 family inet address 192.168.14.2/24
user@R6#set interfaces ge-0/0/2 unit 0 family iso
user@R6#set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
user@R6#set interfaces ge-0/0/3 description To_R7
user@R6#set interfaces ge-0/0/3 unit 0 family inet address 192.168.15.1/24
user@R6#set interfaces ge-0/0/3 unit 0 family iso
user@R6#set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
```

3. Configure the loopback interface to enable tunnel endpoints and service endpoints.

```
[edit]
user@R6#set interfaces lo0 unit 0 family inet address 10.6.6.6/32
user@R6#set interfaces lo0 unit 0 family iso address 49.0001.0006.0606.0600
```

4. Configure routing options to identify the router in the domain.

```
[edit]
user@R6#set routing-options router-id 10.6.6.6
user@R6#set routing-options autonomous-system 65200
```

5. Define import and export policies. For example, configure policies that export EPE TE links from the local TE database to **lsdist.0** and policies to import from **lsdist.0** into the local TE database. You can configure policies to advertise the BGP routes to a peer.

```
[edit]
user@R6#set policy-options policy-statement expresspol1 from route-filter 10.1.1.1/32 exact
install-nexthop lsp lsp6to1_a
user@R6#set policy-options policy-statement expresspol1 then accept
```

```
user@R6#set policy-options policy-statement expresspol2 from route-filter 10.4.4.4/32 exact
install-nexthop lsp lsp6to4_a
user@R6#set policy-options policy-statement expresspol2 then accept
user@R6#set policy-options policy-statement nlri2bgp_epe term 1 from family traffic-
engineering
user@R6#set policy-options policy-statement nlri2bgp_epe term 1 from protocol bgp-ls-epe
user@R6#set policy-options policy-statement nlri2bgp_epe term 1 then next-hop self
user@R6#set policy-options policy-statement nlri2bgp_epe term 1 then accept
user@R6#set policy-options policy-statement nlri2bgp_stat term 1 from family traffic-
engineering
user@R6#set policy-options policy-statement nlri2bgp_stat term 1 from protocol express-
segments
user@R6#set policy-options policy-statement nlri2bgp_stat term 1 then accept
user@R6#set policy-options policy-statement pplb then load-balance per-packet
user@R6#set policy-options policy-statement ted2nlri_epe_stat term 1 from family traffic-
engineering
user@R6#set policy-options policy-statement ted2nlri_epe_stat term 1 from protocol static
user@R6#set policy-options policy-statement ted2nlri_epe_stat term 1 then accept
user@R6#set policy-options policy-statement ted2nlri_epe_stat term 2 from family traffic-
engineering
user@R6#set policy-options policy-statement ted2nlri_epe_stat term 2 from protocol bgp-ls-
epe
user@R6#set policy-options policy-statement ted2nlri_epe_stat term 2 then accept
user@R6#set policy-options policy-statement ted2nlri_epe_stat term 3 from protocol isis
user@R6#set policy-options policy-statement ted2nlri_epe_stat term 3 then reject
```

6. Configure BGP to enable BGP-LS route advertisement for peer and define the EPE links. Since express segment is an internal TE link, this configuration creates an external TE link.

```
[edit]
user@R6#set protocols bgp group ibgp1 type internal
user@R6#set protocols bgp group ibgp1 local-address 10.6.6.6
user@R6#set protocols bgp group ibgp1 family traffic-engineering unicast
user@R6#set protocols bgp group ibgp1 export nlri2bgp_epe
user@R6#set protocols bgp group ibgp1 neighbor 10.2.2.2
user@R6#set protocols bgp group ibgp1 neighbor 10.5.5.5
user@R6#set protocols bgp group ebgp1 type external
user@R6#set protocols bgp group ebgp1 family traffic-engineering unicast
user@R6#set protocols bgp group ebgp1 export nlri2bgp_stat
user@R6#set protocols bgp group ebgp1 neighbor 192.168.15.2 peer-as 300
user@R6#set protocols bgp group ebgp1 neighbor 192.168.15.2 egress-te-adj-segment
epe_adj1_toR7 label 7167
```

```
user@R6#set protocols bgp group ebgp1 neighbor 192.168.15.2 egress-te-adj-segment
epe_adj1_toR7 next-hop 192.168.15.2
user@R6#set protocols bgp group ebgp1 neighbor 192.168.15.2 egress-te-adj-segment
epe_adj1_toR7 te-link-attribute te-metric 20
user@R6#set protocols bgp group ebgp1 neighbor 192.168.15.2 egress-te-adj-segment
epe_adj1_toR7 te-link-attribute igp-metric 10
user@R6#set protocols bgp group ebgp1 neighbor 192.168.15.2 egress-te-adj-segment
epe_adj1_toR7 te-link-attribute admin-group red
user@R6#set protocols bgp group ebgp1 neighbor 192.168.15.2 egress-te-adj-segment
epe_adj1_toR7 te-link-attribute admin-group brown
```

7. Define a mechanism to automatically (dynamic) create express segments and insert them in to the TE database so that they can be advertised through BGP-LS. In this example, express segments are created for all the underlay RSVP tunnels automatically. This is done by configuring a template with a policy and then express segments are automatically created based on the policies.

```
[edit]
user@R6#set protocols express-segments segment-set set1 membership-policy expresspol1
user@R6#set protocols express-segments segment-set set2 membership-policy expresspol2
user@R6#set protocols express-segments traffic-engineering
```

8. Configure IS-IS and MPLS protocol on the interfaces.

```
[edit]
user@R6#set protocols isis interface ge-0/0/0.0
user@R6#set protocols isis interface ge-0/0/1.0
user@R6#set protocols isis interface ge-0/0/2.0
user@R6#set protocols isis interface lo0.0 passive
user@R6#set protocols isis level 1 disable
user@R6#set protocols isis level 2 wide-metrics-only
user@R6#set protocols mpls interface ge-0/0/0.0 admin-group blue
user@R6#set protocols mpls interface ge-0/0/1.0 admin-group red
user@R6#set protocols mpls interface ge-0/0/2.0 admin-group brown
user@R6#set protocols mpls interface ge-0/0/3.0 admin-group red
user@R6#set protocols mpls interface ge-0/0/3.0 admin-group brown
user@R6#set protocols mpls interface all
```

9. Enable import and export of traffic engineering database parameters using policies.

```
[edit]
user@R6#set protocols mpls traffic-engineering database import l3-unicast-topology bgp-link-state
user@R6#set protocols mpls traffic-engineering database import policy ted2nlri_epe_stat
user@R6#set protocols mpls traffic-engineering database export l3-unicast-topology
```

10. Configure MPLS administrative group policies for LSP path computation.

```
[edit]
user@R6#set protocols mpls admin-groups red 0
user@R6#set protocols mpls admin-groups blue 1
user@R6#set protocols mpls admin-groups brown 5
```

11. Configure MPLS with a label-switched path (LSP) and include administrative groups.

```
[edit]
user@R6#set protocols mpls label-switched-path lsp6to1_a to 10.1.1.1
user@R6#set protocols mpls label-switched-path lsp6to1_a admin-group include-any red
user@R6#set protocols mpls label-switched-path lsp6to1_a admin-group include-any brown
user@R6#set protocols mpls label-switched-path lsp6to4_a to 10.4.4.4
user@R6#set protocols mpls label-switched-path lsp6to4_a admin-group include-any red
user@R6#set protocols mpls label-switched-path lsp6to4_a admin-group include-any brown
```

12. Configure the MPLS label range to assign static labels for the EPE links.

```
[edit]
user@R6#set protocols mpls label-range static-label-range 7000 70000
```

13. Enable link protection on all the RSVP interfaces. Using link protection, you can configure a network to reroute traffic quickly around broken links.

```
[edit]
user@R6#set protocols rsvp interface all link-protection
```

Results

From configuration mode, confirm your configuration by entering the `show chassis`, `show interfaces`, `show policy-options`, `show routing-options`, and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
chassis {
    network-services enhanced-ip;
}
interfaces {
    ge-0/0/0 {
        description To_R2;
        unit 0 {
            family inet {
                address 192.168.9.2/24;
            }
            family iso;
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/1 {
        description To_R3;
        unit 0 {
            family inet {
                address 192.168.11.2/24;
            }
            family iso;
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/2 {
        description To_R5;
        unit 0 {
            family inet {
                address 192.168.14.2/24;
            }
            family iso;
            family mpls {
```

```
                maximum-labels 8;
            }
        }
    }
    ge-0/0/3 {
        description To_R7;
        unit 0 {
            family inet {
                address 192.168.15.1/24;
            }
            family iso;
            family mpls {
                maximum-labels 8;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.6.6.6/32;
            }
            family iso {
                address 49.0001.0006.0606.0600;
            }
        }
    }
}
policy-options {
    policy-statement expresspol1 {
        from {
            route-filter 10.1.1.1/32 exact {
                install-nexthop lsp lsp6to1_a;
            }
        }
        then accept;
    }
    policy-statement expresspol2 {
        from {
            route-filter 10.4.4.4/32 exact {
                install-nexthop lsp lsp6to4_a;
            }
        }
        then accept;
```

```
    }
    policy-statement nlri2bgp_epe {
        term 1 {
            from {
                family traffic-engineering;
                protocol bgp-ls-epe;
            }
            then {
                next-hop self;
                accept;
            }
        }
    }
    policy-statement nlri2bgp_stat {
        term 1 {
            from {
                family traffic-engineering;
                protocol express-segments;
            }
            then accept;
        }
    }
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
    }
    policy-statement ted2nlri_epe_stat {
        term 1 {
            from {
                family traffic-engineering;
                protocol static;
            }
            then accept;
        }
        term 2 {
            from {
                family traffic-engineering;
                protocol bgp-ls-epe;
            }
            then accept;
        }
        term 3 {
```

```
            from protocol isis;
            then reject;
        }
    }
}
routing-options {
    router-id 10.6.6.6;
    autonomous-system 65200;
    forwarding-table {
        export pplb;
    }
}
protocols {
    bgp {
        group ibgp1 {
            type internal;
            local-address 10.6.6.6;
            family traffic-engineering {
                unicast;
            }
            export nlri2bgp_epe;
            neighbor 10.2.2.2;
            neighbor 10.5.5.5;
        }
        group ebgp1 {
            type external;
            family traffic-engineering {
                unicast;
            }
            export nlri2bgp_stat;
            neighbor 192.168.15.2 {
                peer-as 65300;
                egress-te-adj-segment epe_adj1_toR7 {
                    label 7167;
                    next-hop 192.168.15.2;
                    te-link-attribute {
                        te-metric 20;
                        igp-metric 10;
                        admin-group [ red brown ];
                    }
                }
            }
        }
```

```
    }
    express-segments {
        segment-set set1 {
            membership-policy expresspol1;
        }
        segment-set set2 {
            membership-policy expresspol2;
        }
        traffic-engineering;
    }
    isis {
        interface ge-0/0/0.0;
        interface ge-0/0/1.0;
        interface ge-0/0/2.0;
        interface lo0.0 {
            passive;
        }
        level 1 disable;
        level 2 wide-metrics-only;
    }
    mpls {
        traffic-engineering {
            database {
                import {
                    l3-unicast-topology {
                        bgp-link-state;
                    }
                    policy ted2nlri_epe_stat;
                }
                export {
                    l3-unicast-topology;
                }
            }
        }
        admin-groups {
            red 0;
            blue 1;
            brown 5;
        }
        label-switched-path lsp6to1_a {
            to 10.1.1.1;
            admin-group include-any [ red brown ];
        }
```

```
        label-switched-path lsp6to4_a {
            to 10.4.4.4;
            admin-group include-any [ red brown ];
        }
        label-range {
            static-label-range 7000 70000;
        }
        interface ge-0/0/0.0 {
            admin-group blue;
        }
        interface ge-0/0/1.0 {
            admin-group red;
        }
        interface ge-0/0/2.0 {
            admin-group brown;
        }
        interface ge-0/0/3.0 {
            admin-group [ red brown ];
        }
        interface all;
    }
    rsvp {
        interface all {
            link-protection;
        }
    }
}
```

Configure R7 (PE2 router)

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure device R7:

1. Configure the network services mode as Enhanced IP. Enhanced IP sets the router's network services to enhanced Internet Protocol and uses enhanced mode capabilities.

```
[edit]
user@R7#set chassis network-services enhanced-ip
```

After you configure the `enhanced-ip` statement and commit the configuration, the following warning message appears prompting you to reboot the router:

```
'chassis'
WARNING: Chassis configuration for network services has been changed. A system reboot is
mandatory. Please reboot the system NOW. Continuing without a reboot might result in
unexpected system behavior.
commit complete
```

The reboot brings up the FPCs on the router.

2. Configure the interfaces to enable IP, MPLS, and ISO transport.

```
user@R7#set interfaces ge-0/0/0 description To_R3
user@R7#set interfaces ge-0/0/0 unit 0 family inet address 192.168.12.2/24
user@R7#set interfaces ge-0/0/0 unit 0 family iso
user@R7#set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 8
user@R7#set interfaces ge-0/0/1 description To_R6
user@R7#set interfaces ge-0/0/1 unit 0 family inet address 192.168.15.2/24
user@R7#set interfaces ge-0/0/1 unit 0 family iso
user@R7#set interfaces ge-0/0/1 unit 0 family mpls maximum-labels 8
```

3. Configure the loopback interface to enable tunnel endpoints and service endpoints.

```
[edit]
user@R7#set interfaces lo0 unit 0 family inet address 10.7.7.7/32
user@R7#set interfaces lo0 unit 0 family inet address 10.7.7.71/32
```

4. Configure routing options to identify the router in the domain.

```
[edit]
user@R7#set routing-options router-id 10.7.7.7
user@R7#set routing-options autonomous-system 65300
```

```
user@R7#set routing-options static route 10.100.100.101/32 next-hop 10.100.100.100
user@R7#set routing-options static route 10.100.100.101/32 resolve
```

5. Define import and export policies. For example, configure policies that export EPE TE links from the local TE database to **lsdist.0** and policies to import from **lsdist.0** into the local TE database. You can configure policies to advertise the BGP routes to a peer.

```
[edit]
user@R7#set policy-options policy-statement nlri2bgp_epe term 1 from family traffic-
engineering
user@R7#set policy-options policy-statement nlri2bgp_epe term 1 from protocol bgp-ls-epe
user@R7#set policy-options policy-statement nlri2bgp_epe term 1 then next-hop self
user@R7#set policy-options policy-statement nlri2bgp_epe term 1 then accept
user@R7#set policy-options policy-statement nlri2ted_bgp term 1 from protocol bgp
user@R7#set policy-options policy-statement nlri2ted_bgp term 1 then accept
user@R7#set policy-options policy-statement pplb then load-balance per-packet
user@R7#set policy-options policy-statement ted2nlri term 1 from protocol bgp-ls-epe
user@R7#set policy-options policy-statement ted2nlri term 1 then accept
user@R7#set policy-options resolution-map map1 mode ip-color
```

6. Configure BGP to enable BGP-LS route advertisement for peer and define the EPE links. Since express segment is an internal TE link, this configuration creates an external TE link.

```
[edit]
user@R7#set protocols bgp group ebgp1 type external
user@R7#set protocols bgp group ebgp1 family inet unicast
user@R7#set protocols bgp group ebgp1 family traffic-engineering unicast
user@R7#set protocols bgp group ebgp1 export nlri2bgp_epe
user@R7#set protocols bgp group ebgp1 neighbor 192.168.12.1 peer-as 200
user@R7#set protocols bgp group ebgp1 neighbor 192.168.12.1 egress-te-adj-segment
epe_adj1_toR3 label 8173
user@R7#set protocols bgp group ebgp1 neighbor 192.168.12.1 egress-te-adj-segment
epe_adj1_toR3 next-hop 192.168.12.1
user@R7#set protocols bgp group ebgp1 neighbor 192.168.12.1 egress-te-adj-segment
epe_adj1_toR3 te-link-attribute te-metric 20
user@R7#set protocols bgp group ebgp1 neighbor 192.168.12.1 egress-te-adj-segment
epe_adj1_toR3 te-link-attribute igp-metric 10
user@R7#set protocols bgp group ebgp1 neighbor 192.168.12.1 egress-te-adj-segment
epe_adj1_toR3 te-link-attribute admin-group red
user@R7#set protocols bgp group ebgp1 neighbor 192.168.12.1 egress-te-adj-segment
epe_adj1_toR3 te-link-attribute admin-group brown
```

```
user@R7#set protocols bgp group ebgp1 neighbor 192.168.15.1 peer-as 65200
user@R7#set protocols bgp group ebgp1 neighbor 192.168.15.1 egress-te-adj-segment
epe_adj1_toR6 label 8176
user@R7#set protocols bgp group ebgp1 neighbor 192.168.15.1 egress-te-adj-segment
epe_adj1_toR6 next-hop 192.168.15.1
user@R7#set protocols bgp group ebgp1 neighbor 192.168.15.1 egress-te-adj-segment
epe_adj1_toR6 te-link-attribute te-metric 20
user@R7#set protocols bgp group ebgp1 neighbor 192.168.15.1 egress-te-adj-segment
epe_adj1_toR6 te-link-attribute igp-metric 10
user@R7#set protocols bgp group ebgp1 neighbor 192.168.15.1 egress-te-adj-segment
epe_adj1_toR6 te-link-attribute admin-group red
user@R7#set protocols bgp group ebgp1 neighbor 192.168.15.1 egress-te-adj-segment
epe_adj1_toR6 te-link-attribute admin-group brown
```

7. Configure MPLS protocol on the interfaces.

```
[edit]
user@R7#set protocols mpls interface all
```

8. Enable import and export of traffic engineering database parameters using policies.

```
[edit]
user@R7#set protocols mpls traffic-engineering database import l3-unicast-topology bgp-link-
state
user@R7#set protocols mpls traffic-engineering database import policy ted2nlri
user@R7#set protocols mpls traffic-engineering database export policy nlri2ted_bgp
user@R7#set protocols mpls traffic-engineering database export l3-unicast-topology
```

9. Configure MPLS administrative group policies for LSP path computation.

```
[edit]
user@R7#set protocols mpls admin-groups red 0
user@R7#set protocols mpls admin-groups blue 1
user@R7#set protocols mpls admin-groups brown 5
```

10. Configure the MPLS label range to assign static labels for the EPE links.

```
[edit]
user@R7#set protocols mpls label-range static-label-range 7000 70000
```

11. Configure SR-TE policies on the ingress router to enable end-to-end SR-TE policy.

```
[edit]
user@R7#set protocols source-packet-routing compute-profile compute1 no-label-stack-
compression
user@R7#set protocols source-packet-routing source-routing-path computelsp1 to
10.100.100.100
user@R7#set protocols source-packet-routing source-routing-path computelsp1 install
10.100.100.101
user@R7#set protocols source-packet-routing source-routing-path computelsp1 primary p1
compute compute1
```

Results

From configuration mode, confirm your configuration by entering the `show chassis`, `show interfaces`, `show policy-options`, `show routing-options`, and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
chassis {
    network-services enhanced-ip;
}
interfaces {
    ge-0/0/0 {
        description To_R3;
        unit 0 {
            family inet {
                address 192.168.12.2/24;
            }
            family iso;
            family mpls {
                maximum-labels 8;
            }
        }
    }
```

```
    ge-0/0/1 {
        description To_R6;
        unit 0 {
            family inet {
                address 192.168.15.2/24;
            }
            family iso;
            family mpls {
                maximum-labels 8;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.7.7.7/32;
                address 10.7.7.71/32;
            }
            family iso {
                address 49.0001.0007.0707.0700;
            }
        }
    }
}
policy-options {
    policy-statement nlri2bgp_epe {
        term 1 {
            from {
                family traffic-engineering;
                protocol bgp-ls-epe;
            }
            then {
                next-hop self;
                accept;
            }
        }
    }
    policy-statement nlri2ted_bgp {
        term 1 {
            from protocol bgp;
            then accept;
        }
    }
```

```
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
    }
    policy-statement ted2nlri {
        term 1 {
            from protocol bgp-ls-epe;
            then accept;
        }
    }
    resolution-map map1 {
        mode ip-color;
    }
}
routing-options {
    static {
        route 10.100.100.101/32 {
            next-hop 10.100.100.100;
            resolve;
        }
    }
    router-id 10.7.7.7;
    autonomous-system 65300;
}
protocols {
    bgp {
        group ebgp1 {
            type external;
            family inet {
                unicast;
            }
            family traffic-engineering {
                unicast;
            }
            export nlri2bgp_epe;
            neighbor 192.168.12.1 {
                peer-as 65200;
                egress-te-adj-segment epe_adj1_toR3 {
                    label 8173;
                    next-hop 192.168.12.1;
                    te-link-attribute {
                        te-metric 20;
```

```
                    igp-metric 10;
                    admin-group [ red brown ];
                }
            }
        }
        neighbor 192.168.15.1 {
            peer-as 65200;
            egress-te-adj-segment epe_adj1_toR6 {
                label 8176;
                next-hop 192.168.15.1;
                te-link-attribute {
                    te-metric 20;
                    igp-metric 10;
                    admin-group [ red brown ];
                }
            }
        }
    }
}
mpls {
    traffic-engineering {
        database {
            import {
                l3-unicast-topology {
                    bgp-link-state;
                }
                policy ted2nlri;
            }
            export {
                policy nlri2ted_bgp;
                l3-unicast-topology;
            }
        }
    }
    admin-groups {
        red 0;
        blue 1;
        brown 5;
    }
    label-range {
        static-label-range 7000 70000;
    }
    interface all;
```

```
    }
    source-packet-routing {
        compute-profile compute1 {
            no-label-stack-compression;
        }
        source-routing-path computelsp1 {
            to 10.100.100.100;
            install 10.100.100.101;
            primary {
                p1 {
                    compute {
                        compute1;
                    }
                }
            }
        }
    }
}
```

Verification

IN THIS SECTION

- Verify the Express Segment | **1031**
- Verify the Express Segment Advertisements | **1034**
- Verify the TE Topology Information | **1040**

To confirm that the configuration is working properly, perform the following tasks:

Verify the Express Segment

Purpose

Verify that the express segments are created correctly.

Action

From operational mode, run the following commands:

- `show express-segments detail`—Verify whether the express segments are created.
- `show ted database topology-type express-segments detail`—Verify that the newly created express segments are inserted into the TE database.
- `show route table mpls.0 protocol express-segments`—Verify whether the forwarding entries have been created.

```
user@R1>show express-segments detail

Name: r1-exp-set1-10.6.6.6
  To: 10.6.6.6, Type: Dynamic (Set: r1-exp-set1)
  Label: 25 (Route installed in mpls.0, TED entry added)
  Status: Up (ElapsedTime: 09:32:00)
  LinkAttributes:
    LocalID: 2147483686
    TE-Metric: 200*, IGP-Metric: 100*
    BW: 0bps
    AdminGroups: red*
  UnderlayPaths: 1
    RSVP LSP: lsp1to6_a
      TE-Metric: 29, IGP-Metric: 20
      BW: 0bps
      AdminGroups: brown red

Name: r1-exp-set2-10.3.3.3
  To: 10.3.3.3, Type: Dynamic (Set: r1-exp-set2)
  Label: 24 (Route installed in mpls.0, TED entry added)
  Status: Up (ElapsedTime: 09:32:00)
  LinkAttributes:
    LocalID: 2147483685
    TE-Metric: 19, IGP-Metric: 20
    BW: 0bps
    AdminGroups: brown red
  UnderlayPaths: 1
    RSVP LSP: lsp1to3_a
      TE-Metric: 19, IGP-Metric: 20
      BW: 0bps
      AdminGroups: brown red
```

On R1

```
user@R1>show ted database topology-type express-segments detail

TED database: 0 ISIS nodes 4 INET nodes 0 INET6 nodes
NodeID: 10.1.1.1
  Type: Rtr, Age: 119174 secs, LinkIn: 0, LinkOut: 3
  Protocol: EXPRESS-SEG(0)
    To: 10.3.3.3, Local: 10.1.1.1, Remote: 10.3.3.3
      Local interface index: 2147483685, Remote interface index: 0
      Link name: r1-exp-set2-10.3.3.3
    To: 10.6.6.6, Local: 10.1.1.1, Remote: 10.6.6.6
      Local interface index: 2147483686, Remote interface index: 0
      Link name: r1-exp-set1-10.6.6.6
NodeID: 10.3.3.3
  Type: Rtr, Age: 34364 secs, LinkIn: 1, LinkOut: 0
  Protocol: EXPRESS-SEG(0)
NodeID: 10.6.6.6
  Type: Rtr, Age: 34364 secs, LinkIn: 1, LinkOut: 0
  Protocol: EXPRESS-SEG(0)
```

On R1

```
user@R1>show route table mpls.0 protocol express-segments

mpls.0: 13 destinations, 13 routes (13 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

24                 *[EXPRESS-SEG/6] 09:33:24, metric 1
                    >  to 192.168.3.2 via ge-0/0/2.0, Swap 33
25                 *[EXPRESS-SEG/6] 09:33:24, metric 1
                    >  to 192.168.3.2 via ge-0/0/2.0, Swap 34
```

Meaning

- In the `show express-segments detail` output, you can see the name of the express segments (**r1-exp-set1-10.6.6.6, r1-exp-set2-10.3.3.3**), express segment labels (**25, 24**), and the underlay LSPs (**lsp1to6_a, lsp1to3_a**).

- In the `show ted database topology-type express-segments detail` output, you can see the express segment entries are inserted into the TE database. The express segments (virtual TE links) are dynamically created. The protocol used is **EXPRESS-SEG(0)**.
- In the `show route table mpls.0 protocol express-segments` output, you can see the express segment labels (**24,25**). Because the express segment is a construct that relies on the underlay LSPs, the express segment label gets swapped to the underlay LSP labels (**33,34**), which is RSVP-LSP.

Verify the Express Segment Advertisements

Purpose

Verify that the originating node advertises express segments to its eBGP/iBGP LS neighbors.

Action

From operational mode, run the following commands:

- `show route table lsdist.0`—Verify that the express segments in the RIB BGP-LS are being advertised.
- `show route advertising-protocol bgp` *neighbor*—Verify that the express segments are sent to the eBGP/iBGP LS neighbors.

```
user@R1>show route table lsdist.0

lsdist.0: 25 destinations, 37 routes (25 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

NODE { AS:65200 IPv4:10.1.1.1 STATIC:0 }/1216
                   *[EXPRESS-SEG/6] 09:34:14
                       Fictitious
NODE { AS:65200 IPv4:10.3.3.3 STATIC:0 }/1216
                   *[EXPRESS-SEG/6] 09:34:14
                       Fictitious
NODE { AS:65200 IPv4:10.6.6.6 STATIC:0 }/1216
                   *[EXPRESS-SEG/6] 09:34:14
                       Fictitious
NODE { AS:65100 IPv4:10.1.1.1 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 1d 09:55:46, localpref 100
                      AS path: 65100 I, validation-state: unverified
                    >  to 192.168.1.1 via ge-0/0/0.0
NODE { AS:65100 IPv4:10.4.4.4 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 1d 09:55:46, localpref 100
```

```
                      AS path: 65100 I, validation-state: unverified
                    >  to 192.168.1.1 via ge-0/0/0.0
NODE { AS:65100 IPv4:10.100.100.100 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 1d 09:55:46, localpref 100
                      AS path: 65100 I, validation-state: unverified
                    >  to 192.168.1.1 via ge-0/0/0.0
NODE { AS:65200 IPv4:10.1.1.1 BGP-LS-EPE:0 }/1216
                   *[BGP-LS-EPE/170] 09:34:17
                       Fictitious
NODE { AS:65200 IPv4:10.3.3.3 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 1d 04:36:26, localpref 65100, from 10.2.2.2
                      AS path: I, validation-state: unverified
                    >  to 192.168.3.2 via ge-0/0/2.0
                       to 192.168.5.2 via ge-0/0/4.0
                    [BGP/170] 1d 04:36:26, localpref 100, from 10.5.5.5
                      AS path: I, validation-state: unverified
                    >  to 192.168.3.2 via ge-0/0/2.0
                       to 192.168.5.2 via ge-0/0/4.0
NODE { AS:65200 IPv4:10.4.4.4 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 1d 09:55:46, localpref 100, from 10.2.2.2
                      AS path: I, validation-state: unverified
                    >  to 192.168.4.2 via ge-0/0/3.0
                    [BGP/170] 1d 09:55:46, localpref 65100, from 10.5.5.5
                      AS path: I, validation-state: unverified
                    >  to 192.168.4.2 via ge-0/0/3.0
NODE { AS:65200 IPv4:10.6.6.6 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 1d 04:36:26, localpref 65100, from 10.2.2.2
                      AS path: I, validation-state: unverified
                    >  to 192.168.3.2 via ge-0/0/2.0
                       to 192.168.5.2 via ge-0/0/4.0
                    [BGP/170] 1d 04:36:26, localpref 100, from 10.5.5.5
                      AS path: I, validation-state: unverified
                    >  to 192.168.3.2 via ge-0/0/2.0
                       to 192.168.5.2 via ge-0/0/4.0
NODE { AS:65200 IPv4:10.7.7.7 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 1d 04:36:26, localpref 100, from 10.2.2.2
                      AS path: I, validation-state: unverified
                    >  to 192.168.3.2 via ge-0/0/2.0
                       to 192.168.5.2 via ge-0/0/4.0
                    [BGP/170] 1d 04:36:26, localpref 100, from 10.5.5.5
                      AS path: I, validation-state: unverified
                    >  to 192.168.3.2 via ge-0/0/2.0
                       to 192.168.5.2 via ge-0/0/4.0
```

```
NODE { AS:65200 IPv4:10.100.100.100 BGP-LS-EPE:0 }/1216
                   *[BGP-LS-EPE/170] 09:34:17
                       Fictitious
NODE { AS:65300 IPv4:3.3.3.3 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 1d 04:36:26, localpref 100, from 10.2.2.2
                      AS path: 65300 I, validation-state: unverified
                    >  to 10.49.127.254 via fxp0.0
                    [BGP/170] 1d 04:36:26, localpref 100, from 5.5.5.5
                      AS path: 300 I, validation-state: unverified
                    >  to 10.49.127.254 via fxp0.0
NODE { AS:65300 IPv4:10.6.6.6 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 1d 04:36:26, localpref 100, from 10.2.2.2
                      AS path: 65300 I, validation-state: unverified
                    >  to 10.49.127.254 via fxp0.0
                    [BGP/170] 1d 04:36:26, localpref 100, from 10.5.5.5
                      AS path: 65300 I, validation-state: unverified
                    >  to 10.49.127.254 via fxp0.0
NODE { AS:65300 IPv4:10.7.7.7 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 1d 04:36:26, localpref 65100, from 10.2.2.2
                      AS path: 65300 I, validation-state: unverified
                    >  to 10.49.127.254 via fxp0.0
                    [BGP/170] 1d 04:36:26, localpref 65100, from 10.5.5.5
                      AS path: 65300 I, validation-state: unverified
                    >  to 10.49.127.254 via fxp0.0
LINK { Local { AS:65200 IPv4:10.1.1.1 }.{ IfIndex:2147483685 } Remote { AS:65200 IPv4:10.3.3.3 }.
{ IfIndex:0 } STATIC:0 }/1216
                   *[EXPRESS-SEG/6] 09:34:14
                       Fictitious
LINK { Local { AS:65200 IPv4:10.1.1.1 }.{ IfIndex:2147483686 } Remote { AS:65200 IPv4:10.6.6.6 }.
{ IfIndex:0 } STATIC:0 }/1216
                   *[EXPRESS-SEG/6] 09:34:14
                       Fictitious
LINK { Local { AS:65100 IPv4:10.100.100.100 }.{ IfIndex:333 } Remote { AS:65200 IPv4:10.1.1.1 }.
{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                   *[BGP/170] 1d 09:55:46, localpref 100
                      AS path: 65100 I, validation-state: unverified
                    >  to 192.168.1.1 via ge-0/0/0.0
LINK { Local { AS:65100 IPv4:10.100.100.100 }.{ IfIndex:359 } Remote { AS:65200 IPv4:10.4.4.4 }.
{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                   *[BGP/170] 1d 09:55:46, localpref 100
                      AS path: 65100 I, validation-state: unverified
                    >  to 192.168.1.1 via ge-0/0/0.0
LINK { Local { AS:65200 IPv4:10.1.1.1 }.{ IfIndex:333 } Remote { AS:65100 IPv4:10.100.100.100 }.
```

```
{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                    *[BGP-LS-EPE/170] 09:34:17
                        Fictitious
LINK { Local { AS:65200 IPv4:10.3.3.3 }.{ IfIndex:362 } Remote { AS:65300 IPv4:10.7.7.7 }.
{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                    *[BGP/170] 1d 04:36:26, localpref 100, from 10.2.2.2
                       AS path: I, validation-state: unverified
                     >  to 192.168.3.2 via ge-0/0/2.0
                        to 192.168.5.2 via ge-0/0/4.0
                     [BGP/170] 1d 04:36:26, localpref 100, from 10.5.5.5
                       AS path: I, validation-state: unverified
                     >  to 192.168.3.2 via ge-0/0/2.0
                        to 192.168.5.2 via ge-0/0/4.0
LINK { Local { AS:65200 IPv4:10.4.4.4 }.{ IfIndex:333 } Remote { AS:65100 IPv4:10.100.100.100 }.
{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                    *[BGP/170] 1d 09:55:46, localpref 100, from 10.2.2.2
                       AS path: I, validation-state: unverified
                     >  to 192.168.4.2 via ge-0/0/3.0
                     [BGP/170] 1d 09:55:46, localpref 100, from 10.5.5.5
                       AS path: I, validation-state: unverified
                     >  to 192.168.4.2 via ge-0/0/3.0
LINK { Local { AS:65200 IPv4:10.6.6.6 }.{ IfIndex:361 } Remote { AS:65300 IPv4:10.7.7.7 }.
{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                    *[BGP/170] 1d 04:36:26, localpref 100, from 10.2.2.2
                       AS path: I, validation-state: unverified
                     >  to 192.168.3.2 via ge-0/0/2.0
                        to 192.168.5.2 via ge-0/0/4.0
                     [BGP/170] 1d 04:36:26, localpref 100, from 10.5.5.5
                       AS path: I, validation-state: unverified
                     >  to 192.168.3.2 via ge-0/0/2.0
                        to 192.168.5.2 via ge-0/0/4.0
LINK { Local { AS:65300 IPv4:10.7.7.7 }.{ IfIndex:334 } Remote { AS:65200 IPv4:10.3.3.3 }.
{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                    *[BGP/170] 1d 04:36:26, localpref 100, from 10.2.2.2
                       AS path: 300 I, validation-state: unverified
                     >  to 10.49.127.254 via fxp0.0
                     [BGP/170] 1d 04:36:26, localpref 100, from 10.5.5.5
                       AS path: 65300 I, validation-state: unverified
                     >  to 10.49.127.254 via fxp0.0
LINK { Local { AS:65300 IPv4:10.7.7.7 }.{ IfIndex:359 } Remote { AS:65200 IPv4:10.6.6.6 }.
{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                    *[BGP/170] 1d 04:36:26, localpref 65100, from 10.2.2.2
                       AS path: 65300 I, validation-state: unverified
```

```
                    >  to 10.49.127.254 via fxp0.0
                    [BGP/170] 1d 04:36:26, localpref 100, from 10.5.5.5
                      AS path: 65300 I, validation-state: unverified
                    >  to 10.49.127.254 via fxp0.0
```

On R1

```
user@R1>show route advertising-protocol bgp 10.2.2.2

lsdist.0: 25 destinations, 37 routes (25 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
  NODE { AS:65100 IPv4:10.1.1.1 BGP-LS-EPE:0 }/1216
*                         192.168.1.1                  100        65100 I
                Area border router: No
                External router: No
                Attached: No
                Overload: No
  Prefix                  Nexthop              MED     Lclpref    AS path
  NODE { AS:65100 IPv4:10.4.4.4 BGP-LS-EPE:0 }/1216
*                         192.168.1.1                  100        65100 I
                Area border router: No
                External router: No
                Attached: No
                Overload: No
  Prefix                  Nexthop              MED     Lclpref    AS path
  NODE { AS:65100 IPv4:10.100.100.100 BGP-LS-EPE:0 }/1216
*                         192.168.1.1                  100        65100 I
                Area border router: No
                External router: No
                Attached: No
                Overload: No
  Prefix                  Nexthop              MED     Lclpref    AS path
  NODE { AS:65200 IPv4:10.1.1.1 BGP-LS-EPE:0 }/1216
*                         Self                         100        I
                Area border router: No
                External router: No
                Attached: No
                Overload: No
  Prefix                  Nexthop              MED     Lclpref    AS path
  NODE { AS:65200 IPv4:10.100.100.100 BGP-LS-EPE:0 }/1216
*                         Self                         100        I
```

```
                Area border router: No
                External router: No
                Attached: No
                Overload: No
  Prefix                  Nexthop              MED     Lclpref    AS path
  LINK { Local { AS:65100 IPv4:10.100.100.100 }.{ IfIndex:333 } Remote { AS:65200
IPv4:10.1.1.1 }.{ IfIndex:0 } BGP-LS-EPE:0 }/1216
*                         192.168.1.1                  100        65100 I
                Color: 33
                Metric: 10
                TE Metric: 20
                Link name: epe_adj1_toR1
                Label: 7101, Flags: 0xd0, Weight: 0
  Prefix                  Nexthop              MED     Lclpref    AS path
  LINK { Local { AS:65100 IPv4:10.100.100.100 }.{ IfIndex:359 } Remote { AS:65200
IPv4:10.4.4.4 }.{ IfIndex:0 } BGP-LS-EPE:0 }/1216
*                         192.168.1.1                  100        65100 I
                Color: 33
                Metric: 10
                TE Metric: 20
                Link name: epe_adj1_toR4
                Label: 7104, Flags: 0xd0, Weight: 0
  Prefix                  Nexthop              MED     Lclpref    AS path
  LINK { Local { AS:65200 IPv4:10.1.1.1 }.{ IfIndex:333 } Remote { AS:65100
IPv4:10.100.100.100 }.{ IfIndex:0 } BGP-LS-EPE:0 }/1216
*                         Self                         100        I
                Color: 33
                Metric: 10
                TE Metric: 20
                Link name: epe_adj1_toR0
                Label: 8110, Flags: 0xd0, Weight: 0
```

Meaning

- In the `show route table lsdist.0` output, BGP advertises the routes in the routing table. The routing table is created from the TE database. You can see the express segments (**EXPRESS-SEG/6**) links and the EPE links (**BGP-LS-EPE:0 }/1216**).
- In the `show route advertising-protocol bgp 10.2.2.2` output, you can see what R1 is advertising to. The express segments are inserted into the TE database, which is copied to RIB. BGP-LS advertises the RIB to the peer router. On the peer, the received RIB information is copied into the local database. The policy in this example only advertises express segments and EPE segments.

Verify the TE Topology Information

Purpose

Verify that the ingress nodes receive TE topology information through eBGP/iBGP LS.

Action

From operational mode, run the following commands:

- `show route receive-protocol bgp` *neighbor*—Verify that the express segments are received from eBGP/iBGP LS neighbors.
- `show route table lsdist.0`—Verify that the express segments are in the BGP-LS RIB.
- `show ted database topology-type l3-unicast detail`—Verify that the express segments are imported into the ingress router's TE database.
- `show spring-traffic-engineering lsp`—Verify that the end-to-end SR policy has been successfully computed and installed.

On R0

```
user@R0>show route receive-protocol bgp 192.168.1.2
…
  Prefix                  Nexthop              MED     Lclpref    AS path
  NODE { AS:65200 IPv4:10.1.1.1 STATIC:0 }/1216
*                         192.168.1.2                             65200 I
                Area border router: No
                External router: No
                Attached: No
                Overload: No
  Prefix                  Nexthop              MED     Lclpref    AS path
  NODE { AS:65200 IPv4:10.3.3.3 STATIC:0 }/1216
*                         192.168.1.2                             65200 I
                Area border router: No
                External router: No
                Attached: No
                Overload: No
...
```

On R0

```
user@R0>show route table lsdist.0

lsdist.0: 28 destinations, 40 routes (28 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

NODE { AS:65200 IPv4:10.1.1.1 STATIC:0 }/1216
                   *[BGP/170] 09:37:43, localpref 100
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0
NODE { AS:65200 IPv4:10.3.3.3 STATIC:0 }/1216
                   *[BGP/170] 09:37:43, localpref 100
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0
                    [BGP/170] 09:35:57, localpref 100
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.2.2 via ge-0/0/2.0
NODE { AS:65200 IPv4:10.4.4.4 STATIC:0 }/1216
                   *[BGP/170] 09:35:57, localpref 100
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.2.2 via ge-0/0/2.0
NODE { AS:65200 IPv4:10.6.6.6 STATIC:0 }/1216
                   *[BGP/170] 09:37:43, localpref 100
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0
                    [BGP/170] 09:35:57, localpref 100
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.2.2 via ge-0/0/2.0
NODE { AS:65100 IPv4:10.1.1.1 BGP-LS-EPE:0 }/1216
                   *[BGP-LS-EPE/170] 1d 04:37:15
                       Fictitious
NODE { AS:65100 IPv4:10.4.4.4 BGP-LS-EPE:0 }/1216
                   *[BGP-LS-EPE/170] 1d 04:37:15
                       Fictitious
NODE { AS:65100 IPv4:10.100.100.100 BGP-LS-EPE:0 }/1216
                   *[BGP-LS-EPE/170] 1d 04:37:15
                       Fictitious
NODE { AS:65200 IPv4:10.1.1.1 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 1d 09:59:16, localpref 100
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.2.2 via ge-0/0/2.0
```

```
NODE { AS:65200 IPv4:10.3.3.3 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 1d 04:39:56, localpref 100
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.2.2 via ge-0/0/2.0
                    [BGP/170] 1d 04:39:56, localpref 100
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0
NODE { AS:65200 IPv4:10.4.4.4 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 1d 09:59:16, localpref 100
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0
NODE { AS:65200 IPv4:10.6.6.6 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 1d 04:39:56, localpref 100
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.2.2 via ge-0/0/2.0
                    [BGP/170] 1d 04:39:56, localpref 100
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0
NODE { AS:65200 IPv4:10.7.7.7 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 1d 04:39:56, localpref 100
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.2.2 via ge-0/0/2.0
                    [BGP/170] 1d 04:39:56, localpref 100
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0
NODE { AS:65200 IPv4:10.100.100.100 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 1d 09:59:16, localpref 100
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.2.2 via ge-0/0/2.0
NODE { AS:65300 IPv4:10.3.3.3 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 1d 04:39:56, localpref 100
                      AS path: 65200 65300 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0
                    [BGP/170] 1d 04:39:56, localpref 100
                      AS path: 65200 65300 I, validation-state: unverified
                    >  to 192.168.2.2 via ge-0/0/2.0
NODE { AS:65300 IPv4:10.6.6.6 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 1d 04:39:56, localpref 100
                      AS path: 65200 65300 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0
                    [BGP/170] 1d 04:39:56, localpref 100
                      AS path: 65200 65300 I, validation-state: unverified
                    >  to 192.168.2.2 via ge-0/0/2.0
```

```
NODE { AS:65300 IPv4:10.7.7.7 BGP-LS-EPE:0 }/1216
                    *[BGP/170] 1d 04:39:56, localpref 100
                       AS path: 65200 65300 I, validation-state: unverified
                     >  to 192.168.1.2 via ge-0/0/0.0
                     [BGP/170] 1d 04:39:56, localpref 100
                       AS path: 65200 65300 I, validation-state: unverified
                     >  to 192.168.2.2 via ge-0/0/2.0
LINK { Local { AS:65200 IPv4:10.1.1.1 }.{ IfIndex:2147483685 } Remote { AS:65200 IPv4:10.3.3.3 }.
{ IfIndex:0 } STATIC:0 }/1216
                    *[BGP/170] 09:37:43, localpref 100
                       AS path: 65200 I, validation-state: unverified
                     >  to 192.168.1.2 via ge-0/0/0.0
LINK { Local { AS:65200 IPv4:10.1.1.1 }.{ IfIndex:2147483686 } Remote { AS:65200 IPv4:10.6.6.6 }.
{ IfIndex:0 } STATIC:0 }/1216
                    *[BGP/170] 09:37:43, localpref 100
                       AS path: 54200 I, validation-state: unverified
                     >  to 192.168.1.2 via ge-0/0/0.0
LINK { Local { AS:65200 IPv4:10.4.4.4 }.{ IfIndex:2147483684 } Remote { AS:65200 IPv4:10.3.3.3 }.
{ IfIndex:0 } STATIC:0 }/1216
                    *[BGP/170] 09:35:57, localpref 100
                       AS path: 65200 I, validation-state: unverified
                     >  to 192.168.2.2 via ge-0/0/2.0
LINK { Local { AS:65200 IPv4:10.4.4.4 }.{ IfIndex:2147483685 } Remote { AS:65200 IPv4:10.6.6.6 }.
{ IfIndex:0 } STATIC:0 }/1216
                    *[BGP/170] 09:35:57, localpref 100
                       AS path: 65200 I, validation-state: unverified
                     >  to 192.168.2.2 via ge-0/0/2.0
LINK { Local { AS:65100 IPv4:10.100.100.100 }.{ IfIndex:333 } Remote { AS:65200 IPv4:10.1.1.1 }.
{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                    *[BGP-LS-EPE/170] 1d 04:37:15
                        Fictitious
LINK { Local { AS:65100 IPv4:10.100.100.100 }.{ IfIndex:359 } Remote { AS:65200 IPv4:10.4.4.4 }.
{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                    *[BGP-LS-EPE/170] 1d 04:37:15
                        Fictitious
LINK { Local { AS:65200 IPv4:10.1.1.1 }.{ IfIndex:333 } Remote { AS:65100 IPv4:10.100.100.100 }.
{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                    *[BGP/170] 1d 09:59:16, localpref 100
                       AS path: 65200 I, validation-state: unverified
                     >  to 192.168.2.2 via ge-0/0/2.0
LINK { Local { AS:65200 IPv4:10.3.3.3 }.{ IfIndex:362 } Remote { AS:65300 IPv4:10.7.7.7 }.
{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                    *[BGP/170] 1d 04:39:56, localpref 100
```

```
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.2.2 via ge-0/0/2.0
                    [BGP/170] 1d 04:39:56, localpref 100
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0
LINK { Local { AS:65200 IPv4:10.4.4.4 }.{ IfIndex:333 } Remote { AS:65100 IPv4:10.100.100.100 }.
{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                   *[BGP/170] 1d 09:59:16, localpref 100
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0
LINK { Local { AS:65200 IPv4:10.6.6.6 }.{ IfIndex:361 } Remote { AS:65300 IPv4:10.7.7.7 }.
{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                   *[BGP/170] 1d 04:39:56, localpref 100
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.2.2 via ge-0/0/2.0
                    [BGP/170] 1d 04:39:56, localpref 100
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0
LINK { Local { AS:65300 IPv4:10.7.7.7 }.{ IfIndex:334 } Remote { AS:65200 IPv4:10.3.3.3 }.
{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                   *[BGP/170] 1d 04:39:56, localpref 100
                      AS path: 65200 65300 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0
                    [BGP/170] 1d 04:39:56, localpref 100
                      AS path: 65200 65300 I, validation-state: unverified
                    >  to 192.168.2.2 via ge-0/0/2.0
LINK { Local { AS:65300 IPv4:10.7.7.7 }.{ IfIndex:359 } Remote { AS:65200 IPv4:10.6.6.6 }.
{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                   *[BGP/170] 1d 04:39:56, localpref 100
                      AS path: 65200 65300 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0
                    [BGP/170] 1d 04:39:56, localpref 100
                      AS path: 65200 65300 I, validation-state: unverified
                    >  to 192.168.2.2 via ge-0/0/2.0
```

On R0

```
user@R0>show ted database topology-type l3-unicast detail

TED database: 0 ISIS nodes 6 INET nodes 0 INET6 nodes
NodeID: 10.1.1.1
  Type: Rtr, Age: 122418 secs, LinkIn: 1, LinkOut: 3
```

```
  Protocol: Exported BGP(6)
    To: 10.100.100.100, Local: 0.0.0.0, Remote: 0.0.0.0
      Local interface index: 333, Remote interface index: 0
      Link name: epe_adj1_toR0
  Protocol: Exported STATIC(4)
    To: 10.6.6.6, Local: 0.0.0.0, Remote: 0.0.0.0
      Local interface index: 2147483686, Remote interface index: 0
      Link name: r1-exp-set1-10.6.6.6
    To: 10.3.3.3, Local: 0.0.0.0, Remote: 0.0.0.0
      Local interface index: 2147483685, Remote interface index: 0
      Link name: r1-exp-set2-10.3.3.3
  Protocol: BGP-LS-EPE(0)
NodeID: 10.3.3.3
  Type: Rtr, Age: 122418 secs, LinkIn: 3, LinkOut: 1
  Protocol: Exported BGP(6)
    To: 10.7.7.7, Local: 0.0.0.0, Remote: 0.0.0.0
      Local interface index: 362, Remote interface index: 0
      Link name: epe_adj1_toR7
  Protocol: Exported BGP(8)
  Protocol: Exported STATIC(4)
NodeID: 10.4.4.4
  Type: Rtr, Age: 122418 secs, LinkIn: 1, LinkOut: 3
  Protocol: Exported BGP(6)
    To: 10.100.100.100, Local: 0.0.0.0, Remote: 0.0.0.0
      Local interface index: 333, Remote interface index: 0
      Link name: epe_adj1_toR0
  Protocol: Exported STATIC(4)
    To: 10.6.6.6, Local: 0.0.0.0, Remote: 0.0.0.0
      Local interface index: 2147483685, Remote interface index: 0
      Link name: r4-exp-set1-10.6.6.6
    To: 10.3.3.3, Local: 0.0.0.0, Remote: 0.0.0.0
      Local interface index: 2147483684, Remote interface index: 0
      Link name: r4-exp-set2-10.3.3.3
  Protocol: BGP-LS-EPE(0)
NodeID: 10.6.6.6
  Type: Rtr, Age: 122418 secs, LinkIn: 3, LinkOut: 1
  Protocol: Exported BGP(6)
    To: 10.7.7.7, Local: 0.0.0.0, Remote: 0.0.0.0
      Local interface index: 361, Remote interface index: 0
      Link name: epe_adj1_toR7
  Protocol: Exported BGP(8)
  Protocol: Exported STATIC(4)
NodeID: 10.7.7.7
```

```
  Type: Rtr, Age: 103258 secs, LinkIn: 2, LinkOut: 2
  Protocol: Exported BGP(6)
  Protocol: Exported BGP(8)
    To: 10.6.6.6, Local: 0.0.0.0, Remote: 0.0.0.0
      Local interface index: 359, Remote interface index: 0
      Link name: epe_adj1_toR6
    To: 10.3.3.3, Local: 0.0.0.0, Remote: 0.0.0.0
      Local interface index: 334, Remote interface index: 0
      Link name: epe_adj1_toR3
NodeID: 10.100.100.100
  Type: Rtr, Age: 103160 secs, LinkIn: 2, LinkOut: 2
  Protocol: Exported BGP(6)
  Protocol: BGP-LS-EPE(0)
    To: 10.1.1.1, Local: 192.168.1.1, Remote: 192.168.1.2
      Local interface index: 333, Remote interface index: 0
      Link name: epe_adj1_toR1
      Local bgp peer as: 100, Remote bgp peer as: 200
    To: 10.4.4.4, Local: 192.168.2.1, Remote: 192.168.2.2
      Local interface index: 359, Remote interface index: 0
      Link name: epe_adj1_toR4
      Local bgp peer as: 65100, Remote bgp peer as: 65200
```

On R0

```
user@R0>show spring-traffic-engineering lsp

To              State     LSPname
10.7.7.7         Up        computelsp1
10.7.7.7-7000<c> Up        ecomputelsp1
10.7.7.7-7001<c> Up        ecomputelsp2


Total displayed LSPs: 3 (Up: 3, Down: 0)
```

On R0

```
user@R0>show spring-traffic-engineering lsp detail

Name: computelsp1
  Tunnel-source: Static configuration
  To: 10.7.7.7
```

```
State: Up
  Path: p1
  Outgoing interface: NA
  Auto-translate status: Disabled Auto-translate result: N/A
  Compute Status:Enabled , Compute Result:success , Compute-Profile Name:compute1
  Total number of computed paths: 2
  Computed-path-index: 1
    BFD status: N/A BFD name: N/A
    TE metric: 59, IGP metric: 40; Metric optimized by type: TE
    computed segments count: 3
      computed segment : 1 (computed-adjacency-segment):
        label: 7104
        source router-id: 10.100.100.100, destination router-id: 10.4.4.4
        source interface-address: 192.168.2.1, destination interface-address: 192.168.2.2
      computed segment : 2 (computed-adjacency-segment):
        label: 21
        source router-id: 10.4.4.4, destination router-id: 10.6.6.6
        source interface-address: 0.0.0.0, destination interface-address: 0.0.0.0
      computed segment : 3 (computed-adjacency-segment):
        label: 7167
        source router-id: 10.6.6.6, destination router-id: 10.7.7.7
        source interface-address: 0.0.0.0, destination interface-address: 0.0.0.0
  Computed-path-index: 2
    BFD status: N/A BFD name: N/A
    TE metric: 59, IGP metric: 40; Metric optimized by type: TE
    computed segments count: 3
      computed segment : 1 (computed-adjacency-segment):
        label: 7101
        source router-id: 10.100.100.100, destination router-id: 10.1.1.1
        source interface-address: 192.168.1.1, destination interface-address: 192.168.1.2
      computed segment : 2 (computed-adjacency-segment):
        label: 24
        source router-id: 10.1.1.1, destination router-id: 10.3.3.3
        source interface-address: 0.0.0.0, destination interface-address: 0.0.0.0
      computed segment : 3 (computed-adjacency-segment):
        label: 7137
        source router-id: 10.3.3.3, destination router-id: 10.7.7.7
        source interface-address: 0.0.0.0, destination interface-address: 0.0.0.0
```

Meaning

- In the `show route receive-protocol bgp 10.1.1.1` output, it shows the routes that have been received by the ingress router (R0) from the BGP neighbor, which describes the express segment (virtual TE links).
- In the `show route table lsdist.0` output, it shows the routes that have been received by the ingress router (R0) and whether they are inserted into the **lsdist.0** RIB. It also shows whether the **lsdist.0** RIB is copied into the local TE database.
- In the `show ted database topology-type l3-unicast detail` output, the routes are copied into the local TE database. The **r1-exp-set1-10.6.6.6** is an express segment with end point as 10.6.6.6 and is sucessfully created on R1. R1 has advertised the express segment and R0 has inserted it into the local TE database. You can also see the EPE segments (**epe_adj1_toR7**).
- In the `show spring-traffic-engineering lsp` output, you can see that the SR policies are up. It shows that you are now able to compute a multi-domain end-to-end (R0 to R7) SR policy.
- In the `show spring-traffic-engineering lsp detail` output, you can see the labels that are selected. In the **computelsp1** LSP, the label **7104** is an EPE segment, **21** is the express segment, and **7167** is also an EPE segment. It shows that you are now able to compute a multi-domain end-to-end (R0 to R7) SR policy.

Example: Inter-domain SR-TE Connectivity Using Express Segments Through SR-TE Underlay

IN THIS SECTION

- Requirements | **1048**
- Overview | **1049**
- Configuration | **1053**
- Verification | **1199**

Use this example to learn how to establish an end-to-end inter-domain SR-TE connectivity using express segments through SR-TE underlay.

Requirements

This example uses the following hardware and software components:

- MX Series routers as provider edge, border nodes, and intermediate routers.

- Junos OS Release 21.2R1 or later releases.

Overview

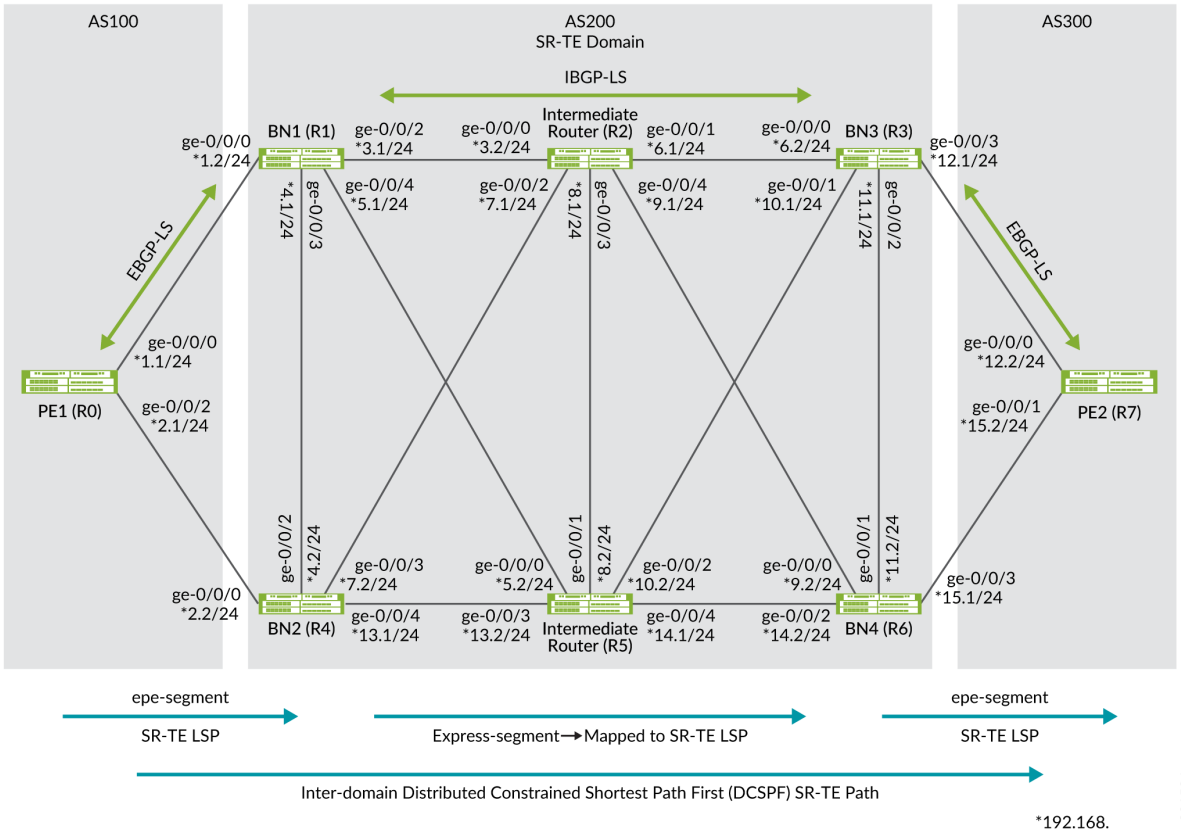
IN THIS SECTION

- Topology | **1049**

The following topology (Figure 67 on page 1049) shows two SR-TE domains (AS100 and AS300) running EBGP-LS inter-connected through another SR-TE (AS200) domain:

Topology

Figure 67: Inter-domain SR-TE Connectivity Using Express Segments Through SR-TE Underlay



In this topology, an end-to-end SR-TE path between PE1 router to PE2 router is established. Egress peer engineering (EPE) segments are defined on PE1 and PE2 routers to steer traffic towards their directly connected border nodes BN1/BN2 and BN3/BN4, respectively. EPE segments defined on the border nodes are advertised internally through the BGP link state. These two SR-TE domains are interconnected through the domain (AS200) that is leveraging SR-TE LSPs for internal path establishment.

The border nodes of the AS200 domain facilitate the abstraction of SR-TE information between domains. Express segments are created on border nodes (BN1, BN2, BN3, and BN4). Express segments are created in a one-on-one relationship with the underlying SR-TE LSPs and all express segments are inserted into the border node's local TE database for subsequent BGP link-state advertisement. The AS200 domain leverages SR-TE LSP underlays for TE management and presents those underlay SR-TE LSPs as express segments to the AS100 and the AS300 domains, enabling the domains to have end-to-end SR-TE LSP connectivity.

The following table describes the domains, routers, and connections in the topology:

Table 30: Describes the domains, routers, and connections in the Topology

Domain	Devices	Router ID/Lo) Address	Connection Details
AS65100 (EBGP-LS/ SR-TE LSP)	R0 (PE1 router)	10.100.100.100 10.100.100.101	Connected to R1 (BN1 router) through interface ge-0/0/0, assigned IP address 192.168.1.1/24. Connected to R4 (BN2 router) through interface ge-0/0/2, assigned IP address 192.168.2.1/24.
AS65200 (SR-TE LSP)	R1 (BN1 router)	10.1.1.1	Connected to R0 (PE1 router) through interface ge-0/0/0, assigned IP address 192.168.1.2/24. Connected to R4 (BN2 router) through interface ge-0/0/3, assigned IP address 192.168.4.1/24. Connected to R2 (Intermediate router) through interface ge-0/0/2, assigned IP address 192.168.3.1/24. Connected to R5 (Intermediate router) through interface ge-0/0/4, assigned IP address 192.168.5.1/24.

Table 30: Describes the domains, routers, and connections in the Topology *(Continued)*

Domain	Devices	Router ID/Lo) Address	Connection Details
	R4(BN2 router)	10.4.4.4	Connected to R0 (PE1 router) through interface ge-0/0/0, assigned IP address 192.168.2.2/24. Connected to R1 (BN1 router) through interface ge-0/0/2, assigned IP address 192.168.4.2/24. Connected to R2 (Intermediate router) through interface ge-0/0/3, assigned IP address 192.168.7.1/24. Connected to R5 (Intermediate router) through interface ge-0/0/4, assigned IP address 192.168.13.1/24.
	R2(Intermediate router)	10.2.2.2	Connected to R1 (BN1 router) through interface ge-0/0/0, assigned IP address 192.168.3.2/24. Connected to R4 (BN2 router) through interface ge-0/0/2, assigned IP address 192.168.7.1/24. Connected to R5 (Intermediate router) through interface ge-0/0/3, assigned IP address 192.168.8.1/24. Connected to R3 (BN3 router) through interface ge-0/0/1, assigned IP address 192.168.6.1/24. Connected to R6 (BN4 router) through interface ge-0/0/4, assigned IP address 192.168.9.1/24.

Table 30: Describes the domains, routers, and connections in the Topology *(Continued)*

Domain	Devices	Router ID/Lo) Address	Connection Details
	R5 (Intermediate router)	10.5.5.5	Connected to R1 (BN1 router) through interface ge-0/0/0, assigned IP address 192.168.5.2/24. Connected to R4 (BN2 router) through interface ge-0/0/3, assigned IP address 192.168.13.2/24. Connected to R2 (Intermediate router) through interface ge-0/0/1, assigned IP address 192.168.8.2/24. Connected to R3 (BN3 router) through interface ge-0/0/2, assigned IP address 192.168.10.2/24. Connected to R6 (BN4 router) through interface ge-0/0/4, assigned IP address 192.168.14.1/24.
	R3 (BN3 router)	10.3.3.3	Connected to R7 (PE2 router) through interface ge-0/0/3, assigned IP address 192.168.12.1/24. Connected to R6 (BN4 router) through interface ge-0/0/2, assigned IP address 192.168.11.1/24. Connected to R2 (Intermediate router) through interface ge-0/0/0, assigned IP address 192.168.6.2/24. Connected to R5 (Intermediate router) through interface ge-0/0/1, assigned IP address 192.168.10.1/24.

Table 30: Describes the domains, routers, and connections in the Topology *(Continued)*

Domain	Devices	Router ID/Lo) Address	Connection Details
	R6 (BN4 router)	10.6.6.6	Connected to R7 (PE2 router) through interface ge-0/0/3, assigned IP address 192.168.15.1/24. Connected to R3 (BN3 router) through interface ge-0/0/1, assigned IP address 192.168.11.2/24. Connected to R2 (Intermediate router) through interface ge-0/0/0, assigned IP address 192.168.9.2/24. Connected to R5 (Intermediate router) through interface ge-0/0/2, assigned IP address 192.168.14.2/24.
AS65300 (EBGP-LS/SR-TE LSP)	R7 (PE2 router)	10.7.7.7	Connected to R3 (BN3 router) through interface ge-0/0/0, assigned IP address 192.168.12.2/24. Connected to R6 (BN4 router) through interface ge-0/0/1, assigned IP address 192.168.15.2/24.

Configuration

IN THIS SECTION

- CLI Quick Configuration | **1054**
- Configure R0 (PE1 router) | **1082**
- Configure R1 (BN1 router) | **1094**
- Configure R4 (BN2 router) | **1110**
- Configure R2 (Intermediate router) | **1125**
- Configure R5 (Intermediate router) | **1140**
- Configure R3 (BN3 router) | **1154**
- Configure R6 (BN4 router) | **1171**
- Configure R7 (PE2 router) | **1186**

To inter-connect a multi-domain network and establish an end-to-end SR path using express segments, perform these tasks:

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

The quick configuration commands provided below can be used to configure express segments through uncolored SR-TE underlay path.

To configure colored SR-TE underlay path, you must do additional configurations on BN1 (R1), BN2 (R4), BN3 (R3), and BN4 (R6) routers. Below are the uncolored configurations for N1 (R1), BN2 (R4), BN3 (R3), and BN4 (R6) routers, you can find the additional colored configuration.

Configure R0 (PE1 router)
Device R0 (PE1 router)

```
set interfaces ge-0/0/0 description To_R1_1
set interfaces ge-0/0/0 vlan-tagging
set interfaces ge-0/0/0 unit 0 vlan-id 1
set interfaces ge-0/0/0 unit 0 family inet address 192.168.1.1/24
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:1000:10::100/64
set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/2 description To_R4_1
set interfaces ge-0/0/2 vlan-tagging
set interfaces ge-0/0/2 unit 0 vlan-id 1
set interfaces ge-0/0/2 unit 0 family inet address 192.168.2.1/24
set interfaces ge-0/0/2 unit 0 family iso
set interfaces ge-0/0/2 unit 0 family inet6 address 2001:db8:4000:10::100/64
set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
set interfaces lo0 unit 0 family inet address 10.100.100.100/32
set interfaces lo0 unit 0 family inet address 10.100.100.101/32
set interfaces lo0 unit 0 family iso address 49.0001.000a.0a0a.0a00
set interfaces lo0 unit 0 family inet6 address 2001:db8:abcd::10:100:100:100/128
set policy-options policy-statement bgplsepe_rt_2_ted term 1 from protocol bgp
set policy-options policy-statement bgplsepe_rt_2_ted term 1 then accept
set policy-options policy-statement direct from protocol direct
set policy-options policy-statement direct then accept
set policy-options policy-statement mpath then multipath-resolve
set policy-options policy-statement nlri2bgp term 1 from family traffic-engineering
```

```
set policy-options policy-statement nlri2bgp term 1 then accept
set policy-options policy-statement nlri2bgp_epe term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp_epe term 1 from protocol bgp-ls-epe
set policy-options policy-statement nlri2bgp_epe term 1 then next-hop self
set policy-options policy-statement nlri2bgp_epe term 1 then accept
set policy-options policy-statement nlri2ted_bgp term 1 from protocol bgp
set policy-options policy-statement nlri2ted_bgp term 1 then accept
set policy-options policy-statement nlri2ted_igp term 1 from traffic-engineering protocol isis-
level-2
set policy-options policy-statement nlri2ted_igp term 1 then accept
set policy-options policy-statement pplb then load-balance per-packet
set policy-options policy-statement ted2nlri term 1 from protocol bgp-ls-epe
set policy-options policy-statement ted2nlri term 1 then accept
set policy-options policy-statement vpn1_res_map1 from route-filter 10.109.1/16 orlonger
set policy-options policy-statement vpn1_res_map1 then accept
set policy-options policy-statement vpn1_res_map1 then resolution-map map1
set policy-options policy-statement vpn2_res_map1 from route-filter 10.110.0.1/16 orlonger
set policy-options policy-statement vpn2_res_map1 then accept
set policy-options policy-statement vpn2_res_map1 then resolution-map map1
set policy-options community color7000 members color:0:7000
set policy-options community color7001 members color:0:7001
set policy-options resolution-map map1 mode ip-color
set routing-options router-id 10.100.100.100
set routing-options autonomous-system 100
set routing-options static route 10.7.7.71/32 next-hop 10.7.7.7
set routing-options static route 10.7.7.71/32 resolve
set protocols bgp group ebgp1 type external
set protocols bgp group ebgp1 multihop ttl 100
set protocols bgp group ebgp1 family inet unicast
set protocols bgp group ebgp1 family inet-vpn unicast
set protocols bgp group ebgp1 family traffic-engineering unicast
set protocols bgp group ebgp1 export nlri2bgp_epe
set protocols bgp group ebgp1 neighbor 192.168.1.2 peer-as 65200
set protocols bgp group ebgp1 neighbor 192.168.1.2 egress-te-adj-segment epe_adj1_toR1 label 7101
set protocols bgp group ebgp1 neighbor 192.168.1.2 egress-te-adj-segment epe_adj1_toR1 next-hop
192.168.1.2
set protocols bgp group ebgp1 neighbor 192.168.1.2 egress-te-adj-segment epe_adj1_toR1 te-link-
attribute te-metric 20
set protocols bgp group ebgp1 neighbor 192.168.1.2 egress-te-adj-segment epe_adj1_toR1 te-link-
attribute igp-metric 10
set protocols bgp group ebgp1 neighbor 192.168.1.2 egress-te-adj-segment epe_adj1_toR1 te-link-
attribute admin-group [ red brown ]
set protocols bgp group ebgp1 neighbor 192.168.2.2 peer-as 65200
```

```
set protocols bgp group ebgp1 neighbor 192.168.2.2 egress-te-adj-segment epe_adj1_toR4 label 7104
set protocols bgp group ebgp1 neighbor 192.168.2.2 egress-te-adj-segment epe_adj1_toR4 next-hop
192.168.12.1192.168.2.2
set protocols bgp group ebgp1 neighbor 192.168.2.2 egress-te-adj-segment epe_adj1_toR4 te-link-
attribute te-metric 20
set protocols bgp group ebgp1 neighbor 192.168.2.2 egress-te-adj-segment epe_adj1_toR4 te-link-
attribute igp-metric 10
set protocols bgp group ebgp1 neighbor 192.168.2.2 egress-te-adj-segment epe_adj1_toR4 te-link-
attribute admin-group [ red brown ]
set protocols bgp group ebgp1 neighbor 10.7.7.71 local-address 00.100.100.101
set protocols bgp group ebgp1 neighbor 10.7.7.71 import [ vpn1_res_map1 vpn2_res_map1 ]
set protocols bgp group ebgp1 neighbor 10.7.7.71 peer-as 65300
set protocols bgp group ebgp1 vpn-apply-export
set protocols isis interface lo0.0 passive
set protocols isis level 1 disable
set protocols isis level 2 wide-metrics-only
set protocols isis traffic-engineering l3-unicast-topology
set protocols isis traffic-engineering advertisement always
set protocols mpls traffic-engineering database import l3-unicast-topology bgp-link-state
set protocols mpls traffic-engineering database import policy ted2nlri
set protocols mpls traffic-engineering database export policy nlri2ted_bgp
set protocols mpls traffic-engineering database export l3-unicast-topology
set protocols mpls admin-groups red 0
set protocols mpls admin-groups blue 1
set protocols mpls admin-groups green 2
set protocols mpls admin-groups yellow 3
set protocols mpls admin-groups orange 4
set protocols mpls admin-groups brown 5
set protocols mpls admin-groups black 6
set protocols mpls admin-groups pink 7
set protocols mpls label-range static-label-range 1000 70000
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols mpls interface ge-0/0/0.0 admin-group [ red brown ]
set protocols source-packet-routing compute-profile compute1 no-label-stack-compression
set protocols source-packet-routing compute-profile ecompute1 no-label-stack-compression
set protocols source-packet-routing source-routing-path computelsp1 to 10.7.7.7
set protocols source-packet-routing source-routing-path computelsp1 install 10.7.7.71
set protocols source-packet-routing source-routing-path computelsp1 primary p1 compute compute1
set protocols source-packet-routing source-routing-path ecomputelsp1 to 10.7.7.7
set protocols source-packet-routing source-routing-path ecomputelsp1 color 7000
set protocols source-packet-routing source-routing-path ecomputelsp1 primary p1 compute ecompute1
```

Device R1 (BN1 router)

```
set interfaces ge-0/0/0 description To_R0_1
set interfaces ge-0/0/0 vlan-tagging
set interfaces ge-0/0/0 unit 0 vlan-id 1
set interfaces ge-0/0/0 unit 0 family inet address 192.168.1.2/24
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:1000:10::1/64
set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/0 unit 1 vlan-id 2
set interfaces ge-0/0/0 unit 1 family inet address 192.168.20.1/24
set interfaces ge-0/0/0 unit 1 family iso
set interfaces ge-0/0/0 unit 1 family inet6 address 2001:db8:1000:20::1/64
set interfaces ge-0/0/0 unit 1 family mpls maximum-labels 8
set interfaces ge-0/0/2 description To_R2
set interfaces ge-0/0/2 vlan-tagging
set interfaces ge-0/0/2 unit 0 vlan-id 1
set interfaces ge-0/0/2 unit 0 family inet address 192.168.3.1/24
set interfaces ge-0/0/2 unit 0 family iso
set interfaces ge-0/0/2 unit 0 family inet6 address 2001:db8:1200:10::1/64
set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/2 unit 1 vlan-id 2
set interfaces ge-0/0/2 unit 1 family inet address 192.168.21.1/24
set interfaces ge-0/0/2 unit 1 family iso
set interfaces ge-0/0/2 unit 1 family inet6 address 2001:db8:1200:20::1/64
set interfaces ge-0/0/2 unit 1 family mpls maximum-labels 8
set interfaces ge-0/0/3 description to-R4
set interfaces ge-0/0/3 vlan-tagging
set interfaces ge-0/0/3 unit 0 vlan-id 1
set interfaces ge-0/0/3 unit 0 family inet address 192.168.4.1/24
set interfaces ge-0/0/3 unit 0 family iso
set interfaces ge-0/0/3 unit 0 family inet6 address 2001:db8:1400:10::1/64
set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/4 description to-R5
set interfaces ge-0/0/4 vlan-tagging
set interfaces ge-0/0/4 unit 0 vlan-id 1
set interfaces ge-0/0/4 unit 0 family inet address 192.168.5.1/24
set interfaces ge-0/0/4 unit 0 family iso
set interfaces ge-0/0/4 unit 0 family inet6 address 2001:db8:1500:10::1/64
set interfaces ge-0/0/4 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/4 unit 1 vlan-id 2
set interfaces ge-0/0/4 unit 1 family inet address 192.168.22.1/24
```

```
set interfaces ge-0/0/4 unit 1 family iso
set interfaces ge-0/0/4 unit 1 family inet6 address 2001:db8:1500:20::1/64
set interfaces ge-0/0/4 unit 1 family mpls maximum-labels 8
set interfaces lo0 unit 0 family inet address 10.1.1.1/32
set interfaces lo0 unit 0 family iso address 49.0001.0001.0101.0100
set interfaces lo0 unit 0 family inet6 address 2001:db8:abcd::10:01:01:01/128
set policy-options policy-statement bgplsepe_rt_2_ted term 1 from protocol bgp
set policy-options policy-statement bgplsepe_rt_2_ted term 1 then accept
set policy-options policy-statement expresspolsr1 from protocol spring-te
set policy-options policy-statement expresspolsr1 from route-filter 10.3.3.3/32 exact
set policy-options policy-statement expresspolsr1 then accept
set policy-options policy-statement nlri2bgp term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp term 1 then next-hop self
set policy-options policy-statement nlri2bgp term 1 then accept
set policy-options policy-statement nlri2bgp_epe term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp_epe term 1 from protocol bgp-ls-epe
set policy-options policy-statement nlri2bgp_epe term 1 then next-hop self
set policy-options policy-statement nlri2bgp_epe term 1 then accept
set policy-options policy-statement nlri2bgp_stat term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp_stat term 1 from protocol express-segments
set policy-options policy-statement nlri2bgp_stat term 1 then accept
set policy-options policy-statement nlri2ted_igp term 1 from traffic-engineering protocol isis-
level-2
set policy-options policy-statement nlri2ted_igp term 1 then accept
set policy-options policy-statement pplb then load-balance per-packet
set policy-options policy-statement prefix-sid term 1 from route-filter 10.1.1.1/32 exact
set policy-options policy-statement prefix-sid term 1 then prefix-segment index 1001
set policy-options policy-statement prefix-sid term 1 then prefix-segment node-segment
set policy-options policy-statement ted2nlri term 1 from protocol bgp-ls-epe
set policy-options policy-statement ted2nlri term 1 then accept
set policy-options policy-statement ted2nlri_epe_stat term 1 from family traffic-engineering
set policy-options policy-statement ted2nlri_epe_stat term 1 from protocol express-segments
set policy-options policy-statement ted2nlri_epe_stat term 1 then accept
set policy-options policy-statement ted2nlri_epe_stat term 2 from family traffic-engineering
set policy-options policy-statement ted2nlri_epe_stat term 2 from protocol bgp-ls-epe
set policy-options policy-statement ted2nlri_epe_stat term 2 then accept
set policy-options policy-statement ted2nlri_epe_stat term 3 from protocol isis
set policy-options policy-statement ted2nlri_epe_stat term 3 then reject
set policy-options policy-statement ted2nlri_igp term 1 from family traffic-engineering
set policy-options policy-statement ted2nlri_igp term 1 from protocol isis
set policy-options policy-statement ted2nlri_igp term 1 then accept
set routing-options router-id 10.1.1.1
set routing-options autonomous-system 65200
```

```
set routing-options forwarding-table export pplb
set protocols bgp group ebgp1 type external
set protocols bgp group ebgp1 family inet-vpn unicast
set protocols bgp group ebgp1 family traffic-engineering unicast
set protocols bgp group ebgp1 export nlri2bgp_stat
set protocols bgp group ebgp1 neighbor 192.168.1.1 peer-as 65100
set protocols bgp group ebgp1 neighbor 192.168.1.1 egress-te-adj-segment epe_adj1_toR0 label 8110
set protocols bgp group ebgp1 neighbor 192.168.1.1 egress-te-adj-segment epe_adj1_toR0 next-hop
192.168.1.1
set protocols bgp group ebgp1 neighbor 192.168.1.1 egress-te-adj-segment epe_adj1_toR0 te-link-
attribute te-metric 20
set protocols bgp group ebgp1 neighbor 192.168.1.1 egress-te-adj-segment epe_adj1_toR0 te-link-
attribute igp-metric 10
set protocols bgp group ebgp1 neighbor 192.168.1.1 egress-te-adj-segment epe_adj1_toR0 te-link-
attribute admin-group [ red brown ]
set protocols bgp group ibgp1 type internal
set protocols bgp group ibgp1 local-address 10.1.1.1
set protocols bgp group ibgp1 family traffic-engineering unicast
set protocols bgp group ibgp1 export nlri2bgp_epe
set protocols bgp group ibgp1 neighbor 10.2.2.2
set protocols bgp group ibgp1 neighbor 10.5.5.5
set protocols express-segments segment-set set1sr membership-policy expresspolsr1
set protocols express-segments traffic-engineering
set protocols isis interface ge-0/0/0.0 passive
set protocols isis interface ge-0/0/0.1 passive
set protocols isis interface ge-0/0/2.0 level 2 lan-neighbor 0100.0202.0202 ipv4-adjacency-
segment protected label 1211
set protocols isis interface ge-0/0/2.0 level 2 lan-neighbor 0100.0202.0202 ipv4-adjacency-
segment unprotected label 1201
set protocols isis interface ge-0/0/2.0 level 2 post-convergence-lfa node-protection
set protocols isis interface ge-0/0/2.1 level 2 lan-neighbor 0100.0202.0202 ipv4-adjacency-
segment protected label 1212
set protocols isis interface ge-0/0/2.1 level 2 lan-neighbor 0100.0202.0202 ipv4-adjacency-
segment unprotected label 1202
set protocols isis interface ge-0/0/2.1 level 2 post-convergence-lfa node-protection
set protocols isis interface ge-0/0/3.0 level 2 lan-neighbor 0100.0404.0404 ipv4-adjacency-
segment protected label 1411
set protocols isis interface ge-0/0/3.0 level 2 lan-neighbor 0100.0404.0404 ipv4-adjacency-
segment unprotected label 1401
set protocols isis interface ge-0/0/3.0 level 2 post-convergence-lfa node-protection
set protocols isis interface ge-0/0/4.0 level 2 lan-neighbor 0100.0505.0505 ipv4-adjacency-
segment protected label 1511
set protocols isis interface ge-0/0/4.0 level 2 lan-neighbor 0100.0505.0505 ipv4-adjacency-
```

```
segment unprotected label 1501
set protocols isis interface ge-0/0/4.0 level 2 post-convergence-lfa node-protection
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0 passive
set protocols isis source-packet-routing srgb start-label 800000 index-range 50000
set protocols isis level 1 disable
set protocols isis level 2 wide-metrics-only
set protocols isis backup-spf-options use-post-convergence-lfa
set protocols isis backup-spf-options use-source-packet-routing
set protocols isis traffic-engineering l3-unicast-topology
set protocols isis traffic-engineering advertisement always
set protocols isis export prefix-sid
set protocols mpls traffic-engineering database import l3-unicast-topology bgp-link-state
set protocols mpls traffic-engineering database import policy ted2nlri_epe_stat
set protocols mpls traffic-engineering database export l3-unicast-topology
set protocols mpls admin-groups red 0
set protocols mpls admin-groups blue 1
set protocols mpls admin-groups green 2
set protocols mpls admin-groups yellow 3
set protocols mpls admin-groups orange 4
set protocols mpls admin-groups brown 5
set protocols mpls admin-groups black 6
set protocols mpls admin-groups pink 7
set protocols mpls label-range static-label-range 1000 70000
set protocols mpls interface ge-0/0/3.0 admin-group red
set protocols mpls interface ge-0/0/2.0 admin-group brown
set protocols mpls interface ge-0/0/2.1 admin-group yellow
set protocols mpls interface ge-0/0/4.0 admin-group blue
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols source-packet-routing segment-list R1-R2-R3 hop1 label 1211
set protocols source-packet-routing segment-list R1-R2-R3 hop2 label 801003
set protocols source-packet-routing source-routing-path lsp1to3_sr to 10.3.3.3
set protocols source-packet-routing source-routing-path lsp1to3_sr primary R1-R2-R3
```

Configure the following additional commands in **Device R1 (BN1 router)** for colored SR-TE underlay path.

```
set routing-options rib-groups color-to-inet3 import-rib [ junos-rti-tc-<color>.inet.0 inet.3 ]
set protocols source-packet-routing no-chained-composite-next-hop
```

```
set protocols source-packet-routing source-routing-path lsp1to3_sr color 1000
set protocols source-packet-routing rib-group ipv4-color color-to-inet3
```

Device R4 (BN2 router)

```
set interfaces ge-0/0/0 description To_R0
set interfaces ge-0/0/0 vlan-tagging
set interfaces ge-0/0/0 unit 0 vlan-id 1
set interfaces ge-0/0/0 unit 0 family inet address 192.168.2.2/24
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:4000:10::4/64
set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/0 unit 1 vlan-id 2
set interfaces ge-0/0/0 unit 1 family inet address 192.168.40.2/24
set interfaces ge-0/0/0 unit 1 family iso
set interfaces ge-0/0/0 unit 1 family inet6 address 2001:db8:4000:20::4/64
set interfaces ge-0/0/0 unit 1 family mpls maximum-labels 8
set interfaces ge-0/0/2 description To_R1
set interfaces ge-0/0/2 vlan-tagging
set interfaces ge-0/0/2 unit 0 vlan-id 1
set interfaces ge-0/0/2 unit 0 family inet address 192.168.4.2/24
set interfaces ge-0/0/2 unit 0 family iso
set interfaces ge-0/0/2 unit 0 family inet6 address 2001:db8:1400:10::4/64
set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/3 description To_R2
set interfaces ge-0/0/3 vlan-tagging
set interfaces ge-0/0/3 unit 0 vlan-id 1
set interfaces ge-0/0/3 unit 0 family inet address 192.168.7.1/24
set interfaces ge-0/0/3 unit 0 family iso
set interfaces ge-0/0/3 unit 0 family inet6 address 2001:db8:2400:10::4/64
set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/3 unit 1 vlan-id 2
set interfaces ge-0/0/3 unit 1 family inet address 192.168.24.1/24
set interfaces ge-0/0/3 unit 1 family iso
set interfaces ge-0/0/3 unit 1 family inet6 address 2001:db8:2400:20::4/64
set interfaces ge-0/0/3 unit 1 family mpls maximum-labels 8
set interfaces ge-0/0/4 description To_R5
set interfaces ge-0/0/4 vlan-tagging
set interfaces ge-0/0/4 unit 0 vlan-id 1
set interfaces ge-0/0/4 unit 0 family inet address 192.168.13.1/24
set interfaces ge-0/0/4 unit 0 family iso
set interfaces ge-0/0/4 unit 0 family inet6 address 2001:db8:4500:10::4/64
```

```
set interfaces ge-0/0/4 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/4 unit 1 vlan-id 2
set interfaces ge-0/0/4 unit 1 family inet address 192.168.45.1/24
set interfaces ge-0/0/4 unit 1 family iso
set interfaces ge-0/0/4 unit 1 family inet6 address 2001:db8:4500:20::4/64
set interfaces ge-0/0/4 unit 1 family mpls maximum-labels 8
set interfaces lo0 unit 0 family inet address 10.4.4.4/32
set interfaces lo0 unit 0 family iso address 49.0001.0004.0404.0400
set interfaces lo0 unit 0 family inet6 address abcd::04:04:04:04/128
set policy-options policy-statement expresspolsr1 from protocol spring-te
set policy-options policy-statement expresspolsr1 from route-filter 10.6.6.6/32 exact
set policy-options policy-statement expresspolsr1 then accept
set policy-options policy-statement nlri2bgp_epe term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp_epe term 1 from protocol bgp-ls-epe
set policy-options policy-statement nlri2bgp_epe term 1 then next-hop self
set policy-options policy-statement nlri2bgp_epe term 1 then accept
set policy-options policy-statement nlri2bgp_stat term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp_stat term 1 from protocol express-segments
set policy-options policy-statement nlri2bgp_stat term 1 then accept
set policy-options policy-statement pplb then load-balance per-packet
set policy-options policy-statement prefix-sid term 1 from route-filter 10.4.4.4/32 exact
set policy-options policy-statement prefix-sid term 1 then prefix-segment index 1004
set policy-options policy-statement prefix-sid term 1 then prefix-segment node-segment
set policy-options policy-statement ted2nlri_epe_stat term 1 from family traffic-engineering
set policy-options policy-statement ted2nlri_epe_stat term 1 from protocol express-segments
set policy-options policy-statement ted2nlri_epe_stat term 1 then accept
set policy-options policy-statement ted2nlri_epe_stat term 2 from family traffic-engineering
set policy-options policy-statement ted2nlri_epe_stat term 2 from protocol bgp-ls-epe
set policy-options policy-statement ted2nlri_epe_stat term 2 then accept
set policy-options policy-statement ted2nlri_epe_stat term 3 from protocol isis
set policy-options policy-statement ted2nlri_epe_stat term 3 then reject
set routing-options router-id 10.4.4.4
set routing-options autonomous-system 65200

set routing-options forwarding-table export pplb
set protocols bgp group ibgp1 type internal
set protocols bgp group ibgp1 local-address 10.4.4.4
set protocols bgp group ibgp1 family traffic-engineering unicast
set protocols bgp group ibgp1 export nlri2bgp_epe
set protocols bgp group ibgp1 neighbor 10.2.2.2
set protocols bgp group ibgp1 neighbor 10.5.5.5
set protocols bgp group ebgp1 type external
set protocols bgp group ebgp1 family inet-vpn unicast
```

```
set protocols bgp group ebgp1 family traffic-engineering unicast
set protocols bgp group ebgp1 export nlri2bgp_stat
set protocols bgp group ebgp1 neighbor 192.168.2.1 peer-as 65100
set protocols bgp group ebgp1 neighbor 192.168.2.1 egress-te-adj-segment epe_adj1_toR0 label 8140
set protocols bgp group ebgp1 neighbor 192.168.2.1 egress-te-adj-segment epe_adj1_toR0 next-hop
192.168.2.1
set protocols bgp group ebgp1 neighbor 192.168.2.1 egress-te-adj-segment epe_adj1_toR0 te-link-
attribute te-metric 20
set protocols bgp group ebgp1 neighbor 192.168.2.1 egress-te-adj-segment epe_adj1_toR0 te-link-
attribute igp-metric 10
set protocols bgp group ebgp1 neighbor 192.168.2.1 egress-te-adj-segment epe_adj1_toR0 te-link-
attribute admin-group [ red brown ]
set protocols express-segments segment-set set4sr membership-policy expresspolsr1
set protocols express-segments traffic-engineering
set protocols isis interface ge-0/0/0.0 level 2 post-convergence-lfa node-protection
set protocols isis interface ge-0/0/0.0 passive
set protocols isis interface ge-0/0/2.0 level 2 lan-neighbor 0100.0101.0101 ipv4-adjacency-
segment protected label 4111
set protocols isis interface ge-0/0/2.0 level 2 lan-neighbor 0100.0101.0101 ipv4-adjacency-
segment unprotected label 4101
set protocols isis interface ge-0/0/2.0 level 2 post-convergence-lfa node-protection
set protocols isis interface ge-0/0/3.0 level 2 lan-neighbor 0100.0202.0202 ipv4-adjacency-
segment protected label 4211
set protocols isis interface ge-0/0/3.0 level 2 lan-neighbor 0100.0202.0202 ipv4-adjacency-
segment unprotected label 4201
set protocols isis interface ge-0/0/3.0 level 2 post-convergence-lfa node-protection
set protocols isis interface ge-0/0/4.0 level 2 lan-neighbor 0100.0505.0505 ipv4-adjacency-
segment protected label 4511
set protocols isis interface ge-0/0/4.0 level 2 lan-neighbor 0100.0505.0505 ipv4-adjacency-
segment unprotected label 4501
set protocols isis interface ge-0/0/4.0 level 2 post-convergence-lfa node-protection
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0 passive
set protocols isis source-packet-routing srgb start-label 800000 index-range 50000
set protocols isis level 1 disable
set protocols isis level 2 wide-metrics-only
set protocols isis backup-spf-options use-post-convergence-lfa
set protocols isis backup-spf-options use-source-packet-routing
set protocols isis traffic-engineering l3-unicast-topology
set protocols isis traffic-engineering advertisement always
set protocols isis export prefix-sid
set protocols mpls traffic-engineering database import l3-unicast-topology bgp-link-state
set protocols mpls traffic-engineering database import policy ted2nlri_epe_stat
```

```
set protocols mpls traffic-engineering database export l3-unicast-topology
set protocols mpls admin-groups red 0
set protocols mpls admin-groups blue 1
set protocols mpls admin-groups green 2
set protocols mpls admin-groups yellow 3
set protocols mpls admin-groups orange 4
set protocols mpls admin-groups brown 5
set protocols mpls admin-groups black 6
set protocols mpls admin-groups pink 7
set protocols mpls label-range static-label-range 1000 70000
set protocols mpls interface ge-0/0/2.0 admin-group red
set protocols mpls interface ge-0/0/3.0 admin-group green
set protocols mpls interface ge-0/0/4.0 admin-group brown
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols source-packet-routing segment-list R4-R5-R6 hop1 label 4511
set protocols source-packet-routing segment-list R4-R5-R6 hop2 label 5601
set protocols source-packet-routing source-routing-path lsp4to6_sr to 10.6.6.6
set protocols source-packet-routing source-routing-path lsp4to6_sr primary R4-R5-R6
```

Configure the following additional commands in **Device R4 (BN2 router)** for colored SR-TE underlay path.

```
set routing-options rib-groups color-to-inet3 import-rib [ junos-rti-tc-<color>.inet.0 inet.3 ]
set protocols source-packet-routing no-chained-composite-next-hop
set protocols source-packet-routing source-routing-path lsp4to6_sr color 1000
set protocols source-packet-routing rib-group ipv4-color color-to-inet3
```

Device R2 (Intermediate router)

```
set interfaces ge-0/0/0 description To_R1
set interfaces ge-0/0/0 vlan-tagging
set interfaces ge-0/0/0 unit 0 vlan-id 1
set interfaces ge-0/0/0 unit 0 family inet address 192.168.3.2/24
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:1200:10::2/64
set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/0 unit 1 vlan-id 2
set interfaces ge-0/0/0 unit 1 family inet address 192.168.21.2/24
set interfaces ge-0/0/0 unit 1 family iso
set interfaces ge-0/0/0 unit 1 family inet6 address 2001:db8:1200:20::2/64
```

```
set interfaces ge-0/0/0 unit 1 family mpls maximum-labels 8
set interfaces ge-0/0/1 description To_R3
set interfaces ge-0/0/1 vlan-tagging
set interfaces ge-0/0/1 unit 0 vlan-id 1
set interfaces ge-0/0/1 unit 0 family inet address 192.168.6.1/24
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:2300:10::2/64
set interfaces ge-0/0/1 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/1 unit 1 vlan-id 2
set interfaces ge-0/0/1 unit 1 family inet address 192.168.23.1/24
set interfaces ge-0/0/1 unit 1 family iso
set interfaces ge-0/0/1 unit 1 family inet6 address 2001:db8:2300:20::2/64
set interfaces ge-0/0/1 unit 1 family mpls maximum-labels 8
set interfaces ge-0/0/2 description To_R4
set interfaces ge-0/0/2 vlan-tagging
set interfaces ge-0/0/2 unit 0 vlan-id 1
set interfaces ge-0/0/2 unit 0 family inet address 192.168.7.1/24
set interfaces ge-0/0/2 unit 0 family iso
set interfaces ge-0/0/2 unit 0 family inet6 address 2001:db8:2400:10::2/64
set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/2 unit 1 vlan-id 2
set interfaces ge-0/0/2 unit 1 family inet address 192.168.24.1/24
set interfaces ge-0/0/2 unit 1 family iso
set interfaces ge-0/0/2 unit 1 family inet6 address 2001:db8:2400:20::2/64
set interfaces ge-0/0/2 unit 1 family mpls maximum-labels 8
set interfaces ge-0/0/3 description To_R5
set interfaces ge-0/0/3 vlan-tagging
set interfaces ge-0/0/3 unit 0 vlan-id 1
set interfaces ge-0/0/3 unit 0 family inet address 192.168.8.1/24
set interfaces ge-0/0/3 unit 0 family iso
set interfaces ge-0/0/3 unit 0 family inet6 address 2001:db8:2500:10::2/64
set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/4 description To_R6
set interfaces ge-0/0/4 vlan-tagging
set interfaces ge-0/0/4 unit 0 vlan-id 1
set interfaces ge-0/0/4 unit 0 family inet address 192.168.9.1/24
set interfaces ge-0/0/4 unit 0 family iso
set interfaces ge-0/0/4 unit 0 family inet6 address 2001:db8:2600:10::2/64
set interfaces ge-0/0/4 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/4 unit 1 vlan-id 2
set interfaces ge-0/0/4 unit 1 family inet address 192.168.26.1/24
set interfaces ge-0/0/4 unit 1 family iso
set interfaces ge-0/0/4 unit 1 family inet6 address 2001:db8:2600:20::2/64
```

```
set interfaces ge-0/0/4 unit 1 family mpls maximum-labels 8
set interfaces lo0 unit 0 family inet address 10.2.2.2/32
set interfaces lo0 unit 0 family iso address 49.0001.0002.0202.0200
set interfaces lo0 unit 0 family inet6 address 2001:db8:abcd::02:02:02:02/128
set policy-options policy-statement bgplsepe_rt_2_ted term 1 from protocol bgp
set policy-options policy-statement bgplsepe_rt_2_ted term 1 then accept
set policy-options policy-statement nlri2bgp term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp term 1 then next-hop self
set policy-options policy-statement nlri2bgp term 1 then accept
set policy-options policy-statement nlri2bgp_igp term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp_igp term 1 from protocol isis
set policy-options policy-statement nlri2bgp_igp term 1 then accept
set policy-options policy-statement nlri2ted_igp term 1 from traffic-engineering protocol isis-
level-2
set policy-options policy-statement nlri2ted_igp term 1 then accept
set policy-options policy-statement pplb then load-balance per-packet
set policy-options policy-statement prefix-sid term 1 from route-filter 10.2.2.2/32 exact
set policy-options policy-statement prefix-sid term 1 then prefix-segment index 1002
set policy-options policy-statement prefix-sid term 1 then prefix-segment node-segment
set policy-options policy-statement ted2nlri term 1 from protocol bgp-ls-epe
set policy-options policy-statement ted2nlri term 1 then accept
set policy-options policy-statement ted2nlri_1 term 1 from traffic-engineering
set policy-options policy-statement ted2nlri_1 term 1 then accept
set policy-options policy-statement ted2nlri_igp term 1 from family traffic-engineering
set policy-options policy-statement ted2nlri_igp term 1 from protocol isis
set policy-options policy-statement ted2nlri_igp term 1 then accept
set routing-options router-id 10.2.2.2
set routing-options autonomous-system 65200
set routing-options forwarding-table export pplb
set protocols bgp group RR1 type internal
set protocols bgp group RR1 local-address 10.2.2.2
set protocols bgp group RR1 family traffic-engineering unicast
set protocols bgp group RR1 neighbor 10.1.1.1
set protocols bgp group RR1 neighbor 10.3.3.3
set protocols bgp group RR1 neighbor 10.6.6.6
set protocols bgp group RR1 neighbor 10.4.4.4
set protocols bgp cluster 10.2.2.2
set protocols isis interface ge-0/0/0.0 level 2 lan-neighbor 0100.0101.0101 ipv4-adjacency-
segment protected label 2111
set protocols isis interface ge-0/0/0.0 level 2 lan-neighbor 0100.0101.0101 ipv4-adjacency-
segment unprotected label 2101
set protocols isis interface ge-0/0/0.0 level 2 post-convergence-lfa node-protection
set protocols isis interface ge-0/0/0.1 level 2 lan-neighbor 0100.0101.0101 ipv4-adjacency-
```

```
segment protected label 2112
set protocols isis interface ge-0/0/0.1 level 2 lan-neighbor 0100.0101.0101 ipv4-adjacency-
segment unprotected label 2102
set protocols isis interface ge-0/0/0.1 level 2 post-convergence-lfa node-protection
set protocols isis interface ge-0/0/1.0 level 2 lan-neighbor 0100.0303.0303 ipv4-adjacency-
segment protected label 2311
set protocols isis interface ge-0/0/1.0 level 2 lan-neighbor 0100.0303.0303 ipv4-adjacency-
segment unprotected label 2301
set protocols isis interface ge-0/0/1.0 level 2 post-convergence-lfa node-protection
set protocols isis interface ge-0/0/2.0 level 2 lan-neighbor 0100.0404.0404 ipv4-adjacency-
segment protected label 2411
set protocols isis interface ge-0/0/2.0 level 2 lan-neighbor 0100.0404.0404 ipv4-adjacency-
segment unprotected label 2401
set protocols isis interface ge-0/0/2.0 level 2 post-convergence-lfa node-protection
set protocols isis interface ge-0/0/3.0 level 2 lan-neighbor 0100.0505.0505 ipv4-adjacency-
segment protected label 2511
set protocols isis interface ge-0/0/3.0 level 2 lan-neighbor 0100.0505.0505 ipv4-adjacency-
segment unprotected label 2501
set protocols isis interface ge-0/0/3.0 level 2 post-convergence-lfa node-protection
set protocols isis interface ge-0/0/4.0 level 2 lan-neighbor 0100.0606.0606 ipv4-adjacency-
segment protected label 2611
set protocols isis interface ge-0/0/4.0 level 2 lan-neighbor 0100.0606.0606 ipv4-adjacency-
segment unprotected label 2601
set protocols isis interface ge-0/0/4.0 level 2 post-convergence-lfa node-protection
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0 passive
set protocols isis source-packet-routing srgb start-label 800000 index-range 50000
set protocols isis level 1 disable
set protocols isis level 2 wide-metrics-only
set protocols isis backup-spf-options use-post-convergence-lfa
set protocols isis backup-spf-options use-source-packet-routing
set protocols isis traffic-engineering l3-unicast-topology
set protocols isis traffic-engineering advertisement always
set protocols isis export prefix-sid
set protocols mpls admin-groups red 0
set protocols mpls admin-groups blue 1
set protocols mpls admin-groups green 2
set protocols mpls admin-groups yellow 3
set protocols mpls admin-groups orange 4
set protocols mpls admin-groups brown 5
set protocols mpls admin-groups black 6
set protocols mpls admin-groups pink 7
set protocols mpls label-range static-label-range 1000 70000
```

```
set protocols mpls interface ge-0/0/0.0 admin-group brown
set protocols mpls interface ge-0/0/0.1 admin-group yellow
set protocols mpls interface ge-0/0/2.0 admin-group green
set protocols mpls interface ge-0/0/3.0 admin-group red
set protocols mpls interface ge-0/0/4.0 admin-group blue
set protocols mpls interface ge-0/0/1.0 admin-group brown
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
```

Device R5 (Intermediate router)

```
set interfaces ge-0/0/0 description To_R1
set interfaces ge-0/0/0 vlan-tagging
set interfaces ge-0/0/0 unit 0 vlan-id 1
set interfaces ge-0/0/0 unit 0 family inet address 192.168.5.2/24
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:1500:10::5/64
set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/0 unit 1 vlan-id 2
set interfaces ge-0/0/0 unit 1 family inet address 192.168.22.2/24
set interfaces ge-0/0/0 unit 1 family iso
set interfaces ge-0/0/0 unit 1 family inet6 address 2001:db8:1500:20::5/64
set interfaces ge-0/0/0 unit 1 family mpls maximum-labels 8
set interfaces ge-0/0/1 description To_R2
set interfaces ge-0/0/1 vlan-tagging
set interfaces ge-0/0/1 unit 0 vlan-id 1
set interfaces ge-0/0/1 unit 0 family inet address 192.168.8.2/24
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:2500:10::5/64
set interfaces ge-0/0/1 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/2 description To_R3
set interfaces ge-0/0/2 vlan-tagging
set interfaces ge-0/0/2 unit 0 vlan-id 1
set interfaces ge-0/0/2 unit 0 family inet address 192.168.10.2/24
set interfaces ge-0/0/2 unit 0 family iso
set interfaces ge-0/0/2 unit 0 family inet6 address 2001:db8:3500:10::5/64
set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/2 unit 1 vlan-id 2
set interfaces ge-0/0/2 unit 1 family inet address 192.168.35.2/24
set interfaces ge-0/0/2 unit 1 family iso
set interfaces ge-0/0/2 unit 1 family inet6 address 2001:db8:3500:20::5/64
set interfaces ge-0/0/2 unit 1 family mpls maximum-labels 8
```

```
set interfaces ge-0/0/3 description To_R4
set interfaces ge-0/0/3 vlan-tagging
set interfaces ge-0/0/3 unit 0 vlan-id 1
set interfaces ge-0/0/3 unit 0 family inet address 192.168.13.2/24
set interfaces ge-0/0/3 unit 0 family iso
set interfaces ge-0/0/3 unit 0 family inet6 address 2001:db8:4500:10::5/64
set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/3 unit 1 vlan-id 2
set interfaces ge-0/0/3 unit 1 family inet address 192.168.45.2/24
set interfaces ge-0/0/3 unit 1 family iso
set interfaces ge-0/0/3 unit 1 family inet6 address 2001:db8:4500:20::5/64
set interfaces ge-0/0/3 unit 1 family mpls maximum-labels 8
set interfaces ge-0/0/4 description To_R6
set interfaces ge-0/0/4 vlan-tagging
set interfaces ge-0/0/4 unit 0 vlan-id 1
set interfaces ge-0/0/4 unit 0 family inet address 192.168.14.1/24
set interfaces ge-0/0/4 unit 0 family iso
set interfaces ge-0/0/4 unit 0 family inet6 address 2001:db8:5600:10::5/64
set interfaces ge-0/0/4 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/4 unit 1 vlan-id 2
set interfaces ge-0/0/4 unit 1 family inet address 192.168.56.1/24
set interfaces ge-0/0/4 unit 1 family iso
set interfaces ge-0/0/4 unit 1 family inet6 address 2001:db8:5600:20::5/64
set interfaces ge-0/0/4 unit 1 family mpls maximum-labels 8
set interfaces lo0 unit 0 family inet address 10.5.5.5/32
set interfaces lo0 unit 0 family iso address 49.0001.0005.0505.0500
set interfaces lo0 unit 0 family inet6 address 2001:db8:abcd::05:05:05:05/128
set policy-options policy-statement nlri2bgp term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp term 1 then next-hop self
set policy-options policy-statement nlri2bgp term 1 then accept
set policy-options policy-statement nlri2ted_igp term 1 from traffic-engineering protocol isis-
level-2
set policy-options policy-statement nlri2ted_igp term 1 then accept
set policy-options policy-statement pplb then load-balance per-packet
set policy-options policy-statement prefix-sid term 1 from route-filter 10.5.5.5/32 exact
set policy-options policy-statement prefix-sid term 1 then prefix-segment index 1005
set policy-options policy-statement prefix-sid term 1 then prefix-segment node-segment
set policy-options policy-statement ted2nlri term 1 from protocol bgp-ls-epe
set policy-options policy-statement ted2nlri term 1 then accept
set policy-options policy-statement ted2nlri_igp term 1 from family traffic-engineering
set policy-options policy-statement ted2nlri_igp term 1 from protocol isis
set policy-options policy-statement ted2nlri_igp term 1 then accept
set routing-options router-id 10.5.5.5
```

```
set routing-options autonomous-system 65200
set routing-options forwarding-table export pplb
set protocols bgp group RR2 type internal
set protocols bgp group RR2 family inet unicast
set protocols bgp group RR2 family traffic-engineering unicast
set protocols bgp group RR2 neighbor 10.1.1.1
set protocols bgp group RR2 neighbor 10.3.3.3
set protocols bgp group RR2 neighbor 10.6.6.6
set protocols bgp group RR2 neighbor 10.4.4.4
set protocols bgp cluster 10.5.5.5
set protocols isis interface ge-0/0/0.0 level 2 lan-neighbor 0100.0101.0101 ipv4-adjacency-
segment protected label 5111
set protocols isis interface ge-0/0/0.0 level 2 lan-neighbor 0100.0101.0101 ipv4-adjacency-
segment unprotected label 5101
set protocols isis interface ge-0/0/0.0 level 2 post-convergence-lfa node-protection
set protocols isis interface ge-0/0/1.0 level 2 lan-neighbor 0100.0202.0202 ipv4-adjacency-
segment protected label 5211
set protocols isis interface ge-0/0/1.0 level 2 lan-neighbor 0100.0202.0202 ipv4-adjacency-
segment unprotected label 5201
set protocols isis interface ge-0/0/1.0 level 2 post-convergence-lfa node-protection
set protocols isis interface ge-0/0/2.0 level 2 lan-neighbor 0100.0303.0303 ipv4-adjacency-
segment protected label 5311
set protocols isis interface ge-0/0/2.0 level 2 lan-neighbor 0100.0303.0303 ipv4-adjacency-
segment unprotected label 5301
set protocols isis interface ge-0/0/2.0 level 2 post-convergence-lfa node-protection
set protocols isis interface ge-0/0/3.0 level 2 lan-neighbor 0100.0404.0404 ipv4-adjacency-
segment protected label 5411
set protocols isis interface ge-0/0/3.0 level 2 lan-neighbor 0100.0404.0404 ipv4-adjacency-
segment unprotected label 5401
set protocols isis interface ge-0/0/3.0 level 2 post-convergence-lfa node-protection
set protocols isis interface ge-0/0/4.0 level 2 lan-neighbor 0100.0606.0606 ipv4-adjacency-
segment protected label 5611
set protocols isis interface ge-0/0/4.0 level 2 lan-neighbor 0100.0606.0606 ipv4-adjacency-
segment unprotected label 5601
set protocols isis interface ge-0/0/4.0 level 2 post-convergence-lfa node-protection
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0 passive
set protocols isis source-packet-routing srgb start-label 800000 index-range 50000
set protocols isis level 1 disable
set protocols isis backup-spf-options use-post-convergence-lfa
set protocols isis backup-spf-options use-source-packet-routing
set protocols isis traffic-engineering l3-unicast-topology
set protocols isis traffic-engineering advertisement always
```

```
set protocols isis export prefix-sid
set protocols mpls admin-groups red 0
set protocols mpls admin-groups blue 1
set protocols mpls admin-groups green 2
set protocols mpls admin-groups yellow 3
set protocols mpls admin-groups orange 4
set protocols mpls admin-groups brown 5
set protocols mpls admin-groups black 6
set protocols mpls admin-groups pink 7
set protocols mpls label-range static-label-range 1000 70000
set protocols mpls interface ge-0/0/0.0 admin-group blue
set protocols mpls interface ge-0/0/1.0 admin-group red
set protocols mpls interface ge-0/0/2.0 admin-group green
set protocols mpls interface ge-0/0/3.0 admin-group brown
set protocols mpls interface ge-0/0/4.0 admin-group brown
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
```

Device R3 (BN3 router)

```
set interfaces ge-0/0/0 description To_R2
set interfaces ge-0/0/0 vlan-tagging
set interfaces ge-0/0/0 unit 0 vlan-id 1
set interfaces ge-0/0/0 unit 0 family inet address 192.168.6.2/24
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:2300:10::3/64
set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/0 unit 1 vlan-id 2
set interfaces ge-0/0/0 unit 1 family inet address 192.168.23.2/24
set interfaces ge-0/0/0 unit 1 family iso
set interfaces ge-0/0/0 unit 1 family inet6 address 2001:db8:2300:20::3/64
set interfaces ge-0/0/0 unit 1 family mpls maximum-labels 8
set interfaces ge-0/0/0 unit 2 vlan-id 3
set interfaces ge-0/0/0 unit 2 family inet address 192.168.30.2/24
set interfaces ge-0/0/0 unit 2 family iso
set interfaces ge-0/0/0 unit 2 family inet6 address 2001:db8:2300:30::3/64
set interfaces ge-0/0/0 unit 2 family mpls maximum-labels 8
set interfaces ge-0/0/1 description To_R5
set interfaces ge-0/0/1 vlan-tagging
set interfaces ge-0/0/1 unit 0 vlan-id 1
set interfaces ge-0/0/1 unit 0 family inet address 192.168.10.1/24
set interfaces ge-0/0/1 unit 0 family iso
```

```
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:3500:10::3/64
set interfaces ge-0/0/1 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/1 unit 1 vlan-id 2
set interfaces ge-0/0/1 unit 1 family inet address 192.168.35.1/24
set interfaces ge-0/0/1 unit 1 family iso
set interfaces ge-0/0/1 unit 1 family inet6 address 2001:db8:3500:20::3/64
set interfaces ge-0/0/1 unit 1 family mpls maximum-labels 8
set interfaces ge-0/0/2 description To_R6
set interfaces ge-0/0/2 vlan-tagging
set interfaces ge-0/0/2 unit 0 vlan-id 1
set interfaces ge-0/0/2 unit 0 family inet address 192.168.11.1/24
set interfaces ge-0/0/2 unit 0 family iso
set interfaces ge-0/0/2 unit 0 family inet6 address 2001:db8:3600:10::3/64
set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/3 description To_R7
set interfaces ge-0/0/3 vlan-tagging
set interfaces ge-0/0/3 unit 0 vlan-id 1
set interfaces ge-0/0/3 unit 0 family inet address 192.168.12.1/24
set interfaces ge-0/0/3 unit 0 family iso
set interfaces ge-0/0/3 unit 0 family inet6 address 2001:db8:3700:10::3/6
set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/3 unit 1 vlan-id 2
set interfaces ge-0/0/3 unit 1 family inet address 192.168.37.1/24
set interfaces ge-0/0/3 unit 1 family iso
set interfaces ge-0/0/3 unit 1 family inet6 address 2001:db8:3700:20::3/6
set interfaces ge-0/0/3 unit 1 family mpls maximum-labels 8
set interfaces lo0 unit 0 family inet address 10.3.3.3/32
set interfaces lo0 unit 0 family iso address 49.0001.0003.0303.0300
set interfaces lo0 unit 0 family inet6 address 2001:db8:abcd::03:03:03:03/128
set policy-options policy-statement bgplsepe_rt_2_ted term 1 from protocol bgp
set policy-options policy-statement bgplsepe_rt_2_ted term 1 then accept
set policy-options policy-statement expresspol1 from route-filter 10.1.1.1/32 exact install-
nexthop lsp lsp3to1_a
set policy-options policy-statement expresspol1 then accept
set policy-options policy-statement expresspol2 from route-filter 10.4.4.4/32 exact install-
nexthop lsp lsp3to4_a
set policy-options policy-statement expresspol2 then accept
set policy-options policy-statement expresspolsr1 from protocol spring-te
set policy-options policy-statement expresspolsr1 from route-filter 10.1.1.1/32 exact
set policy-options policy-statement expresspolsr1 then accept
set policy-options policy-statement nlri2bgp term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp term 1 then next-hop self
set policy-options policy-statement nlri2bgp term 1 then accept
```

```
set policy-options policy-statement nlri2bgp_epe term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp_epe term 1 from protocol bgp-ls-epe
set policy-options policy-statement nlri2bgp_epe term 1 then next-hop self
set policy-options policy-statement nlri2bgp_epe term 1 then accept
set policy-options policy-statement nlri2bgp_igp term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp_igp term 1 from protocol isis
set policy-options policy-statement nlri2bgp_igp term 1 then accept
set policy-options policy-statement nlri2bgp_stat term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp_stat term 1 from protocol express-segments
set policy-options policy-statement nlri2bgp_stat term 1 then accept
set policy-options policy-statement pplb then load-balance per-packet
set policy-options policy-statement prefix-sid term 1 from route-filter 10.3.3.3/32 exact
set policy-options policy-statement prefix-sid term 1 then prefix-segment index 1003
set policy-options policy-statement prefix-sid term 1 then prefix-segment node-segment
set policy-options policy-statement ted2nlri term 1 from protocol bgp-ls-epe
set policy-options policy-statement ted2nlri term 1 then accept
set policy-options policy-statement ted2nlri_epe_stat term 1 from family traffic-engineering
set policy-options policy-statement ted2nlri_epe_stat term 1 from protocol static
set policy-options policy-statement ted2nlri_epe_stat term 1 then accept
set policy-options policy-statement ted2nlri_epe_stat term 2 from family traffic-engineering
set policy-options policy-statement ted2nlri_epe_stat term 2 from protocol bgp-ls-epe
set policy-options policy-statement ted2nlri_epe_stat term 2 then accept
set policy-options policy-statement ted2nlri_epe_stat term 3 from protocol isis
set policy-options policy-statement ted2nlri_epe_stat term 3 then reject
set policy-options policy-statement ted2nlri_igp from family traffic-engineering
set policy-options policy-statement ted2nlri_igp from protocol isis
set policy-options policy-statement ted2nlri_igp then accept
set routing-options router-id 10.3.3.3
set routing-options autonomous-system 65200
set routing-options forwarding-table export pplb
set protocols bgp group ibgp1 type internal
set protocols bgp group ibgp1 local-address 10.3.3.3
set protocols bgp group ibgp1 family traffic-engineering unicast
set protocols bgp group ibgp1 export nlri2bgp_epe
set protocols bgp group ibgp1 neighbor 10.2.2.2
set protocols bgp group ibgp1 neighbor 10.5.5.5
set protocols bgp group ebgp1 type external
set protocols bgp group ebgp1 family traffic-engineering unicast
set protocols bgp group ebgp1 export nlri2bgp_stat
set protocols bgp group ebgp1 neighbor 192.168.12.2 peer-as 65300
set protocols bgp group ebgp1 neighbor 192.168.12.2 egress-te-adj-segment epe_adj1_toR7 label
7137
set protocols bgp group ebgp1 neighbor 192.168.12.2 egress-te-adj-segment epe_adj1_toR7 next-hop
```

```
192.168.12.2
set protocols bgp group ebgp1 neighbor 192.168.12.2 egress-te-adj-segment epe_adj1_toR7 te-link-
attribute te-metric 20
set protocols bgp group ebgp1 neighbor 192.168.12.2 egress-te-adj-segment epe_adj1_toR7 te-link-
attribute igp-metric 10
set protocols bgp group ebgp1 neighbor 192.168.12.2 egress-te-adj-segment epe_adj1_toR7 te-link-
attribute admin-group [ red brown ]
set protocols bgp group ebgp1 vpn-apply-export
set protocols express-segments segment-set set3sr membership-policy expresspolsr1
set protocols express-segments traffic-engineering
set protocols isis interface ge-0/0/0.0 level 2 lan-neighbor 0100.0202.0202 ipv4-adjacency-
segment protected label 3211
set protocols isis interface ge-0/0/0.0 level 2 lan-neighbor 0100.0202.0202 ipv4-adjacency-
segment unprotected label 3201
set protocols isis interface ge-0/0/0.0 level 2 post-convergence-lfa node-protection
set protocols isis interface ge-0/0/1.0 level 2 lan-neighbor 0100.0505.0505 ipv4-adjacency-
segment protected label 3511
set protocols isis interface ge-0/0/1.0 level 2 lan-neighbor 0100.0505.0505 ipv4-adjacency-
segment unprotected label 3501
set protocols isis interface ge-0/0/1.0 level 2 post-convergence-lfa node-protection
set protocols isis interface ge-0/0/2.0 level 2 lan-neighbor 0100.0606.0606 ipv4-adjacency-
segment protected label 3611
set protocols isis interface ge-0/0/2.0 level 2 lan-neighbor 0100.0606.0606 ipv4-adjacency-
segment unprotected label 3601
set protocols isis interface ge-0/0/2.0 level 2 post-convergence-lfa node-protection
set protocols isis interface ge-0/0/3.0 passive
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0 passive
set protocols isis source-packet-routing srgb start-label 800000 index-range 50000
set protocols isis level 1 disable
set protocols isis level 2 wide-metrics-only
set protocols isis backup-spf-options use-post-convergence-lfa
set protocols isis backup-spf-options use-source-packet-routing
set protocols isis traffic-engineering l3-unicast-topology
set protocols isis traffic-engineering advertisement always
set protocols isis export prefix-sid
set protocols mpls traffic-engineering database import l3-unicast-topology bgp-link-state
set protocols mpls traffic-engineering database import policy ted2nlri_epe_stat
set protocols mpls traffic-engineering database export l3-unicast-topology
set protocols mpls admin-groups red 0
set protocols mpls admin-groups blue 1
set protocols mpls admin-groups green 2
set protocols mpls admin-groups yellow 3
```

```
set protocols mpls admin-groups orange 4
set protocols mpls admin-groups brown 5
set protocols mpls admin-groups black 6
set protocols mpls admin-groups pink 7
set protocols mpls label-range static-label-range 1000 70000
set protocols mpls interface ge-0/0/0.0 admin-group brown
set protocols mpls interface ge-0/0/1.0 admin-group green
set protocols mpls interface ge-0/0/2.0 admin-group red
set protocols mpls interface ge-0/0/3.0 admin-group [ red brown ]
set protocols mpls interface all
set protocols mpls interface fxp0.0 disableset protocols source-packet-routing segment-list R3-
R2-R1 inherit-label-nexthops
set protocols source-packet-routing segment-list R3-R2-R1 auto-translate
set protocols source-packet-routing segment-list R3-R2-R1 hop1 ip-address 192.168.6.1
set protocols source-packet-routing segment-list R3-R2-R1 hop2 ip-address 192.168.3.1
set protocols source-packet-routing source-routing-path lsp3to1_sr to 10.1.1.1
set protocols source-packet-routing source-routing-path lsp3to1_sr primary R3-R2-R1
```

Configure the following additional commands in **Device R3 (BN3 router)** for colored SR-TE underlay path.

```
set routing-options rib-groups color-to-inet3 import-rib [ junos-rti-tc-<color>.inet.0 inet.3 ]
set protocols source-packet-routing no-chained-composite-next-hop
set protocols source-packet-routing source-routing-path lsp3to1_sr color 1000
set protocols source-packet-routing rib-group ipv4-color color-to-inet3
```

Device R6 (BN4 router)

```
set interfaces ge-0/0/0 description To_R0
set interfaces ge-0/0/0 vlan-tagging
set interfaces ge-0/0/0 unit 0 vlan-id 1
set interfaces ge-0/0/0 unit 0 family inet address 192.168.12.1192.168.2.2/24
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:4000:10::4/64
set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/0 unit 1 vlan-id 2
set interfaces ge-0/0/0 unit 1 family inet address 192.168.40.2/24
set interfaces ge-0/0/0 unit 1 family iso
set interfaces ge-0/0/0 unit 1 family inet6 address 2001:db8:4000:20::4/64
set interfaces ge-0/0/0 unit 1 family mpls maximum-labels 8
set interfaces ge-0/0/2 description To_R1
```

```
set interfaces ge-0/0/2 vlan-tagging
set interfaces ge-0/0/2 unit 0 vlan-id 1
set interfaces ge-0/0/2 unit 0 family inet address 192.168.4.2/24
set interfaces ge-0/0/2 unit 0 family iso
set interfaces ge-0/0/2 unit 0 family inet6 address 2001:db8:1400:10::4/64
set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/3 description To_R2
set interfaces ge-0/0/3 vlan-tagging
set interfaces ge-0/0/3 unit 0 vlan-id 1
set interfaces ge-0/0/3 unit 0 family inet address 192.168.7.1/24
set interfaces ge-0/0/3 unit 0 family iso
set interfaces ge-0/0/3 unit 0 family inet6 address 2001:db8:2400:10::4/64
set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/3 unit 1 vlan-id 2
set interfaces ge-0/0/3 unit 1 family inet address 192.168.24.1/24
set interfaces ge-0/0/3 unit 1 family iso
set interfaces ge-0/0/3 unit 1 family inet6 address 2001:db8:2400:20::4/64
set interfaces ge-0/0/3 unit 1 family mpls maximum-labels 8
set interfaces ge-0/0/4 description To_R5
set interfaces ge-0/0/4 vlan-tagging
set interfaces ge-0/0/4 unit 0 vlan-id 1
set interfaces ge-0/0/4 unit 0 family inet address 192.168.13.1/24
set interfaces ge-0/0/4 unit 0 family iso
set interfaces ge-0/0/4 unit 0 family inet6 address 2001:db8:4500:10::4/64
set interfaces ge-0/0/4 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/4 unit 1 vlan-id 2
set interfaces ge-0/0/4 unit 1 family inet address 192.168.45.1/24
set interfaces ge-0/0/4 unit 1 family iso
set interfaces ge-0/0/4 unit 1 family inet6 address 2001:db8:4500:20::4/64
set interfaces ge-0/0/4 unit 1 family mpls maximum-labels 8
set interfaces lo0 unit 0 family inet address 10.4.4.4/32
set interfaces lo0 unit 0 family iso address 49.0001.0004.0404.0400
set interfaces lo0 unit 0 family inet6 address 2001:db8:abcd::04:04:04:04/128
set policy-options policy-statement expresspolsr1 from protocol spring-te
set policy-options policy-statement expresspolsr1 from route-filter 10.6.6.6/32 exact
set policy-options policy-statement expresspolsr1 then accept
set policy-options policy-statement nlri2bgp_epe term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp_epe term 1 from protocol bgp-ls-epe
set policy-options policy-statement nlri2bgp_epe term 1 then next-hop self
set policy-options policy-statement nlri2bgp_epe term 1 then accept
set policy-options policy-statement nlri2bgp_stat term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp_stat term 1 from protocol express-segments
set policy-options policy-statement nlri2bgp_stat term 1 then accept
```

```
set policy-options policy-statement pplb then load-balance per-packet
set policy-options policy-statement prefix-sid term 1 from route-filter 10.4.4.4/32 exact
set policy-options policy-statement prefix-sid term 1 then prefix-segment index 1004
set policy-options policy-statement prefix-sid term 1 then prefix-segment node-segment
set policy-options policy-statement ted2nlri_epe_stat term 1 from family traffic-engineering
set policy-options policy-statement ted2nlri_epe_stat term 1 from protocol express-segments
set policy-options policy-statement ted2nlri_epe_stat term 1 then accept
set policy-options policy-statement ted2nlri_epe_stat term 2 from family traffic-engineering
set policy-options policy-statement ted2nlri_epe_stat term 2 from protocol bgp-ls-epe
set policy-options policy-statement ted2nlri_epe_stat term 2 then accept
set policy-options policy-statement ted2nlri_epe_stat term 3 from protocol isis
set policy-options policy-statement ted2nlri_epe_stat term 3 then reject
set routing-options router-id 10.4.4.4
set routing-options autonomous-system 65200
set routing-options forwarding-table export pplb
set protocols bgp group ibgp1 type internal
set protocols bgp group ibgp1 local-address 10.4.4.4
set protocols bgp group ibgp1 family traffic-engineering unicast
set protocols bgp group ibgp1 export nlri2bgp_epe
set protocols bgp group ibgp1 neighbor 10.2.2.2
set protocols bgp group ibgp1 neighbor 10.5.5.5
set protocols bgp group ebgp1 type external
set protocols bgp group ebgp1 family inet-vpn unicast
set protocols bgp group ebgp1 family traffic-engineering unicast
set protocols bgp group ebgp1 export nlri2bgp_stat
set protocols bgp group ebgp1 neighbor 192.168.2.1 peer-as 65100
set protocols bgp group ebgp1 neighbor 192.168.2.1 egress-te-adj-segment epe_adj1_toR0 label 8140
set protocols bgp group ebgp1 neighbor 192.168.2.1 egress-te-adj-segment epe_adj1_toR0 next-hop
192.168.2.1
set protocols bgp group ebgp1 neighbor 192.168.2.1 egress-te-adj-segment epe_adj1_toR0 te-link-
attribute te-metric 20
set protocols bgp group ebgp1 neighbor 192.168.2.1 egress-te-adj-segment epe_adj1_toR0 te-link-
attribute igp-metric 10
set protocols bgp group ebgp1 neighbor 192.168.2.1 egress-te-adj-segment epe_adj1_toR0 te-link-
attribute admin-group [ red brown ]
set protocols express-segments segment-set set4sr membership-policy expresspolsr1
set protocols express-segments traffic-engineering
set protocols isis interface ge-0/0/0.0 level 2 post-convergence-lfa node-protection
set protocols isis interface ge-0/0/0.0 passive
set protocols isis interface ge-0/0/2.0 level 2 lan-neighbor 0100.0101.0101 ipv4-adjacency-
segment protected label 4111
set protocols isis interface ge-0/0/2.0 level 2 lan-neighbor 0100.0101.0101 ipv4-adjacency-
segment unprotected label 4101
```

```
set protocols isis interface ge-0/0/2.0 level 2 post-convergence-lfa node-protection
set protocols isis interface ge-0/0/3.0 level 2 lan-neighbor 0100.0202.0202 ipv4-adjacency-
segment protected label 4211
set protocols isis interface ge-0/0/3.0 level 2 lan-neighbor 0100.0202.0202 ipv4-adjacency-
segment unprotected label 4201
set protocols isis interface ge-0/0/3.0 level 2 post-convergence-lfa node-protection
set protocols isis interface ge-0/0/4.0 level 2 lan-neighbor 0100.0505.0505 ipv4-adjacency-
segment protected label 4511
set protocols isis interface ge-0/0/4.0 level 2 lan-neighbor 0100.0505.0505 ipv4-adjacency-
segment unprotected label 4501
set protocols isis interface ge-0/0/4.0 level 2 post-convergence-lfa node-protection
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0 passive
set protocols isis source-packet-routing srgb start-label 800000 index-range 50000
set protocols isis level 1 disable
set protocols isis level 2 wide-metrics-only
set protocols isis backup-spf-options use-post-convergence-lfa
set protocols isis backup-spf-options use-source-packet-routing
set protocols isis traffic-engineering l3-unicast-topology
set protocols isis traffic-engineering advertisement always
set protocols isis export prefix-sid
set protocols mpls traffic-engineering database import l3-unicast-topology bgp-link-state
set protocols mpls traffic-engineering database import policy ted2nlri_epe_stat
set protocols mpls traffic-engineering database export l3-unicast-topology
set protocols mpls admin-groups red 0
set protocols mpls admin-groups blue 1
set protocols mpls admin-groups green 2
set protocols mpls admin-groups yellow 3
set protocols mpls admin-groups orange 4
set protocols mpls admin-groups brown 5
set protocols mpls admin-groups black 6
set protocols mpls admin-groups pink 7
set protocols mpls label-range static-label-range 1000 70000
set protocols mpls interface ge-0/0/2.0 admin-group red
set protocols mpls interface ge-0/0/3.0 admin-group green
set protocols mpls interface ge-0/0/4.0 admin-group brown
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols source-packet-routing segment-list R4-R5-R6 hop1 label 4511
set protocols source-packet-routing segment-list R4-R5-R6 hop2 label 5601
set protocols source-packet-routing source-routing-path lsp4to6_sr to 10.6.6.6
set protocols source-packet-routing source-routing-path lsp4to6_sr primary R4-R5-R6
```

Configure the following additional commands in **Device R6 (BN4 router)** for colored SR-TE underlay path.

```
set routing-options rib-groups color-to-inet3 import-rib [ junos-rti-tc-<color>.inet.0 inet.3 ]
set protocols source-packet-routing no-chained-composite-next-hop
set protocols source-packet-routing source-routing-path lsp6to4_sr color 1000
set protocols source-packet-routing rib-group ipv4-color color-to-inet3
```

Device R7 (PE2 router)

```
set interfaces ge-0/0/0 description To_R3
set interfaces ge-0/0/0 vlan-tagging
set interfaces ge-0/0/0 unit 0 vlan-id 1
set interfaces ge-0/0/0 unit 0 family inet address 192.168.12.2/24
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:3700:10::7/64
set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/0 unit 1 vlan-id 2
set interfaces ge-0/0/0 unit 1 family inet address 192.168.37.2/24
set interfaces ge-0/0/0 unit 1 family iso
set interfaces ge-0/0/0 unit 1 family inet6 address 2001:db8:3700:20::7/64
set interfaces ge-0/0/0 unit 1 family mpls maximum-labels 8
set interfaces ge-0/0/1 description To_R6
set interfaces ge-0/0/1 vlan-tagging
set interfaces ge-0/0/1 unit 0 vlan-id 1
set interfaces ge-0/0/1 unit 0 family inet address 192.168.15.2/24
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:6700:10::7/64
set interfaces ge-0/0/1 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/1 unit 1 vlan-id 2
set interfaces ge-0/0/1 unit 1 family inet address 192.168.67.2/24
set interfaces ge-0/0/1 unit 1 family iso
set interfaces ge-0/0/1 unit 1 family inet6 address 2001:db8:6700:20::7/64
set interfaces ge-0/0/1 unit 1 family mpls maximum-labels 8
set interfaces lo0 unit 0 family inet address 10.7.7.7/32
set interfaces lo0 unit 0 family inet address 10.7.7.71/32
set interfaces lo0 unit 0 family iso address 49.0001.0007.0707.0700
set interfaces lo0 unit 0 family inet6 address 2001:db8:abcd::07:07:07:07/128
set interfaces lo0 unit 0 family inet6 address 2001:db8:abcd::7:7:7:71/128
set policy-options policy-statement bgplsepe_rt_2_ted term 1 from protocol bgp
set policy-options policy-statement bgplsepe_rt_2_ted term 1 then accept
```

```
set policy-options policy-statement direct from protocol direct
set policy-options policy-statement direct then accept
set policy-options policy-statement mpath then multipath-resolve
set policy-options policy-statement nlri2bgp term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp term 1 then accept
set policy-options policy-statement nlri2bgp_epe term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp_epe term 1 from protocol bgp-ls-epe
set policy-options policy-statement nlri2bgp_epe term 1 then next-hop self
set policy-options policy-statement nlri2bgp_epe term 1 then accept
set policy-options policy-statement nlri2ted_bgp term 1 from protocol bgp
set policy-options policy-statement nlri2ted_bgp term 1 then accept
set policy-options policy-statement nlri2ted_igp term 1 from traffic-engineering protocol isis-
level-2
set policy-options policy-statement nlri2ted_igp term 1 then accept
set policy-options policy-statement payload_vpn_109 term 1 from route-filter 10.109.0.1/16
orlonger
set policy-options policy-statement payload_vpn_109 term 1 then community add color7000
set policy-options policy-statement payload_vpn_109 term 1 then next-hop 10.7.7.7
set policy-options policy-statement payload_vpn_109 term 1 then accept
set policy-options policy-statement payload_vpn_110 term 1 from route-filter 10.110.0.1/16
orlonger
set policy-options policy-statement payload_vpn_110 term 1 then community add color7001
set policy-options policy-statement payload_vpn_110 term 1 then next-hop 10.7.7.7
set policy-options policy-statement payload_vpn_110 term 1 then accept
set policy-options policy-statement pplb then load-balance per-packet
set policy-options policy-statement ted2nlri term 1 from protocol bgp-ls-epe
set policy-options policy-statement ted2nlri term 1 then accept
set policy-options community color7000 members color:0:7000
set policy-options community color7001 members color:0:7001
set policy-options resolution-map map1 mode ip-color
set routing-options router-id 10.7.7.7
set routing-options autonomous-system 65300
set routing-options static route 10.100.100.101/32 next-hop 10.100.100.100
set routing-options static route 10.100.100.101/32 resolve
set routing-options forwarding-table export pplb
set protocols bgp group ebgp1 type external
set protocols bgp group ebgp1 multihop ttl 100
set protocols bgp group ebgp1 family inet unicast
set protocols bgp group ebgp1 family inet-vpn unicast
set protocols bgp group ebgp1 family traffic-engineering unicast
set protocols bgp group ebgp1 export [ nlri2bgp_epe payload_vpn_109 payload_vpn_110 ]
set protocols bgp group ebgp1 neighbor 192.168.12.1 peer-as 200
set protocols bgp group ebgp1 neighbor 192.168.12.1 egress-te-adj-segment epe_adj1_toR3 label
```

```
8173
set protocols bgp group ebgp1 neighbor 192.168.12.1 egress-te-adj-segment epe_adj1_toR3 next-hop
192.168.12.1
set protocols bgp group ebgp1 neighbor 192.168.12.1 egress-te-adj-segment epe_adj1_toR3 te-link-
attribute te-metric 20
set protocols bgp group ebgp1 neighbor 192.168.12.1 egress-te-adj-segment epe_adj1_toR3 te-link-
attribute igp-metric 10
set protocols bgp group ebgp1 neighbor 192.168.12.1 egress-te-adj-segment epe_adj1_toR3 te-link-
attribute admin-group [ red brown ]
set protocols bgp group ebgp1 neighbor 192.168.15.1 peer-as 200
set protocols bgp group ebgp1 neighbor 192.168.15.1 egress-te-adj-segment epe_adj1_toR6 label
8176
set protocols bgp group ebgp1 neighbor 192.168.15.1 egress-te-adj-segment epe_adj1_toR6 next-hop
192.168.15.1
set protocols bgp group ebgp1 neighbor 192.168.15.1 egress-te-adj-segment epe_adj1_toR6 te-link-
attribute te-metric 20
set protocols bgp group ebgp1 neighbor 192.168.15.1 egress-te-adj-segment epe_adj1_toR6 te-link-
attribute igp-metric 10
set protocols bgp group ebgp1 neighbor 192.168.15.1 egress-te-adj-segment epe_adj1_toR6 te-link-
attribute admin-group [ red brown ]
set protocols bgp group ebgp1 neighbor 10.100.100.101 local-address 10.7.7.71
set protocols bgp group ebgp1 neighbor 10.100.100.101 peer-as 65100
set protocols bgp group ebgp1 vpn-apply-export
set protocols bgp group to-CE1 type external
set protocols bgp group to-CE1 local-address 192.168.50.1
set protocols bgp group to-CE1 neighbor 192.168.50.2 family inet unicast
set protocols bgp group to-CE1 neighbor 192.168.50.2 family inet6 unicast
set protocols bgp group to-CE1 neighbor 192.168.50.2 peer-as 700
set protocols bgp group to-CE1 neighbor 192.168.50.2 local-as 300
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0 passive
set protocols isis level 1 disable
set protocols isis level 2 wide-metrics-only
set protocols isis traffic-engineering l3-unicast-topology
set protocols isis traffic-engineering advertisement always
set protocols mpls traffic-engineering database import l3-unicast-topology bgp-link-state
set protocols mpls traffic-engineering database import policy ted2nlri
set protocols mpls traffic-engineering database export policy nlri2ted_bgp
set protocols mpls traffic-engineering database export l3-unicast-topology
set protocols mpls admin-groups red 0
set protocols mpls admin-groups blue 1
set protocols mpls admin-groups green 2
set protocols mpls admin-groups yellow 3
```

```
set protocols mpls admin-groups orange 4
set protocols mpls admin-groups brown 5
set protocols mpls admin-groups black 6
set protocols mpls admin-groups pink 7
set protocols mpls label-range static-label-range 1000 70000
set protocols mpls interface ge-0/0/1.0 admin-group [ red brown ]
set protocols mpls interface ge-0/0/0.0 admin-group [ red brown ]
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols source-packet-routing compute-profile compute1 no-label-stack-compression
set protocols source-packet-routing source-routing-path computelsp1 to 10.100.100.100
set protocols source-packet-routing source-routing-path computelsp1 install 10.100.100.101
set protocols source-packet-routing source-routing-path computelsp1 primary p1 compute compute1
```

Configure R0 (PE1 router)

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the CLI User Guide.

To configure device R0:

1. Configure the network services mode as Enhanced IP. Enhanced IP sets the router's network services to enhanced Internet Protocol and uses enhanced mode capabilities.

   ```
   [edit]
   user@R0#set chassis network-services enhanced-ip
   ```

   After you configure the `enhanced-ip` statement and commit the configuration, the following warning message appears prompting you to reboot the router:

   ```
   'chassis'
   WARNING: Chassis configuration for network services has been changed. A system reboot is
   mandatory. Please reboot the system NOW. Continuing without a reboot might result in
   unexpected system behavior.
   commit complete
   ```

   The reboot brings up the FPCs on the router.

2. Configure the interfaces to enable IP, MPLS, and ISO transport.

```
[edit]
user@R0#set interfaces ge-0/0/0 description To_R1_1
user@R0#set interfaces ge-0/0/0 vlan-tagging
user@R0#set interfaces ge-0/0/0 unit 0 vlan-id 1
user@R0#set interfaces ge-0/0/0 unit 0 family inet address 192.168.1.1/24
user@R0#set interfaces ge-0/0/0 unit 0 family iso
user@R0#set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:1000:10::100/64
user@R0#set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 8
user@R0#set interfaces ge-0/0/2 description To_R4_1
user@R0#set interfaces ge-0/0/2 vlan-tagging
user@R0#set interfaces ge-0/0/2 unit 0 vlan-id 1
user@R0#set interfaces ge-0/0/2 unit 0 family inet address 192.168.2.1/24
user@R0#set interfaces ge-0/0/2 unit 0 family iso
user@R0#set interfaces ge-0/0/2 unit 0 family inet6 address 2001:db8:4000:10::100/64
user@R0#set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
```

3. Configure the loopback interface to enable tunnel endpoints and service endpoints.

```
[edit]
user@R0#set interfaces lo0 unit 0 family inet address 10.100.100.100/32
user@R0#set interfaces lo0 unit 0 family inet address 10.100.100.101/32
user@R0#set interfaces lo0 unit 0 family iso address 49.0001.000a.0a0a.0a00
user@R0#set interfaces lo0 unit 0 family inet6 address 2001:db8:abcd::10:100:100:100/128
```

4. Define import and export policies. For example, configure policies that export EPE TE links from the local TE database to **lsdist.0** and policies to import from **lsdist.0** into the local TE database. You can configure policies to advertise the BGP routes to a peer.

   Route filter routs are advertised from external AS.

```
[edit]
user@R0#set policy-options policy-statement bgplsepe_rt_2_ted term 1 from protocol bgp
user@R0#set policy-options policy-statement bgplsepe_rt_2_ted term 1 then accept
user@R0#set policy-options policy-statement direct from protocol direct
user@R0#set policy-options policy-statement direct then accept
user@R0#set policy-options policy-statement mpath then multipath-resolve
user@R0#set policy-options policy-statement nlri2bgp term 1 from family traffic-engineering
user@R0#set policy-options policy-statement nlri2bgp term 1 then accept
```

```
user@R0#set policy-options policy-statement nlri2bgp_epe term 1 from family traffic-
engineering
user@R0#set policy-options policy-statement nlri2bgp_epe term 1 from protocol bgp-ls-epe
user@R0#set policy-options policy-statement nlri2bgp_epe term 1 then next-hop self
user@R0#set policy-options policy-statement nlri2bgp_epe term 1 then accept
user@R0#set policy-options policy-statement nlri2ted_bgp term 1 from protocol bgp
user@R0#set policy-options policy-statement nlri2ted_bgp term 1 then accept
user@R0#set policy-options policy-statement nlri2ted_igp term 1 from traffic-engineering
protocol isis-level-2
user@R0#set policy-options policy-statement nlri2ted_igp term 1 then accept
user@R0#set policy-options policy-statement pplb then load-balance per-packet
user@R0#set policy-options policy-statement ted2nlri term 1 from protocol bgp-ls-epe
user@R0#set policy-options policy-statement ted2nlri term 1 then accept
user@R0#set policy-options policy-statement vpn1_res_map1 from route-filter 10.1090.1/16
orlonger
user@R0#set policy-options policy-statement vpn1_res_map1 then accept
user@R0#set policy-options policy-statement vpn1_res_map1 then resolution-map map1
user@R0#set policy-options policy-statement vpn2_res_map1 from route-filter 10.110.0.1/16
orlonger
user@R0#set policy-options policy-statement vpn2_res_map1 then accept
user@R0#set policy-options policy-statement vpn2_res_map1 then resolution-map map1
```

5. Configure policy-options of community to add color attributes and set resolution map.

```
[edit]
user@R0#set policy-options community color7000 members color:0:7000
user@R0#set policy-options community color7001 members color:0:7001
user@R0#set policy-options resolution-map map1 mode ip-color
```

6. Configure routing options to identify the router in the domain.

```
[edit]
user@R0#set routing-options router-id 100.100.100.100
user@R0#set routing-options autonomous-system 100
user@R0#set routing-options static route 10.7.7.71/32 next-hop 10.7.7.7
user@R0#set routing-options static route 10.7.7.71/32 resolve
```

7. Configure BGP to enable BGP-LS route advertisement to the connected peers and define the EPE links. Since express segment is an internal TE link, this configuration creates an external TE link.

```
[edit]
user@R0#set protocols bgp group ebgp1 type external
user@R0#set protocols bgp group ebgp1 multihop ttl 100
user@R0#set protocols bgp group ebgp1 family inet unicast
user@R0#set protocols bgp group ebgp1 family inet-vpn unicast
user@R0#set protocols bgp group ebgp1 family traffic-engineering unicast
user@R0#set protocols bgp group ebgp1 export nlri2bgp_epe
user@R0#set protocols bgp group ebgp1 neighbor 192.168.1.2 peer-as 65200
user@R0#set protocols bgp group ebgp1 neighbor 192.168.1.2 egress-te-adj-segment
epe_adj1_toR1 label 7101
user@R0#set protocols bgp group ebgp1 neighbor 192.168.1.2 egress-te-adj-segment
epe_adj1_toR1 next-hop 192.168.1.2
user@R0#set protocols bgp group ebgp1 neighbor 192.168.1.2 egress-te-adj-segment
epe_adj1_toR1 te-link-attribute te-metric 20
user@R0#set protocols bgp group ebgp1 neighbor 192.168.1.2 egress-te-adj-segment
epe_adj1_toR1 te-link-attribute igp-metric 10
user@R0#set protocols bgp group ebgp1 neighbor 192.168.1.2 egress-te-adj-segment
epe_adj1_toR1 te-link-attribute admin-group red
user@R0#set protocols bgp group ebgp1 neighbor 192.168.1.2 egress-te-adj-segment
epe_adj1_toR1 te-link-attribute admin-group brown
user@R0#set protocols bgp group ebgp1 neighbor 192.168.2.2 peer-as 65200
user@R0#set protocols bgp group ebgp1 neighbor 192.168.2.2 egress-te-adj-segment
epe_adj1_toR4 label 7104
user@R0#set protocols bgp group ebgp1 neighbor 192.168.2.2 egress-te-adj-segment
epe_adj1_toR4 next-hop 192.168.2.2
user@R0#set protocols bgp group ebgp1 neighbor 192.168.2.2 egress-te-adj-segment
epe_adj1_toR4 te-link-attribute te-metric 20
user@R0#set protocols bgp group ebgp1 neighbor 192.168.2.2 egress-te-adj-segment
epe_adj1_toR4 te-link-attribute igp-metric 10
user@R0#set protocols bgp group ebgp1 neighbor 192.168.2.2 egress-te-adj-segment
epe_adj1_toR4 te-link-attribute admin-group red
user@R0#set protocols bgp group ebgp1 neighbor 192.168.2.2 egress-te-adj-segment
epe_adj1_toR4 te-link-attribute admin-group brown
user@R0#set protocols bgp group ebgp1 neighbor 10.7.7.71 local-address 10.100.100.101
user@R0#set protocols bgp group ebgp1 neighbor 10.7.7.71 import [ vpn1_res_map1
vpn2_res_map1 ]
user@R0#set protocols bgp group ebgp1 neighbor 10.7.7.71 peer-as 65300
user@R0#set protocols bgp group ebgp1 vpn-apply-export
```

8. Configure IS-IS protocol.

```
[edit]
user@R0#set protocols isis interface lo0.0 passive
user@R0#set protocols isis level 1 disable
user@R0#set protocols isis level 2 wide-metrics-only
user@R0#set protocols isis traffic-engineering l3-unicast-topology
user@R0#set protocols isis traffic-engineering advertisement always
```

9. Enable import and export of traffic engineering database parameters using policies.

```
[edit]
user@R0#set protocols mpls traffic-engineering database import l3-unicast-topology bgp-link-
state
user@R0#set protocols mpls traffic-engineering database import policy ted2nlri
user@R0#set protocols mpls traffic-engineering database export policy nlri2ted_bgp
user@R0#set protocols mpls traffic-engineering database export l3-unicast-topology
```

10. Configure MPLS administrative group policies for LSP path computation.

```
[edit]
user@R0#set protocols mpls admin-groups red 0
user@R0#set protocols mpls admin-groups blue 1
user@R0#set protocols mpls admin-groups green 2
user@R0#set protocols mpls admin-groups yellow 3
user@R0#set protocols mpls admin-groups orange 4
user@R0#set protocols mpls admin-groups brown 5
user@R0#set protocols mpls admin-groups black 6
user@R0#set protocols mpls admin-groups pink 7
```

11. Configure the MPLS label range to assign static labels for the EPE links.

```
[edit]
user@R0#set protocols mpls label-range static-label-range 1000 70000
```

12. Configure MPLS on the interfaces.

```
[edit]
user@R0#set protocols mpls interface all
user@R0#set protocols mpls interface fxp0.0 disable
user@R0#set protocols mpls interface ge-0/0/0.0 admin-group [ red brown ]
```

13. Configure SR-TE policies on the ingress router to enable end-to-end SR-TE policy.

```
[edit]
user@R0#set protocols source-packet-routing compute-profile compute1 no-label-stack-
compression
user@R0#set protocols source-packet-routing compute-profile ecompute1 no-label-stack-
compression
user@R0#set protocols source-packet-routing source-routing-path computelsp1 to 10.7.7.7
user@R0#set protocols source-packet-routing source-routing-path computelsp1 install
10.7.7.71
user@R0#set protocols source-packet-routing source-routing-path computelsp1 primary p1
compute compute1
user@R0#set protocols source-packet-routing source-routing-path ecomputelsp1 to 10.7.7.7
user@R0#set protocols source-packet-routing source-routing-path ecomputelsp1 color 7000
user@R0#set protocols source-packet-routing source-routing-path ecomputelsp1 primary p1
compute ecompute1
user@R0#set protocols source-packet-routing source-routing-path ecomputelsp2 to 10.7.7.7
```

Results

From configuration mode, confirm your configuration by entering the `show chassis`, `show interfaces`, `show policy-optionsshow routing-options`, and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
chassis {
    network-services enhanced-ip;
}
interfaces {
    ge-0/0/0 {
        description To_R1_1;
        vlan-tagging;
        unit 0 {
```

```
            vlan-id 1;
            family inet {
                address 192.168.1.1/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:1000:10::100/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/2 {
        description To_R4_1;
        vlan-tagging;
        unit 0 {
            vlan-id 1;
            family inet {
                address 192.168.2.1/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:4000:10::100/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.100.100.100/32;
                address 10.100.100.101/32;
            }
            family iso {
                address 49.0001.000a.0a0a.0a00;
            }
            family inet6 {
                address 2001:db8:abcd::10:100:100:100/128;
            }
        }
```

```
    }
}
policy-options {
    policy-statement bgplsepe_rt_2_ted {
        term 1 {
            from protocol bgp;
            then accept;
        }
    }
    policy-statement direct {
        from protocol direct;
        then accept;
    }
    policy-statement mpath {
        then multipath-resolve;
    }
    policy-statement nlri2bgp {
        term 1 {
            from family traffic-engineering;
            then accept;
        }
    }
    policy-statement nlri2bgp_epe {
        term 1 {
            from {
                family traffic-engineering;
                protocol bgp-ls-epe;
            }
            then {
                next-hop self;
                accept;
            }
        }
    }
    policy-statement nlri2ted_bgp {
        term 1 {
            from protocol bgp;
            then accept;
        }
    }
    policy-statement nlri2ted_igp {
        term 1 {
            from {
```

```
                traffic-engineering {
                    protocol isis-level-2;
                }
            }
            then accept;
        }
    }
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
    }
    policy-statement ted2nlri {
        term 1 {
            from protocol bgp-ls-epe;
            then accept;
        }
    }
    policy-statement vpn1_res_map1 {
        from {
            route-filter 10.109.0.1/16 orlonger;
        }
        then {
            accept;
            resolution-map map1;
        }
    }
    policy-statement vpn2_res_map1 {
        from {
            route-filter 10.110.0.1/16 orlonger;
        }
        then {
            accept;
            resolution-map map1;
        }
    }
    community color7000 members color:0:7000;
    community color7001 members color:0:7001;
    resolution-map map1 {
        mode ip-color;
    }
}
routing-options {
```

```
    router-id 10.100.100.100;
    autonomous-system 65100;
    static {
        route 10.7.7.71/32 {
            next-hop 10.7.7.7;
            resolve;
        }
    }
}
protocols {
    bgp {
        group ebgp1 {
            type external;
            multihop {
                ttl 100;
            }
            family inet {
                unicast;
            }
            family inet-vpn {
                unicast;
            }
            family traffic-engineering {
                unicast;
            }
            export nlri2bgp_epe;
            neighbor 192.168.1.2 {
                peer-as 65200;
                egress-te-adj-segment epe_adj1_toR1 {
                    label 7101;
                    next-hop 192.168.1.2;
                    te-link-attribute {
                        te-metric 20;
                        igp-metric 10;
                        admin-group [ red brown ];
                    }
                }
            }
            neighbor 192.168.2.2 {
                peer-as 65200;
                egress-te-adj-segment epe_adj1_toR4 {
                    label 7104;
```

```
                next-hop 192.168.12.1 foo
                te-link-attribute {
                    te-metric 20;
                    igp-metric 10;
                    admin-group [ red brown ];
                }
            }
        }
        neighbor 10.7.7.71 {
            local-address 10.100.100.101;
            import [ vpn1_res_map1 vpn2_res_map1 ];
            peer-as 65300;
        }
        vpn-apply-export;
    }
}
isis {
    interface lo0.0 {
        passive;
    }
    level 1 disable;
    level 2 wide-metrics-only;
    traffic-engineering {
        l3-unicast-topology;
        advertisement always;
    }
}
mpls {
    traffic-engineering {
        database {
            import {
                l3-unicast-topology {
                    bgp-link-state;
                }
                policy ted2nlri;
            }
            export {
                policy nlri2ted_bgp;
                l3-unicast-topology;
            }
        }
    }
    admin-groups {
```

```
            red 0;
            blue 1;
            green 2;
            yellow 3;
            orange 4;
            brown 5;
            black 6;
            pink 7;
        }
        label-range {
            static-label-range 1000 70000;
        }
        interface all;
        interface fxp0.0 {
            disable;
        }
        interface ge-0/0/0.0 {
            admin-group [ red brown ];
        }
    }
    source-packet-routing {
        compute-profile compute1 {
            no-label-stack-compression;
        }
        compute-profile ecompute1 {
            no-label-stack-compression;
        }
        source-routing-path computelsp1 {
            to 10.7.7.7;
            install 10.7.7.71;
            primary {
                p1 {
                    compute {
                        compute1;
                    }
                }
            }
        }
        source-routing-path ecomputelsp1 {
            to 10.7.7.7;
            color 7000;
            primary {
                p1 {
```

```
                    compute {
                        ecompute1;
                    }
                }
            }
        }
    }
}
```

Configure R1 (BN1 router)

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the CLI User Guide.

To configure device R1:

1. Configure the network services mode as Enhanced IP. Enhanced IP sets the router's network services to enhanced Internet Protocol and uses enhanced mode capabilities.

   ```
   [edit]
   user@R1#set chassis network-services enhanced-ip
   ```

   After you configure the `enhanced-ip` statement and commit the configuration, the following warning message appears prompting you to reboot the router:

   ```
   'chassis'
   WARNING: Chassis configuration for network services has been changed. A system reboot is
   mandatory. Please reboot the system NOW. Continuing without a reboot might result in
   unexpected system behavior.
   commit complete
   ```

   The reboot brings up the FPCs on the router.

2. Configure the interfaces to enable IP, MPLS, and ISO transport.

   ```
   [edit]
   user@R1#set interfaces ge-0/0/0 description To_R0_1
   user@R1#set interfaces ge-0/0/0 vlan-tagging
   ```

```
user@R1#set interfaces ge-0/0/0 unit 0 vlan-id 1
user@R1#set interfaces ge-0/0/0 unit 0 family inet address 192.168.1.2/24
user@R1#set interfaces ge-0/0/0 unit 0 family iso
user@R1#set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:1000:10::1/64
user@R1#set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 8
user@R1#set interfaces ge-0/0/0 unit 1 vlan-id 2
user@R1#set interfaces ge-0/0/0 unit 1 family inet address 192.168.20.1/24
user@R1#set interfaces ge-0/0/0 unit 1 family iso
user@R1#set interfaces ge-0/0/0 unit 1 family inet6 address 2001:db8:1000:20::1/64
user@R1#set interfaces ge-0/0/0 unit 1 family mpls maximum-labels 8
user@R1#set interfaces ge-0/0/2 description To_R2
user@R1#set interfaces ge-0/0/2 vlan-tagging
user@R1#set interfaces ge-0/0/2 unit 0 vlan-id 1
user@R1#set interfaces ge-0/0/2 unit 0 family inet address 192.168.3.1/24
user@R1#set interfaces ge-0/0/2 unit 0 family iso
user@R1#set interfaces ge-0/0/2 unit 0 family inet6 address 2001:db8:1200:10::1/64
user@R1#set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
user@R1#set interfaces ge-0/0/2 unit 1 vlan-id 2
user@R1#set interfaces ge-0/0/2 unit 1 family inet address 192.168.21.1/24
user@R1#set interfaces ge-0/0/2 unit 1 family iso
user@R1#set interfaces ge-0/0/2 unit 1 family inet6 address 2001:db8:1200:20::1/64
user@R1#set interfaces ge-0/0/2 unit 1 family mpls maximum-labels 8
user@R1#set interfaces ge-0/0/3 description to-R4
user@R1#set interfaces ge-0/0/3 vlan-tagging
user@R1#set interfaces ge-0/0/3 unit 0 vlan-id 1
user@R1#set interfaces ge-0/0/3 unit 0 family inet address 192.168.4.1/24
user@R1#set interfaces ge-0/0/3 unit 0 family iso
user@R1#set interfaces ge-0/0/3 unit 0 family inet6 address 2001:db8:1400:10::1/64
user@R1#set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
user@R1#set interfaces ge-0/0/4 description to-R5
user@R1#set interfaces ge-0/0/4 vlan-tagging
user@R1#set interfaces ge-0/0/4 unit 0 vlan-id 1
user@R1#set interfaces ge-0/0/4 unit 0 family inet address 192.168.5.1/24
user@R1#set interfaces ge-0/0/4 unit 0 family iso
user@R1#set interfaces ge-0/0/4 unit 0 family inet6 address 2001:db8:1500:10::1/64
user@R1#set interfaces ge-0/0/4 unit 0 family mpls maximum-labels 8
user@R1#set interfaces ge-0/0/4 unit 1 vlan-id 2
user@R1#set interfaces ge-0/0/4 unit 1 family inet address 192.168.22.1/24
user@R1#set interfaces ge-0/0/4 unit 1 family iso
user@R1#set interfaces ge-0/0/4 unit 1 family inet6 address 2001:db8:1500:20::1/64
user@R1#set interfaces ge-0/0/4 unit 1 family mpls maximum-labels 8
```

3. Configure the loopback interface to enable tunnel endpoints and service endpoints.

```
[edit]
user@R1#set interfaces lo0 unit 0 family inet address 10.1.1.1/32
user@R1#set interfaces lo0 unit 0 family iso address 49.0001.0001.0101.0100
user@R1#set interfaces lo0 unit 0 family inet6 address 2001:db8:abcd::01:01:01:01/128
```

4. Define import and export policies. For example, configure policies that export EPE TE links from the local TE database to lsdist.0 and policies to import from lsdist.0 into the local TE database. You can configure policies to advertise the BGP routes to a peer.

```
[edit]
user@R1#set policy-options policy-statement bgplsepe_rt_2_ted term 1 from protocol bgp
user@R1#set policy-options policy-statement bgplsepe_rt_2_ted term 1 then accept
user@R1#set policy-options policy-statement expresspolsr1 from protocol spring-te
user@R1#set policy-options policy-statement expresspolsr1 from route-filter 10.3.3.3/32
exact
user@R1#set policy-options policy-statement expresspolsr1 then accept
user@R1#set policy-options policy-statement nlri2bgp term 1 from family traffic-engineering
user@R1#set policy-options policy-statement nlri2bgp term 1 then next-hop self
user@R1#set policy-options policy-statement nlri2bgp term 1 then accept
user@R1#set policy-options policy-statement nlri2bgp_epe term 1 from family traffic-
engineering
user@R1#set policy-options policy-statement nlri2bgp_epe term 1 from protocol bgp-ls-epe
user@R1#set policy-options policy-statement nlri2bgp_epe term 1 then next-hop self
user@R1#set policy-options policy-statement nlri2bgp_epe term 1 then accept
user@R1#set policy-options policy-statement nlri2bgp_stat term 1 from family traffic-
engineering
user@R1#set policy-options policy-statement nlri2bgp_stat term 1 from protocol express-
segments
user@R1#set policy-options policy-statement nlri2bgp_stat term 1 then accept
user@R1#set policy-options policy-statement nlri2ted_igp term 1 from traffic-engineering
protocol isis-level-2
user@R1#set policy-options policy-statement nlri2ted_igp term 1 then accept
user@R1#set policy-options policy-statement pplb then load-balance per-packet
user@R1#set policy-options policy-statement prefix-sid term 1 from route-filter 10.1.1.1/32
exact
user@R1#set policy-options policy-statement prefix-sid term 1 then prefix-segment index 1001
user@R1#set policy-options policy-statement prefix-sid term 1 then prefix-segment node-
segment
user@R1#set policy-options policy-statement ted2nlri term 1 from protocol bgp-ls-epe
```

```
user@R1#set policy-options policy-statement ted2nlri term 1 then accept
user@R1#set policy-options policy-statement ted2nlri_epe_stat term 1 from family traffic-
engineering
user@R1#set policy-options policy-statement ted2nlri_epe_stat term 1 from protocol express-
segments
user@R1#set policy-options policy-statement ted2nlri_epe_stat term 1 then accept
user@R1#set policy-options policy-statement ted2nlri_epe_stat term 2 from family traffic-
engineering
user@R1#set policy-options policy-statement ted2nlri_epe_stat term 2 from protocol bgp-ls-
epe
user@R1#set policy-options policy-statement ted2nlri_epe_stat term 2 then accept
user@R1#set policy-options policy-statement ted2nlri_epe_stat term 3 from protocol isis
user@R1#set policy-options policy-statement ted2nlri_epe_stat term 3 then reject
user@R1#set policy-options policy-statement ted2nlri_igp term 1 from family traffic-
engineering
user@R1#set policy-options policy-statement ted2nlri_igp term 1 from protocol isis
user@R1#set policy-options policy-statement ted2nlri_igp term 1 then accept
```

5. Configure routing options to identify the router in the domain.

```
[edit]
user@R1#set routing-options router-id 10.1.1.1
user@R1#set routing-options autonomous-system 65200
```

6. Define the RIB group to copy junos-rti-tc-<color>.inet.0 to inet.3 routing table.

```
[edit]
user@R1#set routing-options rib-groups color-to-inet3 import-rib [ junos-rti-tc-
<color>.inet.0 inet.3 ]
```

7. Configure BGP to enable BGP-LS route advertisement to the connected peers and define the EPE links. Since express segment is an internal TE link, this configuration creates an external TE link.

```
[edit]
user@R1#set protocols bgp group ebgp1 type external
user@R1#set protocols bgp group ebgp1 family inet-vpn unicast
user@R1#set protocols bgp group ebgp1 family traffic-engineering unicast
user@R1#set protocols bgp group ebgp1 export nlri2bgp_stat
user@R1#set protocols bgp group ebgp1 neighbor 192.168.1.1 peer-as 65100
user@R1#set protocols bgp group ebgp1 neighbor 192.168.1.1 egress-te-adj-segment
```

```
epe_adj1_toR0 label 8110
user@R1#set protocols bgp group ebgp1 neighbor 192.168.1.1 egress-te-adj-segment
epe_adj1_toR0 next-hop 192.168.1.1
user@R1#set protocols bgp group ebgp1 neighbor 192.168.1.1 egress-te-adj-segment
epe_adj1_toR0 te-link-attribute te-metric 20
user@R1#set protocols bgp group ebgp1 neighbor 192.168.1.1 egress-te-adj-segment
epe_adj1_toR0 te-link-attribute igp-metric 10
user@R1#set protocols bgp group ebgp1 neighbor 192.168.1.1 egress-te-adj-segment
epe_adj1_toR0 te-link-attribute admin-group red
user@R1#set protocols bgp group ebgp1 neighbor 192.168.1.1 egress-te-adj-segment
epe_adj1_toR0 te-link-attribute admin-group brown
user@R1#set protocols bgp group ibgp1 type internal
user@R1#set protocols bgp group ibgp1 local-address 10.1.1.1
user@R1#set protocols bgp group ibgp1 family traffic-engineering unicast
user@R1#set protocols bgp group ibgp1 export nlri2bgp_epe
user@R1#set protocols bgp group ibgp1 neighbor 10.2.2.2
user@R1#set protocols bgp group ibgp1 neighbor 10.5.5.5
```

8. Configure the express segment set and traffic engineering.

```
[edit]
user@R1#set protocols express-segments segment-set membership-policy expresspol1

user@R1#set protocols express-segments traffic-engineering
```

9. Configure IS-IS protocol on the interfaces.

```
[edit]
user@R1#set protocols isis interface ge-0/0/0.0 passive
user@R1#set protocols isis interface ge-0/0/1.0 passive
user@R1#set protocols isis interface ge-0/0/2.0 level 2 lan-neighbor 0100.0202.0202 ipv4-
adjacency-segment protected label 1211
user@R1#set protocols isis interface ge-0/0/2.0 level 2 lan-neighbor 0100.0202.0202 ipv4-
adjacency-segment unprotected label 1201
user@R1#set protocols isis interface ge-0/0/2.0 level 2 post-convergence-lfa node-protection
user@R1#set protocols isis interface ge-0/0/2.1 level 2 lan-neighbor 0100.0202.0202 ipv4-
adjacency-segment protected label 1212
user@R1#set protocols isis interface ge-0/0/2.1 level 2 lan-neighbor 0100.0202.0202 ipv4-
adjacency-segment unprotected label 1202
user@R1#set protocols isis interface ge-0/0/2.1 level 2 post-convergence-lfa node-protection
user@R1#set protocols isis interface ge-0/0/3.0 level 2 lan-neighbor 0100.0404.0404 ipv4-
```

```
adjacency-segment protected label 1411
user@R1#set protocols isis interface ge-0/0/3.0 level 2 lan-neighbor 0100.0404.0404 ipv4-
adjacency-segment unprotected label 1401
user@R1#set protocols isis interface ge-0/0/3.0 level 2 post-convergence-lfa node-protection
user@R1#set protocols isis interface ge-0/0/4.0 level 2 lan-neighbor 0100.0505.0505 ipv4-
adjacency-segment protected label 1511
user@R1#set protocols isis interface ge-0/0/4.0 level 2 lan-neighbor 0100.0505.0505 ipv4-
adjacency-segment unprotected label 1501
user@R1#set protocols isis interface ge-0/0/4.0 level 2 post-convergence-lfa node-protection
user@R1#set protocols isis interface fxp0.0 disable
user@R1#set protocols isis interface lo0.0 passive
user@R1#set protocols isis source-packet-routing srgb start-label 800000 index-range 50000
user@R1#set protocols isis level 1 disable
user@R1#set protocols isis level 2 wide-metrics-only
user@R1#set protocols isis backup-spf-options use-post-convergence-lfa
user@R1#set protocols isis backup-spf-options use-source-packet-routing
user@R1#set protocols isis traffic-engineering l3-unicast-topology
user@R1#set protocols isis traffic-engineering advertisement always
user@R1#set protocols isis export prefix-sid
```

10. Enable import and export of traffic engineering database parameters using the policies.

```
[edit]
user@R1#set protocols mpls traffic-engineering database import l3-unicast-topology bgp-link-
state
user@R1#set protocols mpls traffic-engineering database import policy ted2nlri_epe_stat
user@R1#set protocols mpls traffic-engineering database export l3-unicast-topology
```

11. Configure MPLS administrative group policies for LSP path computation.

```
[edit]
user@R1#set protocols mpls admin-groups red 0
user@R1#set protocols mpls admin-groups blue 1
user@R1#set protocols mpls admin-groups green 2
user@R1#set protocols mpls admin-groups yellow 3
user@R1#set protocols mpls admin-groups orange 4
user@R1#set protocols mpls admin-groups brown 5
user@R1#set protocols mpls admin-groups black 6
user@R1#set protocols mpls admin-groups pink 7
```

12. Configure MPLS with interface and include administrative groups.

```
[edit]
user@R1#set protocols mpls label-range static-label-range 1000 70000
user@R1#set protocols mpls interface ge-0/0/3.0 admin-group red
user@R1#set protocols mpls interface ge-0/0/2.0 admin-group brown
user@R1#set protocols mpls interface ge-0/0/2.1 admin-group yellow
user@R1#set protocols mpls interface ge-0/0/4.0 admin-group blue
user@R1#set protocols mpls interface all
user@R1#set protocols mpls interface fxp0.0 disable
```

13. Configure ST-TE LSP from R1 device to R3 device.

```
[edit]
user@R1#set protocols source-packet-routing no-chained-composite-next-hop
user@R1#set protocols source-packet-routing segment-list R1-R2-R3 hop1 label 1211
user@R1#set protocols source-packet-routing segment-list R1-R2-R3 hop2 label 801003
user@R1#set protocols source-packet-routing source-routing-path lsp1to3_sr to 10.3.3.3
user@R1#set protocols source-packet-routing source-routing-path lsp1to3_sr color 1000
user@R1#set protocols source-packet-routing source-routing-path lsp1to3_sr primary R1-R2-R3
user@R1#set protocols source-packet-routing rib-group ipv4-color color-to-inet3
```

Results

From configuration mode, confirm your configuration by entering the `show chassis`, `show interfaces`, `show policy-options`, `show routing-options`, and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

The following result includes colored SR-TE underlay path configuration also.

```
chassis {
    network-services enhanced-ip;
}
interfaces {
    ge-0/0/0 {
        description To_R0_1;
        vlan-tagging;
        unit 0 {
            vlan-id 1;
            family inet {
```

```
                address 192.168.1.2/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:1000:10::1/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
        unit 1 {
            vlan-id 2;
            family inet {
                address 192.168.20.1/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:1000:20::1/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/2 {
        description To_R2;
        vlan-tagging;
        unit 0 {
            vlan-id 1;
            family inet {
                address 192.168.3.1/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:1200:10::1/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
        unit 1 {
            vlan-id 2;
            family inet {
```

```
                address 192.168.21.1/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:1200:20::1/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/3 {
        description to-R4;
        vlan-tagging;
        unit 0 {
            vlan-id 1;
            family inet {
                address 192.168.4.1/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:1400:10::1/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/4 {
        description to-R5;
        vlan-tagging;
        unit 0 {
            vlan-id 1;
            family inet {
                address 192.168.5.1/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:1500:10::1/64;
            }
            family mpls {
                maximum-labels 8;
            }
```

```
        }
        unit 1 {
            vlan-id 2;
            family inet {
                address 192.168.22.1/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:1500:20::1/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.1.1.1/32;
            }
            family iso {
                address 49.0001.0001.0101.0100;
            }
            family inet6 {
                address 2001:db8:abcd::01:01:01:01/128;
            }
        }
    }
}
policy-options {
    policy-statement bgplsepe_rt_2_ted {
        term 1 {
            from protocol bgp;
            then accept;
        }
    }
    policy-statement expresspolsr1 {
        from {
            protocol spring-te;
            route-filter 10.3.3.3/32 exact;
        }
        then accept;
    }
```

```
    policy-statement nlri2bgp {
        term 1 {
            from family traffic-engineering;
            then {
                next-hop self;
                accept;
            }
        }
    }
    policy-statement nlri2bgp_epe {
        term 1 {
            from {
                family traffic-engineering;
                protocol bgp-ls-epe;
            }
            then {
                next-hop self;
                accept;
            }
        }
    }
    policy-statement nlri2bgp_stat {
        term 1 {
            from {
                family traffic-engineering;
                protocol express-segments;
            }
            then accept;
        }
    }
    policy-statement nlri2ted_igp {
        term 1 {
            from {
                traffic-engineering {
                    protocol isis-level-2;
                }
            }
            then accept;
        }
    }
    policy-statement pplb {
        then {
            load-balance per-packet;
```

```
        }
    }
    policy-statement prefix-sid {
        term 1 {
            from {
                route-filter 10.1.1.1/32 exact;
            }
            then {
                prefix-segment {
                    index 1001;
                    node-segment;
                }
            }
        }
    }
    policy-statement ted2nlri {
        term 1 {
            from protocol bgp-ls-epe;
            then accept;
        }
    }
    policy-statement ted2nlri_epe_stat {
        term 1 {
            from {
                family traffic-engineering;
                protocol express-segments;
            }
            then accept;
        }
        term 2 {
            from {
                family traffic-engineering;
                protocol bgp-ls-epe;
            }
            then accept;
        }
        term 3 {
            from protocol isis;
            then reject;
        }
    }
    policy-statement ted2nlri_igp {
        term 1 {
```

```
            from {
                family traffic-engineering;
                protocol isis;
            }
            then accept;
        }
    }
}
routing-options {
    router-id 10.1.1.1;
    autonomous-system 65200;
    rib-groups {
        color-to-inet3 {
            import-rib [ junos-rti-tc-<color>.inet.0 inet.3 ];
        }
    }
    forwarding-table {
        export pplb;
    }
}
protocols {
    bgp {
        group ebgp1 {
            type external;
            family inet-vpn {
                unicast;
            }
            family traffic-engineering {
                unicast;
            }
            export nlri2bgp_stat;
            neighbor 192.168.1.1 {
                peer-as 65100;
                egress-te-adj-segment epe_adj1_toR0 {
                    label 8110;
                    next-hop 192.168.1.1;
                    te-link-attribute {
                        te-metric 20;
                        igp-metric 10;
                        admin-group [ red brown ];
                    }
                }
            }
```

```
        }
        group ibgp1 {
            type internal;
            local-address 10.1.1.1;
            family traffic-engineering {
                unicast;
            }
            export nlri2bgp_epe;
            neighbor 10.2.2.2;
            neighbor 10.5.5.5;
        }
    }
    express-segments {
        segment-set set1sr {
            membership-policy expresspolsr1;
        }
        traffic-engineering;
    }
    isis {
        interface ge-0/0/0.0 {
            passive;
        }
        interface ge-0/0/0.1 {
            passive;
        }
        interface ge-0/0/2.0 {
            level 2 {
                lan-neighbor 0100.0202.0202 {
                    ipv4-adjacency-segment {
                        protected label 1211;
                        unprotected label 1201;
                    }
                }
                post-convergence-lfa {
                    node-protection;
                }
            }
        }
        interface ge-0/0/2.1 {
            level 2 {
                lan-neighbor 0100.0202.0202 {
                    ipv4-adjacency-segment {
                        protected label 1212;
```

```
                unprotected label 1202;
            }
        }
        post-convergence-lfa {
            node-protection;
        }
    }
}
interface ge-0/0/3.0 {
    level 2 {
        lan-neighbor 0100.0404.0404 {
            ipv4-adjacency-segment {
                protected label 1411;
                unprotected label 1401;
            }
        }
        post-convergence-lfa {
            node-protection;
        }
    }
}
interface ge-0/0/4.0 {
    level 2 {
        lan-neighbor 0100.0505.0505 {
            ipv4-adjacency-segment {
                protected label 1511;
                unprotected label 1501;
            }
        }
        post-convergence-lfa {
            node-protection;
        }
    }
}
interface fxp0.0 {
    disable;
}
interface lo0.0 {
    passive;
}
source-packet-routing {
    srgb start-label 800000 index-range 50000;
}
```

```
        level 1 disable;
        level 2 wide-metrics-only;
        backup-spf-options {
            use-post-convergence-lfa;
            use-source-packet-routing;
        }
        traffic-engineering {
            l3-unicast-topology;
            advertisement always;
        }
        export prefix-sid;
    }
    mpls {
        traffic-engineering {
            database {
                import {
                    l3-unicast-topology {
                        bgp-link-state;
                    }
                    policy ted2nlri_epe_stat;
                }
                export {
                    l3-unicast-topology;
                }
            }
        }
        admin-groups {
            red 0;
            blue 1;
            green 2;
            yellow 3;
            orange 4;
            brown 5;
            black 6;
            pink 7;
        }
        label-range {
            static-label-range 1000 70000;
        }
        interface ge-0/0/3.0 {
            admin-group red;
        }
        interface ge-0/0/2.0 {
```

```
            admin-group brown;
        }
        interface ge-0/0/2.1 {
            admin-group yellow;
        }
        interface ge-0/0/4.0 {
            admin-group blue;
        }
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
    source-packet-routing {
        no-chained-composite-next-hop;
        segment-list R1-R2-R3 {
            hop1 label 1211;
            hop2 label 801003;
        }
        source-routing-path lsp1to3_sr {
            to 10.3.3.3;
            color 1000;
            primary {
                R1-R2-R3;
            }
        }
        rib-group {
            ipv4-color {
                color-to-inet3;
            }
        }
    }
}
```

Configure R4 (BN2 router)

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the CLI User Guide.

To configure device R4:

1. Configure the network services mode as Enhanced IP. Enhanced IP sets the router's network services to enhanced Internet Protocol and uses enhanced mode capabilities.

```
[edit]
user@R4#set chassis network-services enhanced-ip
```

After you configure the `enhanced-ip` statement and commit the configuration, the following warning message appears prompting you to reboot the router:

```
'chassis'
WARNING: Chassis configuration for network services has been changed. A system reboot is
mandatory. Please reboot the system NOW. Continuing without a reboot might result in
unexpected system behavior.
commit complete
```

The reboot brings up the FPCs on the router.

2. Configure the interfaces to enable IP, MPLS, and ISO transport.

```
[edit]
user@R4#set interfaces ge-0/0/0 description To_R0
user@R4#set interfaces ge-0/0/0 vlan-tagging
user@R4#set interfaces ge-0/0/0 unit 0 vlan-id 1
user@R4#set interfaces ge-0/0/0 unit 0 family inet address 192.168.12.1192.168.2.2/24
user@R4#set interfaces ge-0/0/0 unit 0 family iso
user@R4#set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:4000:10::4/64
user@R4#set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 8
user@R4#set interfaces ge-0/0/0 unit 1 vlan-id 2
user@R4#set interfaces ge-0/0/0 unit 1 family inet address 192.168.40.2/24
user@R4#set interfaces ge-0/0/0 unit 1 family iso
user@R4#set interfaces ge-0/0/0 unit 1 family inet6 address 2001:db8:4000:20::4/64
user@R4#set interfaces ge-0/0/0 unit 1 family mpls maximum-labels 8
user@R4#set interfaces ge-0/0/2 description To_R1
user@R4#set interfaces ge-0/0/2 vlan-tagging
user@R4#set interfaces ge-0/0/2 unit 0 vlan-id 1
user@R4#set interfaces ge-0/0/2 unit 0 family inet address 192.168.4.2/24
user@R4#set interfaces ge-0/0/2 unit 0 family iso
user@R4#set interfaces ge-0/0/2 unit 0 family inet6 address 2001:db8:1400:10::4/64
user@R4#set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
```

```
user@R4#set interfaces ge-0/0/3 description To_R2
user@R4#set interfaces ge-0/0/3 vlan-tagging
user@R4#set interfaces ge-0/0/3 unit 0 vlan-id 1
user@R4#set interfaces ge-0/0/3 unit 0 family inet address 192.168.7.1/24
user@R4#set interfaces ge-0/0/3 unit 0 family iso
user@R4#set interfaces ge-0/0/3 unit 0 family inet6 address 2001:db8:2400:10::4/64
user@R4#set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
user@R4#set interfaces ge-0/0/3 unit 1 vlan-id 2
user@R4#set interfaces ge-0/0/3 unit 1 family inet address 192.168.24.1/24
user@R4#set interfaces ge-0/0/3 unit 1 family iso
user@R4#set interfaces ge-0/0/3 unit 1 family inet6 address 2001:db8:2400:20::4/64
user@R4#set interfaces ge-0/0/3 unit 1 family mpls maximum-labels 8
user@R4#set interfaces ge-0/0/4 description To_R5
user@R4#set interfaces ge-0/0/4 vlan-tagging
user@R4#set interfaces ge-0/0/4 unit 0 vlan-id 1
user@R4#set interfaces ge-0/0/4 unit 0 family inet address 192.168.13.1/24
user@R4#set interfaces ge-0/0/4 unit 0 family iso
user@R4#set interfaces ge-0/0/4 unit 0 family inet6 address 2001:db8:4500:10::4/64
user@R4#set interfaces ge-0/0/4 unit 0 family mpls maximum-labels 8
user@R4#set interfaces ge-0/0/4 unit 1 vlan-id 2
user@R4#set interfaces ge-0/0/4 unit 1 family inet address 192.168.45.1/24
user@R4#set interfaces ge-0/0/4 unit 1 family iso
user@R4#set interfaces ge-0/0/4 unit 1 family inet6 address 2001:db8:4500:20::4/64
user@R4#set interfaces ge-0/0/4 unit 1 family mpls maximum-labels 8
```

3. Configure the loopback interface to enable tunnel endpoints and service endpoints.

```
[edit]
user@R4#set interfaces lo0 unit 0 family inet address 10.4.4.4/32
user@R4#set interfaces lo0 unit 0 family iso address 49.0001.0004.0404.0400
user@R4#set interfaces lo0 unit 0 family inet6 address 2001:db8:abcd::04:04:04:04/128
```

4. Define import and export policies. For example, configure policies that export EPE TE links from the local TE database to **lsdist.0** and policies to import from **lsdist.0** into the local TE database. You can configure policies to advertise the BGP routes to a peer.

```
[edit]
user@R4#set policy-options policy-statement expresspolsr1 from protocol spring-te
user@R4#set policy-options policy-statement expresspolsr1 from route-filter 10.6.6.6/32
exact
user@R4#set policy-options policy-statement expresspolsr1 then accept
```

```
user@R4#set policy-options policy-statement nlri2bgp_epe term 1 from family traffic-
engineering
user@R4#set policy-options policy-statement nlri2bgp_epe term 1 from protocol bgp-ls-epe
user@R4#set policy-options policy-statement nlri2bgp_epe term 1 then next-hop self
user@R4#set policy-options policy-statement nlri2bgp_epe term 1 then accept
user@R4#set policy-options policy-statement nlri2bgp_stat term 1 from family traffic-
engineering
user@R4#set policy-options policy-statement nlri2bgp_stat term 1 from protocol express-
segments
user@R4#set policy-options policy-statement nlri2bgp_stat term 1 then accept
user@R4#set policy-options policy-statement pplb then load-balance per-packet
user@R4#set policy-options policy-statement prefix-sid term 1 from route-filter 10.4.4.4/32
exact
user@R4#set policy-options policy-statement prefix-sid term 1 then prefix-segment index 1004
user@R4#set policy-options policy-statement prefix-sid term 1 then prefix-segment node-
segment
user@R4#set policy-options policy-statement ted2nlri_epe_stat term 1 from family traffic-
engineering
user@R4#set policy-options policy-statement ted2nlri_epe_stat term 1 from protocol express-
segments
user@R4#set policy-options policy-statement ted2nlri_epe_stat term 1 then accept
user@R4#set policy-options policy-statement ted2nlri_epe_stat term 2 from family traffic-
engineering
user@R4#set policy-options policy-statement ted2nlri_epe_stat term 2 from protocol bgp-ls-
epe
user@R4#set policy-options policy-statement ted2nlri_epe_stat term 2 then accept
user@R4#set policy-options policy-statement ted2nlri_epe_stat term 3 from protocol isis
user@R4#set policy-options policy-statement ted2nlri_epe_stat term 3 then reject
```

5. Configure routing options to identify the router in the domain.

```
[edit]
user@R4#set routing-options router-id 10.4.4.4
user@R4#set routing-options autonomous-system 65200
```

6. Define the RIB group to copy junos-rti-tc-<color>.inet.0 to inet.3 routing table.

```
[edit]
user@R4#set routing-options rib-groups color-to-inet3 import-rib [ junos-rti-tc-
<color>.inet.0 inet.3 ]
```

7. Configure BGP to enable BGP-LS route advertisement to the connected peers and define the EPE links. Since express segment is an internal TE link, this configuration creates an external TE link.

```
[edit]
user@R4#set protocols bgp group ibgp1 type internal
user@R4set protocols bgp group ibgp1 local-address 10.4.4.4
user@R4set protocols bgp group ibgp1 family traffic-engineering unicast
user@R4set protocols bgp group ibgp1 export nlri2bgp_epe
user@R4set protocols bgp group ibgp1 neighbor 10.2.2.2
user@R4set protocols bgp group ibgp1 neighbor 10.5.5.5
user@R4set protocols bgp group ebgp1 type external
user@R4set protocols bgp group ebgp1 family inet-vpn unicast
user@R4set protocols bgp group ebgp1 family traffic-engineering unicast
user@R4set protocols bgp group ebgp1 export nlri2bgp_stat
user@R4set protocols bgp group ebgp1 neighbor 192.168.2.1 peer-as 65100
user@R4set protocols bgp group ebgp1 neighbor 192.168.2.1 egress-te-adj-segment
epe_adj1_toR0 label 8140
user@R4set protocols bgp group ebgp1 neighbor 192.168.2.1 egress-te-adj-segment
epe_adj1_toR0 next-hop 192.168.2.1
user@R4set protocols bgp group ebgp1 neighbor 192.168.2.1 egress-te-adj-segment
epe_adj1_toR0 te-link-attribute te-metric 20
user@R4set protocols bgp group ebgp1 neighbor 192.168.2.1 egress-te-adj-segment
epe_adj1_toR0 te-link-attribute igp-metric 10
user@R4set protocols bgp group ebgp1 neighbor 192.168.2.1 egress-te-adj-segment
epe_adj1_toR0 te-link-attribute admin-group [ red brown ]
```

8. Configure the express segment set and traffic engineering..

```
[edit]
user@R4#set protocols express-segments segment-set set4sr membership-policy expresspolsr1
user@R4#set protocols express-segments traffic-engineering
```

9. Configure IS-IS protocol.

```
[edit]
user@R4#set protocols isis interface ge-0/0/0.0 level 2 post-convergence-lfa node-protection
user@R4#set protocols isis interface ge-0/0/0.0 passive
user@R4#set protocols isis interface ge-0/0/2.0 level 2 lan-neighbor 0100.0101.0101 ipv4-
adjacency-segment protected label 4111
user@R4#set protocols isis interface ge-0/0/2.0 level 2 lan-neighbor 0100.0101.0101 ipv4-
```

```
adjacency-segment unprotected label 4101
user@R4#set protocols isis interface ge-0/0/2.0 level 2 post-convergence-lfa node-protection
user@R4#set protocols isis interface ge-0/0/3.0 level 2 lan-neighbor 0100.0202.0202 ipv4-
adjacency-segment protected label 4211
user@R4#set protocols isis interface ge-0/0/3.0 level 2 lan-neighbor 0100.0202.0202 ipv4-
adjacency-segment unprotected label 4201
user@R4#set protocols isis interface ge-0/0/3.0 level 2 post-convergence-lfa node-protection
user@R4#set protocols isis interface ge-0/0/4.0 level 2 lan-neighbor 0100.0505.0505 ipv4-
adjacency-segment protected label 4511
user@R4#set protocols isis interface ge-0/0/4.0 level 2 lan-neighbor 0100.0505.0505 ipv4-
adjacency-segment unprotected label 4501
user@R4#set protocols isis interface ge-0/0/4.0 level 2 post-convergence-lfa node-protection
user@R4#set protocols isis interface fxp0.0 disable
user@R4#set protocols isis interface lo0.0 passive
user@R4#set protocols isis source-packet-routing srgb start-label 800000 index-range 50000
user@R4#set protocols isis level 1 disable
user@R4#set protocols isis level 2 wide-metrics-only
user@R4#set protocols isis backup-spf-options use-post-convergence-lfa
user@R4#set protocols isis backup-spf-options use-source-packet-routing
user@R4#set protocols isis traffic-engineering l3-unicast-topology
user@R4#set protocols isis traffic-engineering advertisement always
user@R4#set protocols isis export prefix-sid
```

10. Enable import and export of traffic engineering database parameters using policies.

```
[edit]
user@R4#set protocols mpls traffic-engineering database import l3-unicast-topology bgp-link-
state
user@R4#set protocols mpls traffic-engineering database import policy ted2nlri_epe_stat
user@R4#set protocols mpls traffic-engineering database export l3-unicast-topology
```

11. Configure MPLS administrative group policies for LSP path computation.

```
[edit]
user@R4#set protocols mpls admin-groups red 0
user@R4#set protocols mpls admin-groups blue 1
user@R4#set protocols mpls admin-groups green 2
user@R4#set protocols mpls admin-groups yellow 3
user@R4#set protocols mpls admin-groups orange 4
user@R4#set protocols mpls admin-groups brown 5
```

```
user@R4#set protocols mpls admin-groups black 6
user@R4#set protocols mpls admin-groups pink 7
```

12. Configure the MPLS label range to assign static labels for the EPE links.

```
[edit]
user@R4#set protocols mpls label-range static-label-range 1000 70000
```

13. Configure MPLS with interface and include administrative groups.

```
[edit]
user@R4#set protocols mpls interface ge-0/0/2.0 admin-group red
user@R4#set protocols mpls interface ge-0/0/3.0 admin-group green
user@R4#set protocols mpls interface ge-0/0/4.0 admin-group brown
user@R4#set protocols mpls interface all
user@R4#set protocols mpls interface fxp0.0 disable
```

14. Configure ST-TE LSP from R4 device to R6 device..

```
[edit]
user@R4#set protocols source-packet-routing no-chained-composite-next-hop
user@R4#set protocols source-packet-routing segment-list R4-R5-R6 hop1 label 4511
user@R4#set protocols source-packet-routing segment-list R4-R5-R6 hop2 label 5601
user@R4#set protocols source-packet-routing source-routing-path lsp4to6_sr to 10.6.6.6
user@R4#set protocols source-packet-routing source-routing-path lsp4to6_sr color 1000
user@R4#set protocols source-packet-routing source-routing-path lsp4to6_sr primary R4-R5-R6
user@R4#set protocols source-packet-routing rib-group ipv4-color color-to-inet3
```

Results

From configuration mode, confirm your configuration by entering the `show chassis`, `show interfaces`, `show policy-options`, `show routing-options`, and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

The following result includes colored SR-TE underlay path configuration also.

```
chassis {
    network-services enhanced-ip;
}
```

```
interfaces {
    ge-0/0/0 {
        description To_R0;
        vlan-tagging;
        unit 0 {
            vlan-id 1;
            family inet {
                address 192.168.2.2/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:4000:10::4/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
        unit 1 {
            vlan-id 2;
            family inet {
                address 192.168.40.2/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:4000:20::4/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/2 {
        description To_R1;
        vlan-tagging;
        unit 0 {
            vlan-id 1;
            family inet {
                address 192.168.4.2/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:1400:10::4/64;
            }
```

```
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/3 {
        description To_R2;
        vlan-tagging;
        unit 0 {
            vlan-id 1;
            family inet {
                address 192.168.7.1/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:2400:10::4/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
        unit 1 {
            vlan-id 2;
            family inet {
                address 192.168.24.1/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:2400:20::4/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/4 {
        description To_R5;
        vlan-tagging;
        unit 0 {
            vlan-id 1;
            family inet {
                address 192.168.13.1/24;
            }
```

```
            family iso;
            family inet6 {
                address 2001:db8:4500:10::4/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
        unit 1 {
            vlan-id 2;
            family inet {
                address 192.168.45.1/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:4500:20::4/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.4.4.4/32;
            }
            family iso {
                address 49.0001.0004.0404.0400;
            }
            family inet6 {
                address 2001:db8:abcd::04:04:04:04/128;
            }
        }
    }
}
policy-options {
    policy-statement expresspolsr1 {
        from {
            protocol spring-te;
            route-filter 10.6.6.6/32 exact;
        }
        then accept;
```

```
    }
    policy-statement nlri2bgp_epe {
        term 1 {
            from {
                family traffic-engineering;
                protocol bgp-ls-epe;
            }
            then {
                next-hop self;
                accept;
            }
        }
    }
    policy-statement nlri2bgp_stat {
        term 1 {
            from {
                family traffic-engineering;
                protocol express-segments;
            }
            then accept;
        }
    }
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
    }
    policy-statement prefix-sid {
        term 1 {
            from {
                route-filter 10.4.4.4/32 exact;
            }
            then {
                prefix-segment {
                    index 1004;
                    node-segment;
                }
            }
        }
    }
    policy-statement ted2nlri_epe_stat {
        term 1 {
            from {
```

```
                family traffic-engineering;
                protocol express-segments;
            }
            then accept;
        }
        term 2 {
            from {
                family traffic-engineering;
                protocol bgp-ls-epe;
            }
            then accept;
        }
        term 3 {
            from protocol isis;
            then reject;
        }
    }
}
routing-options {
    router-id 10.4.4.4;
    autonomous-system 65200;
    rib-groups {
        color-to-inet3 {
            import-rib [ junos-rti-tc-<color>.inet.0 inet.3 ];
        }
    }
    forwarding-table {
        export pplb;
    }
}
protocols {
    bgp {
        group ibgp1 {
            type internal;
            local-address 10.4.4.4;
            family traffic-engineering {
                unicast;
            }
            export nlri2bgp_epe;
            neighbor 10.2.2.2;
            neighbor 10.5.5.5;
        }
        group ebgp1 {
```

```
            type external;
            family inet-vpn {
                unicast;
            }
            family traffic-engineering {
                unicast;
            }
            export nlri2bgp_stat;
            neighbor 192.168.2.1 {
                peer-as 65100;
                egress-te-adj-segment epe_adj1_toR0 {
                    label 8140;
                    next-hop 192.168.2.1;
                    te-link-attribute {
                        te-metric 20;
                        igp-metric 10;
                        admin-group [ red brown ];
                    }
                }
            }
        }
    }
    express-segments {
        segment-set set4sr {
            membership-policy expresspolsr1;
        }
        traffic-engineering;
    }
    isis {
        interface ge-0/0/0.0 {
            level 2 {
                post-convergence-lfa {
                    node-protection;
                }
            }
            passive;
        }
        interface ge-0/0/2.0 {
            level 2 {
                lan-neighbor 0100.0101.0101 {
                    ipv4-adjacency-segment {
                        protected label 4111;
                        unprotected label 4101;
```

```
                }
            }
            post-convergence-lfa {
                node-protection;
            }
        }
    }
    interface ge-0/0/3.0 {
        level 2 {
            lan-neighbor 0100.0202.0202 {
                ipv4-adjacency-segment {
                    protected label 4211;
                    unprotected label 4201;
                }
            }
            post-convergence-lfa {
                node-protection;
            }
        }
    }
    interface ge-0/0/4.0 {
        level 2 {
            lan-neighbor 0100.0505.0505 {
                ipv4-adjacency-segment {
                    protected label 4511;
                    unprotected label 4501;
                }
            }
            post-convergence-lfa {
                node-protection;
            }
        }
    }
    interface fxp0.0 {
        disable;
    }
    interface lo0.0 {
        passive;
    }
    source-packet-routing {
        srgb start-label 800000 index-range 50000;
    }
    level 1 disable;
```

```
        level 2 wide-metrics-only;
        backup-spf-options {
            use-post-convergence-lfa;
            use-source-packet-routing;
        }
        traffic-engineering {
            l3-unicast-topology;
            advertisement always;
        }
        export prefix-sid;
    }
    mpls {
        traffic-engineering {
            database {
                import {
                    l3-unicast-topology {
                        bgp-link-state;
                    }
                    policy ted2nlri_epe_stat;
                }
                export {
                    l3-unicast-topology;
                }
            }
        }
        admin-groups {
            red 0;
            blue 1;
            green 2;
            yellow 3;
            orange 4;
            brown 5;
            black 6;
            pink 7;
        }
        label-range {
            static-label-range 1000 70000;
        }
        interface ge-0/0/2.0 {
            admin-group red;
        }
        interface ge-0/0/3.0 {
            admin-group green;
```

```
        }
        interface ge-0/0/4.0 {
            admin-group brown;
        }
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
    source-packet-routing {
        no-chained-composite-next-hop;
        segment-list R4-R5-R6 {
            hop1 label 4511;
            hop2 label 5601;
        }
        source-routing-path lsp4to6_sr {
            to 10.6.6.6;
            color 1000;
            primary {
                R4-R5-R6;
            }
        }
        rib-group {
            ipv4-color {
                color-to-inet3;
            }
        }
    }
}
```

Configure R2 (Intermediate router)

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the CLI User Guide.

To configure device R2:

1. Configure the network services mode as Enhanced IP. Enhanced IP sets the router's network services to enhanced Internet Protocol and uses enhanced mode capabilities.

```
[edit]
user@R2#set chassis network-services enhanced-ip
```

After you configure the `enhanced-ip` statement and commit the configuration, the following warning message appears prompting you to reboot the router:

```
'chassis'
WARNING: Chassis configuration for network services has been changed. A system reboot is
mandatory. Please reboot the system NOW. Continuing without a reboot might result in
unexpected system behavior.
commit complete
```

The reboot brings up the FPCs on the router.

2. Configure the interfaces to enable IP, MPLS, and ISO transport.

```
user@R2#set interfaces ge-0/0/0 description To_R1
user@R2#set interfaces ge-0/0/0 vlan-tagging
user@R2#set interfaces ge-0/0/0 unit 0 vlan-id 1
user@R2#set interfaces ge-0/0/0 unit 0 family inet address 192.168.3.2/24
user@R2#set interfaces ge-0/0/0 unit 0 family iso
user@R2#set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:1200:10::2/64
user@R2#set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 8
user@R2#set interfaces ge-0/0/0 vlan-tagging
user@R2#set interfaces ge-0/0/0 unit 1 vlan-id 2
user@R2#set interfaces ge-0/0/0 unit 1 family inet address 192.168.21.2/24
user@R2#set interfaces ge-0/0/0 unit 1 family iso
user@R2#set interfaces ge-0/0/0 unit 1 family inet6 address 2001:db8:1200:20::2/64
user@R2#set interfaces ge-0/0/0 unit 1 family mpls maximum-labels 8
user@R2#set interfaces ge-0/0/1 description To_R3
user@R2#set interfaces ge-0/0/1 vlan-tagging
user@R2#set interfaces ge-0/0/1 unit 0 vlan-id 1
user@R2#set interfaces ge-0/0/1 unit 0 family inet address 192.168.6.1/24
user@R2#set interfaces ge-0/0/1 unit 0 family iso
user@R2#set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:2300:10::2/64
user@R2#set interfaces ge-0/0/1 unit 0 family mpls maximum-labels 8
user@R2#set interfaces ge-0/0/1 unit 1 vlan-id 2
user@R2#set interfaces ge-0/0/1 unit 1 family inet address 192.168.23.1/24
```

```
user@R2#set interfaces ge-0/0/1 unit 1 family iso
user@R2#set interfaces ge-0/0/1 unit 1 family inet6 address 2001:db8:2300:20::2/64
user@R2#set interfaces ge-0/0/1 unit 1 family mpls maximum-labels 8
user@R2#set interfaces ge-0/0/2 description To_R4
user@R2#set interfaces ge-0/0/2 vlan-tagging
user@R2#set interfaces ge-0/0/2 unit 0 vlan-id 1
user@R2#set interfaces ge-0/0/2 unit 0 family inet address 192.168.7.1/24
user@R2#set interfaces ge-0/0/2 unit 0 family iso
user@R2#set interfaces ge-0/0/2 unit 0 family inet6 address 2001:db8:2400:10::2/64
user@R2#set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
user@R2#set interfaces ge-0/0/2 unit 1 vlan-id 2
user@R2#set interfaces ge-0/0/2 unit 1 family inet address 192.168.24.1/24
user@R2#set interfaces ge-0/0/2 unit 1 family iso
user@R2#set interfaces ge-0/0/2 unit 1 family inet6 address 2001:db8:2400:20::2/64
user@R2#set interfaces ge-0/0/2 unit 1 family mpls maximum-labels 8
user@R2#set interfaces ge-0/0/3 description To_R5
user@R2#set interfaces ge-0/0/3 vlan-tagging
user@R2#set interfaces ge-0/0/3 unit 0 vlan-id 1
user@R2#set interfaces ge-0/0/3 unit 0 family inet address 192.168.8.1/24
user@R2#set interfaces ge-0/0/3 unit 0 family iso
user@R2#set interfaces ge-0/0/3 unit 0 family inet6 address 2001:db8:2500:10::2/64
user@R2#set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
user@R2#set interfaces ge-0/0/4 description To_R6
user@R2#set interfaces ge-0/0/4 vlan-tagging
user@R2#set interfaces ge-0/0/4 unit 0 vlan-id 1
user@R2#set interfaces ge-0/0/4 unit 0 family inet address 192.168.9.1/24
user@R2#set interfaces ge-0/0/4 unit 0 family iso
user@R2#set interfaces ge-0/0/4 unit 0 family inet6 address 2001:db8:2600:10::2/64
user@R2#set interfaces ge-0/0/4 unit 0 family mpls maximum-labels 8
user@R2#set interfaces ge-0/0/4 unit 1 vlan-id 2
user@R2#set interfaces ge-0/0/4 unit 1 family inet address 192.168.26.1/24
user@R2#set interfaces ge-0/0/4 unit 1 family iso
user@R2#set interfaces ge-0/0/4 unit 1 family inet6 address 2001:db8:2600:20::2/64
user@R2#set interfaces ge-0/0/4 unit 1 family mpls maximum-labels 8
```

3. Configure the loopback interface to enable tunnel endpoints and service endpoints.

```
[edit]
user@R2#set interfaces lo0 unit 0 family inet address 10.2.2.2/32
user@R2#set interfaces lo0 unit 0 family iso address 49.0001.0002.0202.0200
user@R2#set interfaces lo0 unit 0 family inet6 address 2001:db8:abcd::02:02:02:02/128
```

4. Define import and export policies. For example, configure policies that export EPE TE links from the local TE database to **lsdist.0** and policies to import from **lsdist.0** into the local TE database. You can configure policies to advertise the BGP routes to a peer.

```
[edit]
user@R2#set policy-options policy-statement bgplsepe_rt_2_ted term 1 from protocol bgp
user@R2#set policy-options policy-statement bgplsepe_rt_2_ted term 1 then accept
user@R2#set policy-options policy-statement nlri2bgp term 1 from family traffic-engineering
user@R2#set policy-options policy-statement nlri2bgp term 1 then next-hop self
user@R2#set policy-options policy-statement nlri2bgp term 1 then accept
user@R2#set policy-options policy-statement nlri2bgp_igp term 1 from family traffic-
engineering
user@R2#set policy-options policy-statement nlri2bgp_igp term 1 from protocol isis
user@R2#set policy-options policy-statement nlri2bgp_igp term 1 then accept
user@R2#set policy-options policy-statement nlri2ted_igp term 1 from traffic-engineering
protocol isis-level-2
user@R2#set policy-options policy-statement nlri2ted_igp term 1 then accept
user@R2#set policy-options policy-statement pplb then load-balance per-packet
user@R2#set policy-options policy-statement prefix-sid term 1 from route-filter 10.2.2.2/32
exact
user@R2#set policy-options policy-statement prefix-sid term 1 then prefix-segment index 1002
user@R2#set policy-options policy-statement prefix-sid term 1 then prefix-segment node-
segment
user@R2#set policy-options policy-statement ted2nlri term 1 from protocol bgp-ls-epe
user@R2#set policy-options policy-statement ted2nlri term 1 then accept
user@R2#set policy-options policy-statement ted2nlri_1 term 1 from traffic-engineering
user@R2#set policy-options policy-statement ted2nlri_1 term 1 then accept
user@R2#set policy-options policy-statement ted2nlri_igp term 1 from family traffic-
engineering
user@R2#set policy-options policy-statement ted2nlri_igp term 1 from protocol isis
user@R2#set policy-options policy-statement ted2nlri_igp term 1 then accept
```

5. Configure routing options to identify the router in the domain.

```
[edit]
user@R2#set routing-options router-id 10.2.2.2
user@R2#set routing-options autonomous-system 65200
user@R2#set routing-options forwarding-table export pplb
```

6. Configure BGP to enable BGP-LS route advertisement to the connected peers.

```
[edit]
user@R2#set protocols bgp group RR1 type internal
user@R2#set protocols bgp group RR1 local-address 10.2.2.2
user@R2#set protocols bgp group RR1 family traffic-engineering unicast
user@R2#set protocols bgp group RR1 neighbor 10.1.1.1
user@R2#set protocols bgp group RR1 neighbor 10.3.3.3
user@R2#set protocols bgp group RR1 neighbor 10.6.6.6
user@R2#set protocols bgp group RR1 neighbor 10.4.4.4
user@R2#set protocols bgp cluster 10.2.2.2
```

7. Configure IS-IS protocols.

```
[edit]
user@R2#set protocols isis interface ge-0/0/0.0 level 2 lan-neighbor 0100.0101.0101 ipv4-
adjacency-segment protected label 2111
user@R2#set protocols isis interface ge-0/0/0.0 level 2 lan-neighbor 0100.0101.0101 ipv4-
adjacency-segment unprotected label 2101
user@R2#set protocols isis interface ge-0/0/0.0 level 2 post-convergence-lfa node-protection
user@R2#set protocols isis interface ge-0/0/0.1 level 2 lan-neighbor 0100.0101.0101 ipv4-
adjacency-segment protected label 2112
user@R2#set protocols isis interface ge-0/0/0.1 level 2 lan-neighbor 0100.0101.0101 ipv4-
adjacency-segment unprotected label 2102
user@R2#set protocols isis interface ge-0/0/0.1 level 2 post-convergence-lfa node-protection
user@R2#set protocols isis interface ge-0/0/1.0 level 2 lan-neighbor 0100.0303.0303 ipv4-
adjacency-segment protected label 2311
user@R2#set protocols isis interface ge-0/0/1.0 level 2 lan-neighbor 0100.0303.0303 ipv4-
adjacency-segment unprotected label 2301
user@R2#set protocols isis interface ge-0/0/1.0 level 2 post-convergence-lfa node-protection
user@R2#set protocols isis interface ge-0/0/2.0 level 2 lan-neighbor 0100.0404.0404 ipv4-
adjacency-segment protected label 2411
user@R2#set protocols isis interface ge-0/0/2.0 level 2 lan-neighbor 0100.0404.0404 ipv4-
adjacency-segment unprotected label 2401
user@R2#set protocols isis interface ge-0/0/2.0 level 2 post-convergence-lfa node-protection
user@R2#set protocols isis interface ge-0/0/3.0 level 2 lan-neighbor 0100.0505.0505 ipv4-
adjacency-segment protected label 2511
user@R2#set protocols isis interface ge-0/0/3.0 level 2 lan-neighbor 0100.0505.0505 ipv4-
adjacency-segment unprotected label 2501
user@R2#set protocols isis interface ge-0/0/3.0 level 2 post-convergence-lfa node-protection
user@R2#set protocols isis interface ge-0/0/4.0 level 2 lan-neighbor 0100.0606.0606 ipv4-
```

```
adjacency-segment protected label 2611
user@R2#set protocols isis interface ge-0/0/4.0 level 2 lan-neighbor 0100.0606.0606 ipv4-
adjacency-segment unprotected label 2601
user@R2#set protocols isis interface ge-0/0/4.0 level 2 post-convergence-lfa node-protection
user@R2#set protocols isis interface fxp0.0 disable
user@R2#set protocols isis interface lo0.0 passive
user@R2#set protocols isis source-packet-routing srgb start-label 800000 index-range 50000
user@R2#set protocols isis level 1 disable
user@R2#set protocols isis level 2 wide-metrics-only
user@R2#set protocols isis backup-spf-options use-post-convergence-lfa
user@R2#set protocols isis backup-spf-options use-source-packet-routing
user@R2#set protocols isis traffic-engineering l3-unicast-topology
user@R2#set protocols isis traffic-engineering advertisement always
user@R2#set protocols isis export prefix-sid
```

8. Configure MPLS administrative group policies for LSP path computation.

```
[edit]
user@R2#set protocols mpls admin-groups red 0
user@R2#set protocols mpls admin-groups blue 1
user@R2#set protocols mpls admin-groups green 2
user@R2#set protocols mpls admin-groups yellow 3
user@R2#set protocols mpls admin-groups orange 4
user@R2#set protocols mpls admin-groups brown 5
user@R2#set protocols mpls admin-groups black 6
user@R2#set protocols mpls admin-groups pink 7
```

9. Configure the MPLS label range to assign static labels for the EPE links.

```
[edit]
user@R2#set protocols mpls label-range static-label-range 1000 70000
```

10. Configure MPLS administrative group policies for interfaces.

```
[edit]
user@R2#set protocols mpls interface ge-0/0/0.0 admin-group brown
user@R2#set protocols mpls interface ge-0/0/0.1 admin-group yellow
user@R2#set protocols mpls interface ge-0/0/2.0 admin-group green
user@R2#set protocols mpls interface ge-0/0/3.0 admin-group red
user@R2#set protocols mpls interface ge-0/0/4.0 admin-group blue
```

```
user@R2#set protocols mpls interface ge-0/0/1.0 admin-group brown
user@R2#set protocols mpls interface all
user@R2#set protocols mpls interface fxp0.0 disable
```

Results

From configuration mode, confirm your configuration by entering the `show chassis`, `show interfaces`, `show policy-options`, `show routing-options`, and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
chassis {
    network-services enhanced-ip;
}
interfaces {
    ge-0/0/0 {
        description To_R1;
        vlan-tagging;
        unit 0 {
            vlan-id 1;
            family inet {
                address 192.168.3.2/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:1200:10::2/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
        unit 1 {
            vlan-id 2;
            family inet {
                address 192.168.21.2/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:1200:20::2/64;
            }
            family mpls {
                maximum-labels 8;
```

```
            }
        }
    }
    ge-0/0/1 {
        description To_R3;
        vlan-tagging;
        unit 0 {
            vlan-id 1;
            family inet {
                address 192.168.6.1/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:2300:10::2/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
        unit 1 {
            vlan-id 2;
            family inet {
                address 192.168.23.1/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:2300:20::2/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/2 {
        description To_R4;
        vlan-tagging;
        unit 0 {
            vlan-id 1;
            family inet {
                address 192.168.7.1/24;
            }
            family iso;
            family inet6 {
```

```
                address 2001:db8:2400:10::2/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
        unit 1 {
            vlan-id 2;
            family inet {
                address 192.168.24.1/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:2400:20::2/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/3 {
        description To_R5;
        vlan-tagging;
        unit 0 {
            vlan-id 1;
            family inet {
                address 192.168.8.1/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:2500:10::2/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/4 {
        description To_R6;
        vlan-tagging;
        unit 0 {
            vlan-id 1;
            family inet {
```

```
                address 192.168.9.1/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:2600:10::2/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
        unit 1 {
            vlan-id 2;
            family inet {
                address 192.168.26.1/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:2600:20::2/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.2.2.2/32;
            }
            family iso {
                address 49.0001.0002.0202.0200;
            }
            family inet6 {
                address 2001:db8:abcd::02:02:02:02/128;
            }
        }
    }
}
policy-options {
    policy-statement bgplsepe_rt_2_ted {
        term 1 {
            from protocol bgp;
            then accept;
```

```
        }
    }
    policy-statement nlri2bgp {
        term 1 {
            from family traffic-engineering;
            then {
                next-hop self;
                accept;
            }
        }
    }
    policy-statement nlri2bgp_igp {
        term 1 {
            from {
                family traffic-engineering;
                protocol isis;
            }
            then accept;
        }
    }
    policy-statement nlri2ted_igp {
        term 1 {
            from {
                traffic-engineering {
                    protocol isis-level-2;
                }
            }
            then accept;
        }
    }
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
    }
    policy-statement prefix-sid {
        term 1 {
            from {
                route-filter 10.2.2.2/32 exact;
            }
            then {
                prefix-segment {
                    index 1002;
```

```
                    node-segment;
                }
            }
        }
    }
    policy-statement ted2nlri {
        term 1 {
            from protocol bgp-ls-epe;
            then accept;
        }
    }
    policy-statement ted2nlri_1 {
        term 1 {
            from {
                traffic-engineering;
            }
            then accept;
        }
    }
    policy-statement ted2nlri_igp {
        term 1 {
            from {
                family traffic-engineering;
                protocol isis;
            }
            then accept;
        }
    }
}
routing-options {
    router-id 10.2.2.2;
    autonomous-system 65200;
    forwarding-table {
        export pplb;
    }
}
protocols {
    bgp {
        group RR1 {
            type internal;
            local-address 10.2.2.2;
            family traffic-engineering {
                unicast;
```

```
            }
            neighbor 10.1.1.1;
            neighbor 10.3.3.3;
            neighbor 10.6.6.6;
            neighbor 10.4.4.4;
        }
        cluster 10.2.2.2;
    }
    isis {
        interface ge-0/0/0.0 {
            level 2 {
                lan-neighbor 0100.0101.0101 {
                    ipv4-adjacency-segment {
                        protected label 2111;
                        unprotected label 2101;
                    }
                }
                post-convergence-lfa {
                    node-protection;
                }
            }
        }
        interface ge-0/0/0.1 {
            level 2 {
                lan-neighbor 0100.0101.0101 {
                    ipv4-adjacency-segment {
                        protected label 2112;
                        unprotected label 2102;
                    }
                }
                post-convergence-lfa {
                    node-protection;
                }
            }
        }
        interface ge-0/0/1.0 {
            level 2 {
                lan-neighbor 0100.0303.0303 {
                    ipv4-adjacency-segment {
                        protected label 2311;
                        unprotected label 2301;
                    }
                }
```

```
            post-convergence-lfa {
                node-protection;
            }
        }
    }
    interface ge-0/0/2.0 {
        level 2 {
            lan-neighbor 0100.0404.0404 {
                ipv4-adjacency-segment {
                    protected label 2411;
                    unprotected label 2401;
                }
            }
            post-convergence-lfa {
                node-protection;
            }
        }
    }
    interface ge-0/0/3.0 {
        level 2 {
            lan-neighbor 0100.0505.0505 {
                ipv4-adjacency-segment {
                    protected label 2511;
                    unprotected label 2501;
                }
            }
            post-convergence-lfa {
                node-protection;
            }
        }
    }
    interface ge-0/0/4.0 {
        level 2 {
            lan-neighbor 0100.0606.0606 {
                ipv4-adjacency-segment {
                    protected label 2611;
                    unprotected label 2601;
                }
            }
            post-convergence-lfa {
                node-protection;
            }
        }
```

```
        }
        interface fxp0.0 {
            disable;
        }
        interface lo0.0 {
            passive;
        }
        source-packet-routing {
            srgb start-label 800000 index-range 50000;
        }
        level 1 disable;
        level 2 wide-metrics-only;
        backup-spf-options {
            use-post-convergence-lfa;
            use-source-packet-routing;
        }
        traffic-engineering {
            l3-unicast-topology;
            advertisement always;
        }
        export prefix-sid;
    }
    mpls {
        admin-groups {
            red 0;
            blue 1;
            green 2;
            yellow 3;
            orange 4;
            brown 5;
            black 6;
            pink 7;
        }
        label-range {
            static-label-range 1000 70000;
        }
        interface ge-0/0/0.0 {
            admin-group brown;
        }
        interface ge-0/0/0.1 {
            admin-group yellow;
        }
        interface ge-0/0/2.0 {
```

```
            admin-group green;
        }
        interface ge-0/0/3.0 {
            admin-group red;
        }
        interface ge-0/0/4.0 {
            admin-group blue;
        }
        interface ge-0/0/1.0 {
            admin-group brown;
        }
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
}
```

Configure R5 (Intermediate router)

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the CLI User Guide.

To configure device R5:

1. Configure the network services mode as Enhanced IP. Enhanced IP sets the router's network services to enhanced Internet Protocol and uses enhanced mode capabilities.

```
[edit]
user@R5#set chassis network-services enhanced-ip
```

After you configure the `enhanced-ip` statement and commit the configuration, the following warning message appears prompting you to reboot the router:

```
'chassis'
WARNING: Chassis configuration for network services has been changed. A system reboot is
mandatory. Please reboot the system NOW. Continuing without a reboot might result in
```

```
unexpected system behavior.
commit complete
```

The reboot brings up the FPCs on the router.

2. Configure the interfaces to enable IP, MPLS, and ISO transport.

```
user@R5#set interfaces ge-0/0/0 description To_R1
user@R5#set interfaces ge-0/0/0 vlan-tagging
user@R5#set interfaces ge-0/0/0 unit 0 vlan-id 1
user@R5#set interfaces ge-0/0/0 unit 0 family inet address 192.168.5.2/24
user@R5#set interfaces ge-0/0/0 unit 0 family iso
user@R5#set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:1500:10::5/64
user@R5#set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 8
user@R5#set interfaces ge-0/0/0 unit 1 vlan-id 2
user@R5#set interfaces ge-0/0/0 unit 1 family inet address 192.168.22.2/24
user@R5#set interfaces ge-0/0/0 unit 1 family iso
user@R5#set interfaces ge-0/0/0 unit 1 family inet6 address 2001:db8:1500:20::5/64
user@R5#set interfaces ge-0/0/0 unit 1 family mpls maximum-labels 8
user@R5#set interfaces ge-0/0/1 description To_R2
user@R5#set interfaces ge-0/0/1 vlan-tagging
user@R5#set interfaces ge-0/0/1 unit 0 vlan-id 1
user@R5#set interfaces ge-0/0/1 unit 0 family inet address 192.168.8.2/24
user@R5#set interfaces ge-0/0/1 unit 0 family iso
user@R5#set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:2500:10::5/64
user@R5#set interfaces ge-0/0/1 unit 0 family mpls maximum-labels 8
user@R5#set interfaces ge-0/0/2 description To_R3
user@R5#set interfaces ge-0/0/2 vlan-tagging
user@R5#set interfaces ge-0/0/2 unit 0 vlan-id 1
user@R5#set interfaces ge-0/0/2 unit 0 family inet address 192.168.10.2/24
user@R5#set interfaces ge-0/0/2 unit 0 family iso
suser@R5#et interfaces ge-0/0/2 unit 0 family inet6 address 2001:db8:3500:10::5/64
user@R5#set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
user@R5#set interfaces ge-0/0/2 unit 1 vlan-id 2
user@R5#set interfaces ge-0/0/2 unit 1 family inet address 192.168.35.2/24
user@R5#set interfaces ge-0/0/2 unit 1 family iso
user@R5#set interfaces ge-0/0/2 unit 1 family inet6 address 2001:db8:3500:20::5/64
user@R5#set interfaces ge-0/0/2 unit 1 family mpls maximum-labels 8
user@R5#set interfaces ge-0/0/3 description To_R4
user@R5#set interfaces ge-0/0/3 vlan-tagging
user@R5#set interfaces ge-0/0/3 unit 0 vlan-id 1
user@R5#set interfaces ge-0/0/3 unit 0 family inet address 192.168.13.2/24
```

```
user@R5#set interfaces ge-0/0/3 unit 0 family iso
user@R5#set interfaces ge-0/0/3 unit 0 family inet6 address 2001:db8:4500:10::5/64
user@R5#set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
user@R5#set interfaces ge-0/0/3 unit 1 vlan-id 2
user@R5#set interfaces ge-0/0/3 unit 1 family inet address 192.168.45.2/24
user@R5#set interfaces ge-0/0/3 unit 1 family iso
user@R5#set interfaces ge-0/0/3 unit 1 family inet6 address 2001:db8:4500:20::5/64
user@R5#set interfaces ge-0/0/3 unit 1 family mpls maximum-labels 8
user@R5#set interfaces ge-0/0/4 description To_R6
user@R5#set interfaces ge-0/0/4 vlan-tagging
user@R5#set interfaces ge-0/0/4 unit 0 vlan-id 1
user@R5#set interfaces ge-0/0/4 unit 0 family inet address 192.168.14.1/24
user@R5#set interfaces ge-0/0/4 unit 0 family iso
user@R5#set interfaces ge-0/0/4 unit 0 family inet6 address 2001:db8:5600:10::5/64
user@R5#set interfaces ge-0/0/4 unit 0 family mpls maximum-labels 8
user@R5#set interfaces ge-0/0/4 unit 1 vlan-id 2
user@R5#set interfaces ge-0/0/4 unit 1 family inet address 192.168.56.1/24
user@R5#set interfaces ge-0/0/4 unit 1 family iso
user@R5#set interfaces ge-0/0/4 unit 1 family inet6 address 2001:db8:5600:20::5/64
user@R5#set interfaces ge-0/0/4 unit 1 family mpls maximum-labels 8
```

3. Configure the loopback interface to enable tunnel endpoints and service endpoints.

```
[edit]
user@R5#set interfaces lo0 unit 0 family inet address 10.5.5.5/32
user@R5#set interfaces lo0 unit 0 family iso address 49.0001.0005.0505.0500
user@R5#set interfaces lo0 unit 0 family inet6 address 2001:db8:abcd::05:05:05:05/128
```

4. Define import and export policies. For example, configure policies that export EPE TE links from the local TE database to **lsdist.0** and policies to import from **lsdist.0** into the local TE database. You can configure policies to advertise the BGP routes to a peer.

```
[edit]
user@R5#set policy-options policy-statement nlri2bgp term 1 from family traffic-engineering
user@R5#set policy-options policy-statement nlri2bgp term 1 then next-hop self
user@R5#set policy-options policy-statement nlri2bgp term 1 then accept
user@R5#set policy-options policy-statement nlri2ted_igp term 1 from traffic-engineering
protocol isis-level-2
user@R5#set policy-options policy-statement nlri2ted_igp term 1 then accept
user@R5#set policy-options policy-statement pplb then load-balance per-packet
user@R5#set policy-options policy-statement prefix-sid term 1 from route-filter 10.5.5.5/32
```

```
exact
user@R5#set policy-options policy-statement prefix-sid term 1 then prefix-segment index 1005
user@R5#set policy-options policy-statement prefix-sid term 1 then prefix-segment node-
segment
user@R5#set policy-options policy-statement ted2nlri term 1 from protocol bgp-ls-epe
user@R5#set policy-options policy-statement ted2nlri term 1 then accept
user@R5#set policy-options policy-statement ted2nlri_igp term 1 from family traffic-
engineering
user@R5#set policy-options policy-statement ted2nlri_igp term 1 from protocol isis
user@R5#set policy-options policy-statement ted2nlri_igp term 1 then accept
```

5. Configure routing options to identify the router in the domain.

```
[edit]
user@R5#set routing-options router-id 10.5.5.5
user@R5#set routing-options autonomous-system 65200
```

6. Define export policies for forwarding table.

```
[edit]
user@R5#set routing-options forwarding-table export pplb
```

7. Configure BGP to enable BGP-LS route advertisement to the connected peers.

```
[edit]
user@R5#set protocols bgp group RR2 type internal
user@R5#set protocols bgp group RR2 family inet unicast
user@R5#set protocols bgp group RR2 family traffic-engineering unicast
user@R5#set protocols bgp group RR2 neighbor 10.1.1.1
user@R5#set protocols bgp group RR2 neighbor 10.3.3.3
user@R5#set protocols bgp group RR2 neighbor 10.6.6.6
user@R5#set protocols bgp group RR2 neighbor 10.4.4.4
user@R5#set protocols bgp cluster 10.5.5.5
```

8. Configure IS-IS protocol on the interfaces.

```
[edit]
user@R5#set protocols isis interface ge-0/0/0.0 level 2 lan-neighbor 0100.0101.0101 ipv4-
adjacency-segment protected label 5111
user@R5#set protocols isis interface ge-0/0/0.0 level 2 lan-neighbor 0100.0101.0101 ipv4-
adjacency-segment unprotected label 5101
user@R5#set protocols isis interface ge-0/0/0.0 level 2 post-convergence-lfa node-protection
user@R5#set protocols isis interface ge-0/0/1.0 level 2 lan-neighbor 0100.0202.0202 ipv4-
adjacency-segment protected label 5211
user@R5#set protocols isis interface ge-0/0/1.0 level 2 lan-neighbor 0100.0202.0202 ipv4-
adjacency-segment unprotected label 5201
user@R5#set protocols isis interface ge-0/0/1.0 level 2 post-convergence-lfa node-protection
user@R5#set protocols isis interface ge-0/0/2.0 level 2 lan-neighbor 0100.0303.0303 ipv4-
adjacency-segment protected label 5311
user@R5#set protocols isis interface ge-0/0/2.0 level 2 lan-neighbor 0100.0303.0303 ipv4-
adjacency-segment unprotected label 5301
user@R5#set protocols isis interface ge-0/0/2.0 level 2 post-convergence-lfa node-protection
user@R5#set protocols isis interface ge-0/0/3.0 level 2 lan-neighbor 0100.0404.0404 ipv4-
adjacency-segment protected label 5411
user@R5#set protocols isis interface ge-0/0/3.0 level 2 lan-neighbor 0100.0404.0404 ipv4-
adjacency-segment unprotected label 5401
user@R5#set protocols isis interface ge-0/0/3.0 level 2 post-convergence-lfa node-protection
user@R5#set protocols isis interface ge-0/0/4.0 level 2 lan-neighbor 0100.0606.0606 ipv4-
adjacency-segment protected label 5611
user@R5#set protocols isis interface ge-0/0/4.0 level 2 lan-neighbor 0100.0606.0606 ipv4-
adjacency-segment unprotected label 5601
user@R5#set protocols isis interface ge-0/0/4.0 level 2 post-convergence-lfa node-protection
user@R5#set protocols isis interface fxp0.0 disable
user@R5#set protocols isis interface lo0.0 passive
user@R5#set protocols isis source-packet-routing srgb start-label 800000 index-range 50000
user@R5#set protocols isis level 1 disable
user@R5#set protocols isis backup-spf-options use-post-convergence-lfa
user@R5#set protocols isis backup-spf-options use-source-packet-routing
user@R5#set protocols isis traffic-engineering l3-unicast-topology
user@R5#set protocols isis traffic-engineering advertisement always
user@R5#set protocols isis export prefix-sid
```

9. Configure MPLS administrative group policies for LSP path computation.

```
[edit]
user@R5#set protocols mpls admin-groups red 0
user@R5#set protocols mpls admin-groups blue 1
user@R5#set protocols mpls admin-groups green 2
user@R5#set protocols mpls admin-groups yellow 3
user@R5#set protocols mpls admin-groups orange 4
user@R5#set protocols mpls admin-groups brown 5
user@R5#set protocols mpls admin-groups black 6
user@R5#set protocols mpls admin-groups pink 7
```

10. Configure the MPLS label range to assign static labels for the EPE links.

```
[edit]
user@R5#set protocols mpls label-range static-label-range 1000 70000
```

11. Configure MPLS with interface and include administrative groups

```
[edit]
user@R5#set protocols mpls interface ge-0/0/0.0 admin-group blue
user@R5#set protocols mpls interface ge-0/0/1.0 admin-group red
user@R5#set protocols mpls interface ge-0/0/2.0 admin-group green
user@R5#set protocols mpls interface ge-0/0/3.0 admin-group brown
```

Results

From configuration mode, confirm your configuration by entering the `show chassis`, `show interfaces`, `show policy-options`, `show routing-options`, and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
chassis {
    network-services enhanced-ip;
}
interfaces {
    ge-0/0/0 {
        description To_R1;
        vlan-tagging;
```

```
        unit 0 {
            vlan-id 1;
            family inet {
                address 192.168.5.2/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:1500:10::5/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
        unit 1 {
            vlan-id 2;
            family inet {
                address 192.168.22.2/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:1500:20::5/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/1 {
        description To_R2;
        vlan-tagging;
        unit 0 {
            vlan-id 1;
            family inet {
                address 192.168.8.2/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:2500:10::5/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
```

```
    }
    ge-0/0/2 {
        description To_R3;
        vlan-tagging;
        unit 0 {
            vlan-id 1;
            family inet {
                address 192.168.10.2/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:3500:10::5/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
        unit 1 {
            vlan-id 2;
            family inet {
                address 192.168.35.2/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:3500:20::5/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/3 {
        description To_R4;
        vlan-tagging;
        unit 0 {
            vlan-id 1;
            family inet {
                address 192.168.13.2/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:4500:10::5/64;
            }
```

```
            family mpls {
                maximum-labels 8;
            }
        }
        unit 1 {
            vlan-id 2;
            family inet {
                address 192.168.45.2/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:4500:20::5/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/4 {
        description To_R6;
        vlan-tagging;
        unit 0 {
            vlan-id 1;
            family inet {
                address 192.168.14.1/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:5600:10::5/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
        unit 1 {
            vlan-id 2;
            family inet {
                address 192.168.56.1/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:5600:20::5/64;
            }
```

```
            family mpls {
                maximum-labels 8;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.5.5.5/32;
            }
            family iso {
                address 49.0001.0005.0505.0500;
            }
            family inet6 {
                address 2001:db8:abcd::05:05:05:05/128;
            }
        }
    }
}
policy-options {
    policy-statement nlri2bgp {
        term 1 {
            from family traffic-engineering;
            then {
                next-hop self;
                accept;
            }
        }
    }
    policy-statement nlri2ted_igp {
        term 1 {
            from {
                traffic-engineering {
                    protocol isis-level-2;
                }
            }
            then accept;
        }
    }
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
```

```
    }
    policy-statement prefix-sid {
        term 1 {
            from {
                route-filter 10.5.5.5/32 exact;
            }
            then {
                prefix-segment {
                    index 1005;
                    node-segment;
                }
            }
        }
    }
    policy-statement ted2nlri {
        term 1 {
            from protocol bgp-ls-epe;
            then accept;
        }
    }
    policy-statement ted2nlri_igp {
        term 1 {
            from {
                family traffic-engineering;
                protocol isis;
            }
            then accept;
        }
    }
}
routing-options {
    router-id 10.5.5.5;
    autonomous-system 65200;
    forwarding-table {
        export pplb;
    }
}
protocols {
    bgp {
        group RR2 {
            type internal;
            family inet {
                unicast;
```

```
            }
            family traffic-engineering {
                unicast;
            }
            neighbor 10.1.1.1;
            neighbor 10.3.3.3;
            neighbor 10.6.6.6;
            neighbor 10.4.4.4;
        }
        cluster 10.5.5.5;
    }
    isis {
        interface ge-0/0/0.0 {
            level 2 {
                lan-neighbor 0100.0101.0101 {
                    ipv4-adjacency-segment {
                        protected label 5111;
                        unprotected label 5101;
                    }
                }
                post-convergence-lfa {
                    node-protection;
                }
            }
        }
        interface ge-0/0/1.0 {
            level 2 {
                lan-neighbor 0100.0202.0202 {
                    ipv4-adjacency-segment {
                        protected label 5211;
                        unprotected label 5201;
                    }
                }
                post-convergence-lfa {
                    node-protection;
                }
            }
        }
        interface ge-0/0/2.0 {
            level 2 {
                lan-neighbor 0100.0303.0303 {
                    ipv4-adjacency-segment {
                        protected label 5311;
```

```
                unprotected label 5301;
            }
        }
        post-convergence-lfa {
            node-protection;
        }
    }
}
interface ge-0/0/3.0 {
    level 2 {
        lan-neighbor 0100.0404.0404 {
            ipv4-adjacency-segment {
                protected label 5411;
                unprotected label 5401;
            }
        }
        post-convergence-lfa {
            node-protection;
        }
    }
}
interface ge-0/0/4.0 {
    level 2 {
        lan-neighbor 0100.0606.0606 {
            ipv4-adjacency-segment {
                protected label 5611;
                unprotected label 5601;
            }
        }
        post-convergence-lfa {
            node-protection;
        }
    }
}
interface fxp0.0 {
    disable;
}
interface lo0.0 {
    passive;
}
source-packet-routing {
    srgb start-label 800000 index-range 50000;
}
```

```
        level 1 disable;
        backup-spf-options {
            use-post-convergence-lfa;
            use-source-packet-routing;
        }
        traffic-engineering {
            l3-unicast-topology;
            advertisement always;
        }
        export prefix-sid;
    }
    mpls {
        admin-groups {
            red 0;
            blue 1;
            green 2;
            yellow 3;
            orange 4;
            brown 5;
            black 6;
            pink 7;
        }
        label-range {
            static-label-range 1000 70000;
        }
        interface ge-0/0/0.0 {
            admin-group blue;
        }
        interface ge-0/0/1.0 {
            admin-group red;
        }
        interface ge-0/0/2.0 {
            admin-group green;
        }
        interface ge-0/0/3.0 {
            admin-group brown;
        }
        interface ge-0/0/4.0 {
            admin-group brown;
        }
        interface all;
        interface fxp0.0 {
            disable;
```

```
        }
    }
}
```

Configure R3 (BN3 router)

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the CLI User Guide.

To configure device R3:

1. Configure the network services mode as Enhanced IP. Enhanced IP sets the router's network services to enhanced Internet Protocol and uses enhanced mode capabilities.

```
[edit]
user@R3#set chassis network-services enhanced-ip
```

After you configure the `enhanced-ip` statement and commit the configuration, the following warning message appears prompting you to reboot the router:

```
'chassis'
WARNING: Chassis configuration for network services has been changed. A system reboot is
mandatory. Please reboot the system NOW. Continuing without a reboot might result in
unexpected system behavior.
commit complete
```

The reboot brings up the FPCs on the router.

2. Configure the interfaces to enable IP, MPLS, and ISO transport.

```
user@R3#set interfaces ge-0/0/0 description To_R2
user@R3#set interfaces ge-0/0/0 vlan-tagging
user@R3#set interfaces ge-0/0/0 unit 0 vlan-id 1
user@R3#set interfaces ge-0/0/0 unit 0 family inet address 192.168.6.2/24
user@R3#set interfaces ge-0/0/0 unit 0 family iso
user@R3#set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:2300:10::3/64
user@R3#set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 8
user@R3#set interfaces ge-0/0/0 unit 1 vlan-id 2
```

```
user@R3#set interfaces ge-0/0/0 unit 1 family inet address 192.168.23.2/24
user@R3#set interfaces ge-0/0/0 unit 1 family iso
user@R3#set interfaces ge-0/0/0 unit 1 family inet6 address 2001:db8:2300:20::3/64
user@R3#set interfaces ge-0/0/0 unit 1 family mpls maximum-labels 8
user@R3#set interfaces ge-0/0/0 unit 2 vlan-id 3
user@R3#set interfaces ge-0/0/0 unit 2 family inet address 192.168.30.2/24
user@R3#set interfaces ge-0/0/0 unit 2 family iso
user@R3#set interfaces ge-0/0/0 unit 2 family inet6 address 2001:db8:2300:30::3/64
user@R3#set interfaces ge-0/0/0 unit 2 family mpls maximum-labels 8
user@R3#set interfaces ge-0/0/1 description To_R5
user@R3#set interfaces ge-0/0/1 vlan-tagging
user@R3#set interfaces ge-0/0/1 unit 0 vlan-id 1
user@R3#set interfaces ge-0/0/1 unit 0 family inet address 192.168.10.1/24
user@R3#set interfaces ge-0/0/1 unit 0 family iso
user@R3#set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:3500:10::3/64
user@R3#set interfaces ge-0/0/1 unit 0 family mpls maximum-labels 8
user@R3#set interfaces ge-0/0/1 unit 1 vlan-id 2
user@R3#set interfaces ge-0/0/1 unit 1 family inet address 192.168.35.1/24
user@R3#set interfaces ge-0/0/1 unit 1 family iso
user@R3#set interfaces ge-0/0/1 unit 1 family inet6 address 2001:db8:3500:20::3/64
user@R3#set interfaces ge-0/0/1 unit 1 family mpls maximum-labels 8
user@R3#set interfaces ge-0/0/2 description To_R6
user@R3#set interfaces ge-0/0/2 vlan-tagging
user@R3#set interfaces ge-0/0/2 unit 0 vlan-id 1
user@R3#set interfaces ge-0/0/2 unit 0 family inet address 192.168.11.1/24
user@R3#set interfaces ge-0/0/2 unit 0 family iso
user@R3#set interfaces ge-0/0/2 unit 0 family inet6 address 2001:db8:3600:10::3/64
user@R3#set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
user@R3#set interfaces ge-0/0/3 description To_R7
user@R3#set interfaces ge-0/0/3 vlan-tagging
user@R3#set interfaces ge-0/0/3 unit 0 vlan-id 1
user@R3#set interfaces ge-0/0/3 unit 0 family inet address 192.168.12.1/24
user@R3#set interfaces ge-0/0/3 unit 0 family iso
user@R3#set interfaces ge-0/0/3 unit 0 family inet6 address 2001:db8:3700:10::3/6
user@R3#set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
user@R3#set interfaces ge-0/0/3 unit 1 vlan-id 2
user@R3#set interfaces ge-0/0/3 unit 1 family inet address 192.168.37.1/24
user@R3#set interfaces ge-0/0/3 unit 1 family iso
user@R3#set interfaces ge-0/0/3 unit 1 family inet6 address 2001:db8:3700:20::3/6
user@R3#set interfaces ge-0/0/3 unit 1 family mpls maximum-labels 8
```

3. Configure the loopback interface to enable tunnel endpoints and service endpoints.

```
[edit]
user@R3#set interfaces lo0 unit 0 family inet address 10.3.3.3/32
user@R3#set interfaces lo0 unit 0 family iso address 49.0001.0003.0303.0300
user@R3#set interfaces lo0 unit 0 family inet6 address 2001:db8:abcd::03:03:03:03/128
```

4. Define import and export policies. For example, configure policies that export EPE TE links from the local TE database to **lsdist.0** and policies to import from **lsdist.0** into the local TE database. You can configure policies to advertise the BGP routes to a peer.

```
[edit]
user@R3#set policy-options policy-statement bgplsepe_rt_2_ted term 1 from protocol bgp
user@R3#set policy-options policy-statement bgplsepe_rt_2_ted term 1 then accept
user@R3#set policy-options policy-statement expresspol1 from route-filter 10.1.1.1/32 exact
install-nexthop lsp lsp3to1_a
user@R3#set policy-options policy-statement expresspol1 then accept
user@R3#set policy-options policy-statement expresspol2 from route-filter 10.4.4.4/32 exact
install-nexthop lsp lsp3to4_a
user@R3#set policy-options policy-statement expresspol2 then accept
user@R3#set policy-options policy-statement expresspolsr1 from protocol spring-te
user@R3#set policy-options policy-statement expresspolsr1 from route-filter 10.1.1.1/32
exact
user@R3#set policy-options policy-statement expresspolsr1 then accept
user@R3#set policy-options policy-statement nlri2bgp term 1 from family traffic-engineering
user@R3#set policy-options policy-statement nlri2bgp term 1 then next-hop self
user@R3#set policy-options policy-statement nlri2bgp term 1 then accept
user@R3#set policy-options policy-statement nlri2bgp_epe term 1 from family traffic-
engineering
user@R3#set policy-options policy-statement nlri2bgp_epe term 1 from protocol bgp-ls-epe
user@R3#set policy-options policy-statement nlri2bgp_epe term 1 then next-hop self
user@R3#set policy-options policy-statement nlri2bgp_epe term 1 then accept
user@R3#set policy-options policy-statement nlri2bgp_igp term 1 from family traffic-
engineering
user@R3#set policy-options policy-statement nlri2bgp_igp term 1 from protocol isis
user@R3#set policy-options policy-statement nlri2bgp_igp term 1 then accept
user@R3#set policy-options policy-statement nlri2bgp_stat term 1 from family traffic-
engineering
user@R3#set policy-options policy-statement nlri2bgp_stat term 1 from protocol express-
segments
user@R3#set policy-options policy-statement nlri2bgp_stat term 1 then accept
```

```
user@R3#set policy-options policy-statement pplb then load-balance per-packet
user@R3#set policy-options policy-statement prefix-sid term 1 from route-filter 10.3.3.3/32
exact
user@R3#set policy-options policy-statement prefix-sid term 1 then prefix-segment index 1003
user@R3#set policy-options policy-statement prefix-sid term 1 then prefix-segment node-
segment
user@R3#set policy-options policy-statement ted2nlri term 1 from protocol bgp-ls-epe
user@R3#set policy-options policy-statement ted2nlri term 1 then accept
user@R3#set policy-options policy-statement ted2nlri_epe_stat term 1 from family traffic-
engineering
user@R3#set policy-options policy-statement ted2nlri_epe_stat term 1 from protocol static
user@R3#set policy-options policy-statement ted2nlri_epe_stat term 1 then accept
user@R3#set policy-options policy-statement ted2nlri_epe_stat term 2 from family traffic-
engineering
user@R3#set policy-options policy-statement ted2nlri_epe_stat term 2 from protocol bgp-ls-
epe
user@R3#set policy-options policy-statement ted2nlri_epe_stat term 2 then accept
user@R3#set policy-options policy-statement ted2nlri_epe_stat term 3 from protocol isis
user@R3#set policy-options policy-statement ted2nlri_epe_stat term 3 then reject
user@R3#set policy-options policy-statement ted2nlri_igp from family traffic-engineering
user@R3#set policy-options policy-statement ted2nlri_igp from protocol isis
user@R3#set policy-options policy-statement ted2nlri_igp then accept
```

5. Configure routing options to identify the router in the domain.

```
[edit]
user@R3#set routing-options router-id 10.3.3.3
user@R3#set routing-options autonomous-system 65200
```

6. Define the RIB group to copy junos-rti-tc-<color>.inet.0 to inet.3 routing table.

```
[edit]
user@R3#set routing-options rib-groups color-to-inet3 import-rib [ junos-rti-tc-
<color>.inet.0 inet.3 ]
```

7. Configure BGP to enable BGP-LS route advertisement for peer and define the EPE links. Since express segment is an internal TE link, this configuration creates an external TE link.

```
[edit]
user@R3#set protocols bgp group ibgp1 type internal
```

```
user@R3#set protocols bgp group ibgp1 local-address 10.3.3.3
user@R3#set protocols bgp group ibgp1 family traffic-engineering unicast
user@R3#set protocols bgp group ibgp1 export nlri2bgp_epe
user@R3#set protocols bgp group ibgp1 neighbor 10.2.2.2
user@R3#set protocols bgp group ibgp1 neighbor 10.5.5.5
user@R3#set protocols bgp group ebgp1 type external
user@R3#set protocols bgp group ebgp1 family traffic-engineering unicast
user@R3#set protocols bgp group ebgp1 export nlri2bgp_stat
user@R3#set protocols bgp group ebgp1 neighbor 192.168.12.2 peer-as 65300
user@R3#set protocols bgp group ebgp1 neighbor 192.168.12.2 egress-te-adj-segment
epe_adj1_toR7 label 7137
user@R3#set protocols bgp group ebgp1 neighbor 192.168.12.2 egress-te-adj-segment
epe_adj1_toR7 next-hop 192.168.12.2
user@R3#set protocols bgp group ebgp1 neighbor 192.168.12.2 egress-te-adj-segment
epe_adj1_toR7 te-link-attribute te-metric 20
user@R3#set protocols bgp group ebgp1 neighbor 192.168.12.2 egress-te-adj-segment
epe_adj1_toR7 te-link-attribute igp-metric 10
user@R3#set protocols bgp group ebgp1 neighbor 192.168.12.2 egress-te-adj-segment
epe_adj1_toR7 te-link-attribute admin-group red
user@R3#set protocols bgp group ebgp1 neighbor 192.168.12.2 egress-te-adj-segment
epe_adj1_toR7 te-link-attribute admin-group brown
user@R3#set protocols bgp group ebgp1 vpn-apply-export
```

8. Define a mechanism to automatically (dynamic) create express segments and insert them in to the TE database so that they can be advertised through BGP-LS. In this example, express segments are created for all the underlay SR tunnels automatically. This is done by configuring a template with a policy and then express segments are automatically created based on the policies.

```
[edit]
user@R3#set protocols express-segments segment-set set3sr membership-policy expresspolsr1
user@R3#set protocols express-segments traffic-engineering
```

9. Configure IS-IS protocol on the interfaces.

```
[edit]
user@R3#set protocols isis interface ge-0/0/0.0 level 2 lan-neighbor 0100.0202.0202 ipv4-
adjacency-segment protected label 3211
user@R3#set protocols isis interface ge-0/0/0.0 level 2 lan-neighbor 0100.0202.0202 ipv4-
adjacency-segment unprotected label 3201
user@R3#set protocols isis interface ge-0/0/0.0 level 2 post-convergence-lfa node-protection
user@R3#set protocols isis interface ge-0/0/1.0 level 2 lan-neighbor 0100.0505.0505 ipv4-
```

```
adjacency-segment protected label 3511
user@R3#set protocols isis interface ge-0/0/1.0 level 2 lan-neighbor 0100.0505.0505 ipv4-
adjacency-segment unprotected label 3501
user@R3#set protocols isis interface ge-0/0/1.0 level 2 post-convergence-lfa node-protection
user@R3#set protocols isis interface ge-0/0/2.0 level 2 lan-neighbor 0100.0606.0606 ipv4-
adjacency-segment protected label 3611
user@R3#set protocols isis interface ge-0/0/2.0 level 2 lan-neighbor 0100.0606.0606 ipv4-
adjacency-segment unprotected label 3601
user@R3#set protocols isis interface ge-0/0/2.0 level 2 post-convergence-lfa node-protection
user@R3#set protocols isis interface ge-0/0/3.0 passive
user@R3#set protocols isis interface fxp0.0 disable
user@R3#set protocols isis interface lo0.0 passive
user@R3#set protocols isis source-packet-routing srgb start-label 800000 index-range 50000
user@R3#set protocols isis level 1 disable
user@R3#set protocols isis level 2 wide-metrics-only
user@R3#set protocols isis backup-spf-options use-post-convergence-lfa
user@R3#set protocols isis backup-spf-options use-source-packet-routing
user@R3#set protocols isis traffic-engineering l3-unicast-topology
user@R3#set protocols isis traffic-engineering advertisement always
user@R3#set protocols isis export prefix-sid
```

10. Enable import and export of traffic engineering database parameters using policies.

```
[edit]
user@R3#set protocols mpls traffic-engineering database import l3-unicast-topology bgp-link-
state
user@R3#set protocols mpls traffic-engineering database import policy ted2nlri_epe_stat
user@R3#set protocols mpls traffic-engineering database export l3-unicast-topology
```

11. Configure MPLS administrative group policies for LSP path computation.

```
[edit]
user@R3#set protocols mpls admin-groups red 0
user@R3#set protocols mpls admin-groups blue 1
user@R3#set protocols mpls admin-groups green 2
user@R3#set protocols mpls admin-groups yellow 3
user@R3#set protocols mpls admin-groups orange 4
user@R3#set protocols mpls admin-groups brown 5
user@R3#set protocols mpls admin-groups black 6
user@R3#set protocols mpls admin-groups pink 7
```

12. Configure the MPLS label range to assign static labels for the EPE links.

```
[edit]
user@R3#set protocols mpls label-range static-label-range 1000 70000
```

13. Configure MPLS with interface and include administrative groups.

```
[edit]
user@R3#set protocols mpls interface ge-0/0/0.0 admin-group brown
user@R3#set protocols mpls interface ge-0/0/1.0 admin-group green
user@R3#set protocols mpls interface ge-0/0/2.0 admin-group red
user@R3#set protocols mpls interface ge-0/0/3.0 admin-group [ red brown ]
user@R3#set protocols mpls interface all
user@R3#set protocols mpls interface fxp0.0 disable
```

14. Configure ST-TE LSP from R3 device to R1 device.

```
[edit]
user@R3#set protocols source-packet-routing no-chained-composite-next-hop
user@R3#set protocols source-packet-routing segment-list R3-R2-R1 inherit-label-nexthops
user@R3#set protocols source-packet-routing segment-list R3-R2-R1 auto-translate
user@R3#set protocols source-packet-routing segment-list R3-R2-R1 hop1 ip-address
192.168.6.1
user@R3#set protocols source-packet-routing segment-list R3-R2-R1 hop2 ip-address
192.168.3.1
user@R3#set protocols source-packet-routing source-routing-path lsp3to1_sr to 10.1.1.1
user@R3#set protocols source-packet-routing source-routing-path lsp3to1_sr color 1000
user@R3#set protocols source-packet-routing source-routing-path lsp3to1_sr primary R3-R2-R1
user@R3#set protocols source-packet-routing rib-group ipv4-color color-to-inet3
```

Results

From configuration mode, confirm your configuration by entering the `show chassis`, `show interfaces`, `show policy-options`, `show routing-options`, and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

The following result includes colored SR-TE underlay path configuration also.

```
chassis {
    network-services enhanced-ip;
}
interfaces {
    ge-0/0/0 {
        description To_R2;
        vlan-tagging;
        unit 0 {
            vlan-id 1;
            family inet {
                address 192.168.6.2/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:2300:10::3/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
        unit 1 {
            vlan-id 2;
            family inet {
                address 192.168.23.2/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:2300:20::3/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
        unit 2 {
            vlan-id 3;
            family inet {
                address 192.168.30.2/24;
            }
            family iso;
            family inet6 {
```

```
                address 2001:db8:2300:30::3/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/1 {
        description To_R5;
        vlan-tagging;
        unit 0 {
            vlan-id 1;
            family inet {
                address 192.168.10.1/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:3500:10::3/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
        unit 1 {
            vlan-id 2;
            family inet {
                address 192.168.35.1/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:3500:20::3/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/2 {
        description To_R6;
        vlan-tagging;
        unit 0 {
            vlan-id 1;
            family inet {
```

```
                address 192.168.11.1/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:3600:10::3/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/3 {
        description To_R7;
        vlan-tagging;
        unit 0 {
            vlan-id 1;
            family inet {
                address 192.168.12.1/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:3700:10::3/6;
            }
            family mpls {
                maximum-labels 8;
            }
        }
        unit 1 {
            vlan-id 2;
            family inet {
                address 192.168.37.1/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:3700:20::3/6;
            }
            family mpls {
                maximum-labels 8;
            }
        }
    }
    lo0 {
        unit 0 {
```

```
            family inet {
                address 10.3.3.3/32;
            }
            family iso {
                address 49.0001.0003.0303.0300;
            }
            family inet6 {
                address 2001:db8:abcd::03:03:03:03/128;
            }
        }
    }
}
policy-options {
    policy-statement bgplsepe_rt_2_ted {
        term 1 {
            from protocol bgp;
            then accept;
        }
    }
    policy-statement expresspol1 {
        from {
            route-filter 10.1.1.1/32 exact {
                install-nexthop lsp lsp3to1_a;
            }
        }
        then accept;
    }
    policy-statement expresspol2 {
        from {
            route-filter 10.4.4.4/32 exact {
                install-nexthop lsp lsp3to4_a;
            }
        }
        then accept;
    }
    policy-statement expresspolsr1 {
        from {
            protocol spring-te;
            route-filter 10.1.1.1/32 exact;
        }
        then accept;
    }
    policy-statement nlri2bgp {
```

```
        term 1 {
            from family traffic-engineering;
            then {
                next-hop self;
                accept;
            }
        }
    }
    policy-statement nlri2bgp_epe {
        term 1 {
            from {
                family traffic-engineering;
                protocol bgp-ls-epe;
            }
            then {
                next-hop self;
                accept;
            }
        }
    }
    policy-statement nlri2bgp_igp {
        term 1 {
            from {
                family traffic-engineering;
                protocol isis;
            }
            then accept;
        }
    }
    policy-statement nlri2bgp_stat {
        term 1 {
            from {
                family traffic-engineering;
                protocol express-segments;
            }
            then accept;
        }
    }
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
    }
```

```
    policy-statement prefix-sid {
        term 1 {
            from {
                route-filter 10.3.3.3/32 exact;
            }
            then {
                prefix-segment {
                    index 1003;
                    node-segment;
                }
            }
        }
    }
    policy-statement ted2nlri {
        term 1 {
            from protocol bgp-ls-epe;
            then accept;
        }
    }
    policy-statement ted2nlri_epe_stat {
        term 1 {
            from {
                family traffic-engineering;
                protocol static;
            }
            then accept;
        }
        term 2 {
            from {
                family traffic-engineering;
                protocol bgp-ls-epe;
            }
            then accept;
        }
        term 3 {
            from protocol isis;
            then reject;
        }
    }
    policy-statement ted2nlri_igp {
        from {
            family traffic-engineering;
            protocol isis;
```

```
        }
        then accept;
    }
}
routing-options {
    router-id 10.3.3.3;
    autonomous-system 65200;
    rib-groups {
        color-to-inet3 {
            import-rib [ junos-rti-tc-<color>.inet.0 inet.3 ];
        }
    }
    forwarding-table {
        export pplb;
    }
}
protocols {
    bgp {
        group ibgp1 {
            type internal;
            local-address 10.3.3.3;
            family traffic-engineering {
                unicast;
            }
            export nlri2bgp_epe;
            neighbor 10.2.2.2;
            neighbor 10.5.5.5;
        }
        group ebgp1 {
            type external;
            family traffic-engineering {
                unicast;
            }
            export nlri2bgp_stat;
            neighbor 192.168.12.2 {
                peer-as 65300;
                egress-te-adj-segment epe_adj1_toR7 {
                    label 7137;
                    next-hop 192.168.12.2;
                    te-link-attribute {
                        te-metric 20;
                        igp-metric 10;
                        admin-group [ red brown ];
```

```
                }
            }
        }
        vpn-apply-export;
    }
}
express-segments {
    segment-set set3sr {
        membership-policy expresspolsr1;
    }
    traffic-engineering;
}
isis {
    interface ge-0/0/0.0 {
        level 2 {
            lan-neighbor 0100.0202.0202 {
                ipv4-adjacency-segment {
                    protected label 3211;
                    unprotected label 3201;
                }
            }
            post-convergence-lfa {
                node-protection;
            }
        }
    }
    interface ge-0/0/1.0 {
        level 2 {
            lan-neighbor 0100.0505.0505 {
                ipv4-adjacency-segment {
                    protected label 3511;
                    unprotected label 3501;
                }
            }
            post-convergence-lfa {
                node-protection;
            }
        }
    }
    interface ge-0/0/2.0 {
        level 2 {
            lan-neighbor 0100.0606.0606 {
                ipv4-adjacency-segment {
```

```
                    protected label 3611;
                    unprotected label 3601;
                }
            }
            post-convergence-lfa {
                node-protection;
            }
        }
    }
    interface ge-0/0/3.0 {
        passive;
    }
    interface fxp0.0 {
        disable;
    }
    interface lo0.0 {
        passive;
    }
    source-packet-routing {
        srgb start-label 800000 index-range 50000;
    }
    level 1 disable;
    level 2 wide-metrics-only;
    backup-spf-options {
        use-post-convergence-lfa;
        use-source-packet-routing;
    }
    traffic-engineering {
        l3-unicast-topology;
        advertisement always;
    }
    export prefix-sid;
}
mpls {
    traffic-engineering {
        database {
            import {
                l3-unicast-topology {
                    bgp-link-state;
                }
                policy ted2nlri_epe_stat;
            }
            export {
```

```
                    l3-unicast-topology;
                }
            }
        }
        admin-groups {
            red 0;
            blue 1;
            green 2;
            yellow 3;
            orange 4;
            brown 5;
            black 6;
            pink 7;
        }
        label-range {
            static-label-range 1000 70000;
        }
        interface ge-0/0/0.0 {
            admin-group brown;
        }
        interface ge-0/0/1.0 {
            admin-group green;
        }
        interface ge-0/0/2.0 {
            admin-group red;
        }
        interface ge-0/0/3.0 {
            admin-group [ red brown ];
        }
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
    source-packet-routing {
        no-chained-composite-next-hop;
        segment-list R3-R2-R1 {
            inherit-label-nexthops;
            auto-translate;
            hop1 ip-address 192.168.6.1;
            hop2 ip-address 192.168.3.1;
        }
        source-routing-path lsp3to1_sr {
```

```
            to 10.1.1.1;
            color 1000;
            primary {
                R3-R2-R1;
            }
        }
        rib-group {
            ipv4-color {
                color-to-inet3;
            }
        }
    }
}
```

Configure R6 (BN4 router)

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the CLI User Guide.

To configure device R6:

1. Configure the network services mode as Enhanced IP. Enhanced IP sets the router's network services to enhanced Internet Protocol and uses enhanced mode capabilities.

   ```
   [edit]
   user@R6#set chassis network-services enhanced-ip
   ```

   After you configure the `enhanced-ip` statement and commit the configuration, the following warning message appears prompting you to reboot the router:

   ```
   'chassis'
   WARNING: Chassis configuration for network services has been changed. A system reboot is
   mandatory. Please reboot the system NOW. Continuing without a reboot might result in
   unexpected system behavior.
   commit complete
   ```

   The reboot brings up the FPCs on the router.

2. Configure the interfaces to enable IP, MPLS, and ISO transport.

```
user@R6#set interfaces ge-0/0/0 description To_R2
user@R6#set interfaces ge-0/0/0 vlan-tagging
user@R6#set interfaces ge-0/0/0 unit 0 vlan-id 1
user@R6#set interfaces ge-0/0/0 unit 0 family inet address 192.168.9.2/24
user@R6#set interfaces ge-0/0/0 unit 0 family iso
user@R6#set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:2600:10::6/64
user@R6#set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 8
user@R6#set interfaces ge-0/0/0 unit 1 vlan-id 2
user@R6#set interfaces ge-0/0/0 unit 1 family inet address 26.26.20.6/24
user@R6#set interfaces ge-0/0/0 unit 1 family iso
user@R6#set interfaces ge-0/0/0 unit 1 family inet6 address 2001:db8:2600:20::6/64
user@R6#set interfaces ge-0/0/0 unit 1 family mpls maximum-labels 8
user@R6#set interfaces ge-0/0/1 description To_R3
user@R6#set interfaces ge-0/0/1 vlan-tagging
user@R6#set interfaces ge-0/0/1 unit 0 vlan-id 1
user@R6#set interfaces ge-0/0/1 unit 0 family inet address 192.168.11.2/24
user@R6#set interfaces ge-0/0/1 unit 0 family iso
user@R6#set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:3600:10::6/64
user@R6#set interfaces ge-0/0/1 unit 0 family mpls maximum-labels 8
user@R6#set interfaces ge-0/0/2 description To_R5
user@R6#set interfaces ge-0/0/2 vlan-tagging
user@R6#set interfaces ge-0/0/2 unit 0 vlan-id 1
user@R6#set interfaces ge-0/0/2 unit 0 family inet address 192.168.14.2/24
user@R6#set interfaces ge-0/0/2 unit 0 family iso
user@R6#set interfaces ge-0/0/2 unit 0 family inet6 address 2001:db8:5600:10::6/64
user@R6#set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
user@R6#set interfaces ge-0/0/2 unit 1 vlan-id 2
user@R6#set interfaces ge-0/0/2 unit 1 family inet address 56.56.20.6/24
user@R6#set interfaces ge-0/0/2 unit 1 family iso
user@R6#set interfaces ge-0/0/2 unit 1 family inet6 address 2001:db8:5600:20::6/64
user@R6#set interfaces ge-0/0/2 unit 1 family mpls maximum-labels 8
user@R6#set interfaces ge-0/0/3 description To_R7
user@R6#set interfaces ge-0/0/3 vlan-tagging
user@R6#set interfaces ge-0/0/3 unit 0 vlan-id 1
user@R6#set interfaces ge-0/0/3 unit 0 family inet address 192.168.15.1/24
user@R6#set interfaces ge-0/0/3 unit 0 family iso
user@R6#set interfaces ge-0/0/3 unit 0 family inet6 address 2001:db8:6700:10::6/64
user@R6#set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
user@R6#set interfaces ge-0/0/3 unit 1 vlan-id 2
user@R6#set interfaces ge-0/0/3 unit 1 family inet address 67.67.20.6/24
```

```
user@R6#set interfaces ge-0/0/3 unit 1 family iso
user@R6#set interfaces ge-0/0/3 unit 1 family inet6 address 2001:db8:6700:20::6/64
user@R6#set interfaces ge-0/0/3 unit 1 family mpls maximum-labels 8
```

3. Configure the loopback interface to enable tunnel endpoints and service endpoints.

```
[edit]
user@R6#set interfaces lo0 unit 0 family inet address 10.6.6.6/32
user@R6#set interfaces lo0 unit 0 family iso address 49.0001.0006.0606.0600
user@R6#set interfaces lo0 unit 0 family inet6 address 2001:db8:abcd::06:06:06:06/128
```

4. Define import and export policies. For example, configure policies that export EPE TE links from the local TE database to **lsdist.0** and policies to import from **lsdist.0** into the local TE database. You can configure policies to advertise the BGP routes to a peer.

```
[edit]
user@R6#set policy-options policy-statement expresspolsr1 from protocol spring-te
user@R6#set policy-options policy-statement expresspolsr1 from route-filter 10.4.4.4/32
exact
user@R6#set policy-options policy-statement expresspolsr1 then accept
user@R6#set policy-options policy-statement nlri2bgp_epe term 1 from family traffic-
engineering
user@R6#set policy-options policy-statement nlri2bgp_epe term 1 from protocol bgp-ls-epe
user@R6#set policy-options policy-statement nlri2bgp_epe term 1 then next-hop self
user@R6#set policy-options policy-statement nlri2bgp_epe term 1 then accept
user@R6#set policy-options policy-statement nlri2bgp_stat term 1 from family traffic-
engineering
user@R6#set policy-options policy-statement nlri2bgp_stat term 1 from protocol express-
segments
user@R6#set policy-options policy-statement nlri2bgp_stat term 1 then accept
user@R6#set policy-options policy-statement pplb then load-balance per-packet
user@R6#set policy-options policy-statement prefix-sid term 1 from route-filter 10.6.6.6/32
exact
user@R6#set policy-options policy-statement prefix-sid term 1 then prefix-segment index 1006
user@R6#set policy-options policy-statement prefix-sid term 1 then prefix-segment node-
segment
user@R6#set policy-options policy-statement ted2nlri_epe_stat term 1 from family traffic-
engineering
user@R6#set policy-options policy-statement ted2nlri_epe_stat term 1 from protocol static
user@R6#set policy-options policy-statement ted2nlri_epe_stat term 1 then accept
user@R6#set policy-options policy-statement ted2nlri_epe_stat term 2 from family traffic-
```

```
engineering
user@R6#set policy-options policy-statement ted2nlri_epe_stat term 2 from protocol bgp-ls-
epe
user@R6#set policy-options policy-statement ted2nlri_epe_stat term 2 then accept
user@R6#set policy-options policy-statement ted2nlri_epe_stat term 3 from protocol isis
user@R6#set policy-options policy-statement ted2nlri_epe_stat term 3 then reject
```

5. Configure routing options to identify the router in the domain.

```
[edit]
user@R6#set routing-options router-id 10.6.6.6
user@R6#set routing-options autonomous-system 65200
```

6. Define the RIB group to copy junos-rti-tc-<color>.inet.0 to inet.3 routing table.

```
[edit]
user@R6#set routing-options rib-groups color-to-inet3 import-rib [ junos-rti-tc-
<color>.inet.0 inet.3 ]
```

7. Configure BGP to enable BGP-LS route advertisement for peer and define the EPE links. Since express segment is an internal TE link, this configuration creates an external TE link.

```
[edit]
user@R6#set protocols bgp group ibgp1 type internal
user@R6#set protocols bgp group ibgp1 local-address 10.6.6.6
user@R6#set protocols bgp group ibgp1 family traffic-engineering unicast
user@R6#set protocols bgp group ibgp1 export nlri2bgp_epe
user@R6#set protocols bgp group ibgp1 neighbor 10.2.2.2
user@R6#set protocols bgp group ibgp1 neighbor 10.5.5.5
user@R6#set protocols bgp group ebgp1 type external
user@R6#set protocols bgp group ebgp1 family traffic-engineering unicast
user@R6#set protocols bgp group ebgp1 export nlri2bgp_stat
user@R6#set protocols bgp group ebgp1 neighbor 192.168.15.2 peer-as 65300
user@R6#set protocols bgp group ebgp1 neighbor 192.168.15.2 egress-te-adj-segment
epe_adj1_toR7 label 7167
user@R6#set protocols bgp group ebgp1 neighbor 192.168.15.2 egress-te-adj-segment
epe_adj1_toR7 next-hop 192.168.15.2
user@R6#set protocols bgp group ebgp1 neighbor 192.168.15.2 egress-te-adj-segment
epe_adj1_toR7 te-link-attribute te-metric 20
user@R6#set protocols bgp group ebgp1 neighbor 192.168.15.2 egress-te-adj-segment
```

```
epe_adj1_toR7 te-link-attribute igp-metric 10
user@R6#set protocols bgp group ebgp1 neighbor 192.168.15.2 egress-te-adj-segment
epe_adj1_toR7 te-link-attribute admin-group red
user@R6#set protocols bgp group ebgp1 neighbor 192.168.15.2 egress-te-adj-segment
epe_adj1_toR7 te-link-attribute admin-group brown
```

8. Define a mechanism to automatically (dynamic) create express segments and insert them in to the TE database so that they can be advertised through BGP-LS. In this example, express segments are created for all the underlay SR tunnels automatically. This is done by configuring a template with a policy and then express segments are automatically created based on the policies.

```
[edit]
user@R6#set protocols express-segments segment-set set6sr membership-policy expresspolsr1
user@R6#set protocols express-segments traffic-engineering
```

9. Configure IS-IS protocol on the interfaces.

```
[edit]
user@R6#set protocols isis interface ge-0/0/0.0 level 2 lan-neighbor 0100.0202.0202 ipv4-
adjacency-segment protected label 6211
user@R6#set protocols isis interface ge-0/0/0.0 level 2 lan-neighbor 0100.0202.0202 ipv4-
adjacency-segment unprotected label 6201
user@R6#set protocols isis interface ge-0/0/0.0 level 2 post-convergence-lfa node-protection
user@R6#set protocols isis interface ge-0/0/1.0 level 2 lan-neighbor 0100.0303.0303 ipv4-
adjacency-segment protected label 6311
user@R6#set protocols isis interface ge-0/0/1.0 level 2 lan-neighbor 0100.0303.0303 ipv4-
adjacency-segment unprotected label 6301
user@R6#set protocols isis interface ge-0/0/1.0 level 2 post-convergence-lfa node-protection
user@R6#set protocols isis interface ge-0/0/2.0 level 2 lan-neighbor 0100.0505.0505 ipv4-
adjacency-segment protected label 6511
user@R6#set protocols isis interface ge-0/0/2.0 level 2 lan-neighbor 0100.0505.0505 ipv4-
adjacency-segment unprotected label 6501
user@R6#set protocols isis interface ge-0/0/2.0 level 2 post-convergence-lfa node-protection
user@R6#set protocols isis interface ge-0/0/3.0 level 2 post-convergence-lfa node-protection
user@R6#set protocols isis interface ge-0/0/3.0 passive
user@R6#set protocols isis interface fxp0.0 disable
user@R6#set protocols isis interface lo0.0 passive
user@R6#set protocols isis source-packet-routing srgb start-label 800000 index-range 50000
user@R6#set protocols isis level 1 disable
user@R6#set protocols isis level 2 wide-metrics-only
user@R6#set protocols isis backup-spf-options use-post-convergence-lfa
```

```
user@R6#set protocols isis backup-spf-options use-source-packet-routing
user@R6#set protocols isis traffic-engineering l3-unicast-topology
user@R6#set protocols isis traffic-engineering advertisement always
user@R6#set protocols isis export prefix-sid
```

10. Enable import and export of traffic engineering database parameters using policies.

```
[edit]
user@R6#set protocols mpls traffic-engineering database import l3-unicast-topology bgp-link-state
user@R6#set protocols mpls traffic-engineering database import policy ted2nlri_epe_stat
user@R6#set protocols mpls traffic-engineering database export l3-unicast-topology
```

11. Configure MPLS administrative group policies for LSP path computation.

```
[edit]
user@R6#set protocols mpls admin-groups red 0
user@R6#set protocols mpls admin-groups blue 1
user@R6#set protocols mpls admin-groups green 2
user@R6#set protocols mpls admin-groups yellow 3
user@R6#set protocols mpls admin-groups orange 4
user@R6#set protocols mpls admin-groups brown 5
user@R6#set protocols mpls admin-groups black 6
user@R6#set protocols mpls admin-groups pink 7
```

12. Configure the MPLS label range to assign static labels for the EPE links.

```
[edit]
user@R6#set protocols mpls label-range static-label-range 1000 70000
```

13. Configure MPLS with interface and include administrative groups.

```
[edit]
user@R6#set protocols mpls interface ge-0/0/0.0 admin-group blue
user@R6#set protocols mpls interface ge-0/0/1.0 admin-group red
user@R6#set protocols mpls interface ge-0/0/2.0 admin-group brown
user@R6#set protocols mpls interface ge-0/0/3.0 admin-group [ red brown ]
```

```
user@R6#set protocols mpls interface all
user@R6#set protocols mpls interface fxp0.0 disable
```

14. Configure ST-TE LSP from R6 device to R4 device.

```
[edit]
user@R6#set protocols source-packet-routing no-chained-composite-next-hop
user@R6#set protocols source-packet-routing segment-list R6-R5-R4 hop1 label 801005
user@R6#set protocols source-packet-routing segment-list R6-R5-R4 hop2 label 801004
user@R6#set protocols source-packet-routing source-routing-path lsp6to4_sr to 10.4.4.4
user@R6#set protocols source-packet-routing source-routing-path lsp6to4_sr color 1000
user@R6#set protocols source-packet-routing source-routing-path lsp6to4_sr primary R6-R5-R4
user@R6#set protocols source-packet-routing rib-group ipv4-color color-to-inet3
```

Results

From configuration mode, confirm your configuration by entering the `show chassis`, `show interfaces`, `show policy-options`, `show routing-options`, and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

The following result includes colored SR-TE underlay path configuration also.

```
chassis {
    network-services enhanced-ip;
}
interfaces {
    ge-0/0/0 {
        description To_R2;
        vlan-tagging;
        unit 0 {
            vlan-id 1;
            family inet {
                address 192.168.9.2/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:2600:10::6/64;
            }
            family mpls {
                maximum-labels 8;
            }
```

```
        }
        unit 1 {
            vlan-id 2;
            family inet {
                address 26.26.20.6/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:2600:20::6/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/1 {
        description To_R3;
        vlan-tagging;
        unit 0 {
            vlan-id 1;
            family inet {
                address 192.168.11.2/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:3600:10::6/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/2 {
        description To_R5;
        vlan-tagging;
        unit 0 {
            vlan-id 1;
            family inet {
                address 192.168.14.2/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:5600:10::6/64;
```

```
            }
            family mpls {
                maximum-labels 8;
            }
        }
        unit 1 {
            vlan-id 2;
            family inet {
                address 56.56.20.6/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:5600:20::6/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/3 {
        description To_R7;
        vlan-tagging;
        unit 0 {
            vlan-id 1;
            family inet {
                address 192.168.15.1/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:6700:10::6/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
        unit 1 {
            vlan-id 2;
            family inet {
                address 67.67.20.6/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:6700:20::6/64;
```

```
            }
            family mpls {
                maximum-labels 8;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 6.6.6.6/32;
            }
            family iso {
                address 49.0001.0006.0606.0600;
            }
            family inet6 {
                address 2001:db8:abcd::06:06:06:06/128;
            }
        }
    }
}
policy-options {
    policy-statement expresspolsr1 {
        from {
            protocol spring-te;
            route-filter 10.4.4.4/32 exact;
        }
        then accept;
    }
    policy-statement nlri2bgp_epe {
        term 1 {
            from {
                family traffic-engineering;
                protocol bgp-ls-epe;
            }
            then {
                next-hop self;
                accept;
            }
        }
    }
    policy-statement nlri2bgp_stat {
        term 1 {
            from {
```

```
            family traffic-engineering;
            protocol express-segments;
        }
        then accept;
    }
}
policy-statement pplb {
    then {
        load-balance per-packet;
    }
}
policy-statement prefix-sid {
    term 1 {
        from {
            route-filter 10.6.6.6/32 exact;
        }
        then {
            prefix-segment {
                index 1006;
                node-segment;
            }
        }
    }
}
policy-statement ted2nlri_epe_stat {
    term 1 {
        from {
            family traffic-engineering;
            protocol static;
        }
        then accept;
    }
    term 2 {
        from {
            family traffic-engineering;
            protocol bgp-ls-epe;
        }
        then accept;
    }
    term 3 {
        from protocol isis;
        then reject;
    }
```

```
    }
}
routing-options {
    router-id 10.6.6.6;
    autonomous-system 65200;
    rib-groups {
        color-to-inet3 {
            import-rib [ junos-rti-tc-<color>.inet.0 inet.3 ];
        }
    }
    forwarding-table {
        export pplb;
    }
}
protocols {
    bgp {
        group ibgp1 {
            type internal;
            local-address 10.6.6.6;
            family traffic-engineering {
                unicast;
            }
            export nlri2bgp_epe;
            neighbor 10.2.2.2;
            neighbor 10.5.5.5;
        }
        group ebgp1 {
            type external;
            family traffic-engineering {
                unicast;
            }
            export nlri2bgp_stat;
            neighbor 192.168.15.2 {
                peer-as 65300;
                egress-te-adj-segment epe_adj1_toR7 {
                    label 7167;
                    next-hop 192.168.15.2;
                    te-link-attribute {
                        te-metric 20;
                        igp-metric 10;
                        admin-group [ red brown ];
                    }
                }
```

```
            }
        }
    }
    express-segments {
        segment-set set6sr {
            membership-policy expresspolsr1;
        }
        traffic-engineering;
    }
    isis {
        interface ge-0/0/0.0 {
            level 2 {
                lan-neighbor 0100.0202.0202 {
                    ipv4-adjacency-segment {
                        protected label 6211;
                        unprotected label 6201;
                    }
                }
                post-convergence-lfa {
                    node-protection;
                }
            }
        }
        interface ge-0/0/1.0 {
            level 2 {
                lan-neighbor 0100.0303.0303 {
                    ipv4-adjacency-segment {
                        protected label 6311;
                        unprotected label 6301;
                    }
                }
                post-convergence-lfa {
                    node-protection;
                }
            }
        }
        interface ge-0/0/2.0 {
            level 2 {
                lan-neighbor 0100.0505.0505 {
                    ipv4-adjacency-segment {
                        protected label 6511;
                        unprotected label 6501;
                    }
```

```
                }
                post-convergence-lfa {
                    node-protection;
                }
            }
        }
        interface ge-0/0/3.0 {
            level 2 {
                post-convergence-lfa {
                    node-protection;
                }
            }
            passive;
        }
        interface fxp0.0 {
            disable;
        }
        interface lo0.0 {
            passive;
        }
        source-packet-routing {
            srgb start-label 800000 index-range 50000;
        }
        level 1 disable;
        level 2 wide-metrics-only;
        backup-spf-options {
            use-post-convergence-lfa;
            use-source-packet-routing;
        }
        traffic-engineering {
            l3-unicast-topology;
            advertisement always;
        }
        export prefix-sid;
    }
    mpls {
        traffic-engineering {
            database {
                import {
                    l3-unicast-topology {
                        bgp-link-state;
                    }
                    policy ted2nlri_epe_stat;
```

```
                }
                export {
                    l3-unicast-topology;
                }
            }
        }
        admin-groups {
            red 0;
            blue 1;
            green 2;
            yellow 3;
            orange 4;
            brown 5;
            black 6;
            pink 7;
        }
        label-range {
            static-label-range 1000 70000;
        }
        interface ge-0/0/0.0 {
            admin-group blue;
        }
        interface ge-0/0/1.0 {
            admin-group red;
        }
        interface ge-0/0/2.0 {
            admin-group brown;
        }
        interface ge-0/0/3.0 {
            admin-group [ red brown ];
        }
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
    source-packet-routing {
        no-chained-composite-next-hop;
        segment-list R6-R5-R4 {
            hop1 label 801005;
            hop2 label 801004;
        }
        source-routing-path lsp6to4_sr {
```

```
            to 10.4.4.4;
            color 1000;
            primary {
                R6-R5-R4;
            }
        }
        rib-group {
            ipv4-color {
                color-to-inet3;
            }
        }
    }
}
```

Configure R7 (PE2 router)

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the CLI User Guide.

To configure device R7:

1. Configure the network services mode as Enhanced IP. Enhanced IP sets the router's network services to enhanced Internet Protocol and uses enhanced mode capabilities.

   ```
   [edit]
   user@R7#set chassis network-services enhanced-ip
   ```

   After you configure the `enhanced-ip` statement and commit the configuration, the following warning message appears prompting you to reboot the router:

   ```
   'chassis'
   WARNING: Chassis configuration for network services has been changed. A system reboot is
   mandatory. Please reboot the system NOW. Continuing without a reboot might result in
   unexpected system behavior.
   commit complete
   ```

   The reboot brings up the FPCs on the router.

2. Configure the interfaces to enable IP, MPLS, and ISO transport.

```
user@R7#set interfaces ge-0/0/0 description To_R3
user@R7#set interfaces ge-0/0/0 vlan-tagging
user@R7#set interfaces ge-0/0/0 unit 0 vlan-id 1
user@R7#set interfaces ge-0/0/0 unit 0 family inet address 192.168.12.2/24
user@R7#set interfaces ge-0/0/0 unit 0 family iso
user@R7#set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:3700:10::7/64
user@R7#set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 8
user@R7#set interfaces ge-0/0/0 unit 1 vlan-id 2
user@R7#set interfaces ge-0/0/0 unit 1 family inet address 192.168.37.2/24
user@R7#set interfaces ge-0/0/0 unit 1 family iso
user@R7#set interfaces ge-0/0/0 unit 1 family inet6 address 2001:db8:3700:20::7/64
user@R7#set interfaces ge-0/0/0 unit 1 family mpls maximum-labels 8
user@R7#set interfaces ge-0/0/1 description To_R6
user@R7#set interfaces ge-0/0/1 vlan-tagging
user@R7#set interfaces ge-0/0/1 unit 0 vlan-id 1
user@R7#set interfaces ge-0/0/1 unit 0 family inet address 192.168.15.2/24
user@R7#set interfaces ge-0/0/1 unit 0 family iso
user@R7#set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:6700:10::7/64
user@R7#set interfaces ge-0/0/1 unit 0 family mpls maximum-labels 8
user@R7#set interfaces ge-0/0/1 unit 1 vlan-id 2
user@R7#set interfaces ge-0/0/1 unit 1 family inet address 192.168.67.2/24
user@R7#set interfaces ge-0/0/1 unit 1 family iso
user@R7#set interfaces ge-0/0/1 unit 1 family inet6 address 2001:db8:6700:20::7/64
user@R7#set interfaces ge-0/0/1 unit 1 family mpls maximum-labels 8
```

3. Configure the loopback interface to enable tunnel endpoints and service endpoints.

```
[edit]
user@R7#set interfaces lo0 unit 0 family inet address 10.7.7.7/32
user@R7#set interfaces lo0 unit 0 family inet address 10.7.7.71/32
user@R7#set interfaces lo0 unit 0 family iso address 49.0001.0007.0707.0700
user@R7#set interfaces lo0 unit 0 family inet6 address 2001:db8:abcd::07:07:07:07/128
user@R7#set interfaces lo0 unit 0 family inet6 address 2001:db8:abcd::7:7:7:71/128
```

4. Define import and export policies. For example, configure policies that export EPE TE links from the local TE database to **lsdist.0** and policies to import from **lsdist.0** into the local TE database. You can configure policies to advertise the BGP routes to a peer.

Route filter routs are advertised from external AS.

```
[edit]
user@R7#set policy-options policy-statement bgplsepe_rt_2_ted term 1 from protocol bgp
user@R7#set policy-options policy-statement bgplsepe_rt_2_ted term 1 then accept
user@R7#set policy-options policy-statement direct from protocol direct
user@R7#set policy-options policy-statement direct then accept
user@R7#set policy-options policy-statement mpath then multipath-resolve
user@R7#set policy-options policy-statement nlri2bgp term 1 from family traffic-engineering
user@R7#set policy-options policy-statement nlri2bgp term 1 then accept
user@R7#set policy-options policy-statement nlri2bgp_epe term 1 from family traffic-
engineering
user@R7#set policy-options policy-statement nlri2bgp_epe term 1 from protocol bgp-ls-epe
user@R7#set policy-options policy-statement nlri2bgp_epe term 1 then next-hop self
user@R7#set policy-options policy-statement nlri2bgp_epe term 1 then accept
user@R7#set policy-options policy-statement nlri2ted_bgp term 1 from protocol bgp
user@R7#set policy-options policy-statement nlri2ted_bgp term 1 then accept
user@R7#set policy-options policy-statement nlri2ted_igp term 1 from traffic-engineering
protocol isis-level-2
user@R7#set policy-options policy-statement nlri2ted_igp term 1 then accept
user@R7#set policy-options policy-statement payload_vpn_109 term 1 from route-filter
10.109.0.1/16 orlonger
user@R7#set policy-options policy-statement payload_vpn_109 term 1 then community add
color7000
user@R7#set policy-options policy-statement payload_vpn_109 term 1 then next-hop 10.7.7.7
user@R7#set policy-options policy-statement payload_vpn_109 term 1 then accept
user@R7#set policy-options policy-statement payload_vpn_110 term 1 from route-filter
10.110.0.1/16 orlonger
user@R7#set policy-options policy-statement payload_vpn_110 term 1 then community add
color7001
user@R7#set policy-options policy-statement payload_vpn_110 term 1 then next-hop 10.7.7.7
user@R7#set policy-options policy-statement payload_vpn_110 term 1 then accept
user@R7#set policy-options policy-statement pplb then load-balance per-packet
user@R7#set policy-options policy-statement ted2nlri term 1 from protocol bgp-ls-epe
user@R7#set policy-options policy-statement ted2nlri term 1 then accept
user@R7#set policy-options community color7000 members color:0:7000
user@R7#set policy-options community color7001 members color:0:7001
user@R7#set policy-options resolution-map map1 mode ip-color
```

5. Configure routing options to identify the router in the domain.

```
[edit]
user@R7#set routing-options router-id 10.7.7.7
user@R7#set routing-options autonomous-system 65300
user@R7#set routing-options static route 10.100.100.101/32 next-hop 10.100.100.100
user@R7#set routing-options static route 10.100.100.101/32 resolve
user@R7#set routing-options forwarding-table export pplb
```

6. Configure BGP to enable BGP-LS route advertisement for peer and define the EPE links. Since express segment is an internal TE link, this configuration creates an external TE link.

```
[edit]
user@R7#set protocols bgp group ebgp1 type external
user@R7#set protocols bgp group ebgp1 multihop ttl 100
user@R7#set protocols bgp group ebgp1 family inet unicast
user@R7#set protocols bgp group ebgp1 family inet-vpn unicast
user@R7#set protocols bgp group ebgp1 family traffic-engineering unicast
user@R7#set protocols bgp group ebgp1 export [ nlri2bgp_epe payload_vpn_109
payload_vpn_110 ]
user@R7#set protocols bgp group ebgp1 neighbor 192.168.12.1 peer-as 65200
user@R7#set protocols bgp group ebgp1 neighbor 192.168.12.1 egress-te-adj-segment
epe_adj1_toR3 label 8173
user@R7#set protocols bgp group ebgp1 neighbor 192.168.12.1 egress-te-adj-segment
epe_adj1_toR3 next-hop 192.168.12.1
user@R7#set protocols bgp group ebgp1 neighbor 192.168.12.1 egress-te-adj-segment
epe_adj1_toR3 te-link-attribute te-metric 20
user@R7#set protocols bgp group ebgp1 neighbor 192.168.12.1 egress-te-adj-segment
epe_adj1_toR3 te-link-attribute igp-metric 10
user@R7#set protocols bgp group ebgp1 neighbor 192.168.12.1 egress-te-adj-segment
epe_adj1_toR3 te-link-attribute admin-group red
user@R7#set protocols bgp group ebgp1 neighbor 192.168.12.1 egress-te-adj-segment
epe_adj1_toR3 te-link-attribute admin-group brown
user@R7#set protocols bgp group ebgp1 neighbor 192.168.15.1 peer-as 200
user@R7#set protocols bgp group ebgp1 neighbor 192.168.15.1 egress-te-adj-segment
epe_adj1_toR6 label 8176
user@R7#set protocols bgp group ebgp1 neighbor 192.168.15.1 egress-te-adj-segment
epe_adj1_toR6 next-hop 192.168.15.1
user@R7#set protocols bgp group ebgp1 neighbor 192.168.15.1 egress-te-adj-segment
epe_adj1_toR6 te-link-attribute te-metric 20
user@R7#set protocols bgp group ebgp1 neighbor 192.168.15.1 egress-te-adj-segment
```

```
epe_adj1_toR6 te-link-attribute igp-metric 10
user@R7#set protocols bgp group ebgp1 neighbor 192.168.15.1 egress-te-adj-segment
epe_adj1_toR6 te-link-attribute admin-group red
user@R7#set protocols bgp group ebgp1 neighbor 192.168.15.1 egress-te-adj-segment
epe_adj1_toR6 te-link-attribute admin-group brown
user@R7#set protocols bgp group ebgp1 neighbor 10.100.100.101 local-address 10.7.7.71
user@R7#set protocols bgp group ebgp1 neighbor 10.100.100.101 peer-as 100
user@R7#set protocols bgp group ebgp1 vpn-apply-export
user@R7#set protocols bgp group to-CE1 type external
user@R7#set protocols bgp group to-CE1 local-address 192.168.50.1
user@R7#set protocols bgp group to-CE1 neighbor 192.168.50.2 family inet unicast
user@R7#set protocols bgp group to-CE1 neighbor 192.168.50.2 family inet6 unicast
user@R7#set protocols bgp group to-CE1 neighbor 192.168.50.2 peer-as 65007
user@R7#set protocols bgp group to-CE1 neighbor 192.168.50.2 local-as 65300
```

7. Configure IS-IS protocol.

```
[edit]user@R7#set protocols isis interface fxp0.0 disable
user@R7#set protocols isis interface lo0.0 passive
user@R7#set protocols isis level 1 disable
user@R7#set protocols isis level 2 wide-metrics-only
user@R7#set protocols isis traffic-engineering l3-unicast-topology
user@R7#set protocols isis traffic-engineering advertisement always
```

8. Enable import and export of traffic engineering database parameters using policies.

```
[edit]
user@R7#set protocols mpls traffic-engineering database import l3-unicast-topology bgp-link-
state
user@R7#set protocols mpls traffic-engineering database import policy ted2nlri
user@R7#set protocols mpls traffic-engineering database export policy nlri2ted_bgp
user@R7#set protocols mpls traffic-engineering database export l3-unicast-topology
```

9. Configure MPLS administrative group policies for LSP path computation.

```
[edit]
user@R7#set protocols mpls admin-groups red 0
user@R7#set protocols mpls admin-groups blue 1
user@R7#set protocols mpls admin-groups green 2
user@R7#set protocols mpls admin-groups yellow 3
```

```
user@R7#set protocols mpls admin-groups orange 4
user@R7#set protocols mpls admin-groups brown 5
user@R7#set protocols mpls admin-groups black 6
user@R7#set protocols mpls admin-groups pink 7
```

10. Configure the MPLS label range to assign static labels for the EPE links.

```
[edit]
user@R7#set protocols mpls label-range static-label-range 1000 70000
```

11. Configure MPLS with interface and include administrative groups.

```
[edit]
user@R7#set protocols mpls interface ge-0/0/1.0 admin-group [ red brown ]
user@R7#set protocols mpls interface ge-0/0/0.0 admin-group [ red brown ]
user@R7#set protocols mpls interface all
user@R7#set protocols mpls interface fxp0.0 disable
```

12. Configure SR-TE policies on the ingress router to enable end-to-end SR-TE policy.

```
[edit]
user@R7#set protocols source-packet-routing compute-profile compute1 no-label-stack-
compression
user@R7#set protocols source-packet-routing source-routing-path computelsp1 to
10.100.100.100
user@R7#set protocols source-packet-routing source-routing-path computelsp1 install
10.100.100.101
user@R7#set protocols source-packet-routing source-routing-path computelsp1 primary p1
compute compute1
```

Results

From configuration mode, confirm your configuration by entering the `show chassis`, `show interfaces`, `show policy-options`, `show routing-options`, and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
chassis {
    network-services enhanced-ip;
```

```
}
interfaces {
    ge-0/0/0 {
        description To_R3;
        vlan-tagging;
        unit 0 {
            vlan-id 1;
            family inet {
                address 192.168.12.2/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:3700:10::7/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
        unit 1 {
            vlan-id 2;
            family inet {
                address 192.168.37.2/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:3700:20::7/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
    }
    ge-0/0/1 {
        description To_R6;
        vlan-tagging;
        unit 0 {
            vlan-id 1;
            family inet {
                address 192.168.15.2/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:6700:10::7/64;
```

```
            }
            family mpls {
                maximum-labels 8;
            }
        }
        unit 1 {
            vlan-id 2;
            family inet {
                address 192.168.67.2/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:6700:20::7/64;
            }
            family mpls {
                maximum-labels 8;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.7.7.7/32;
                address 10.7.7.71/32;
            }
            family iso {
                address 49.0001.0007.0707.0700;
            }
            family inet6 {
                address 2001:db8:abcd::07:07:07:07/128;
                address 2001:db8:abcd::7:7:7:71/128;
            }
        }
    }
}
policy-options {
    policy-statement bgplsepe_rt_2_ted {
        term 1 {
            from protocol bgp;
            then accept;
        }
    }
    policy-statement direct {
```

```
        from protocol direct;
        then accept;
    }
    policy-statement mpath {
        then multipath-resolve;
    }
    policy-statement nlri2bgp {
        term 1 {
            from family traffic-engineering;
            then accept;
        }
    }
    policy-statement nlri2bgp_epe {
        term 1 {
            from {
                family traffic-engineering;
                protocol bgp-ls-epe;
            }
            then {
                next-hop self;
                accept;
            }
        }
    }
    policy-statement nlri2ted_bgp {
        term 1 {
            from protocol bgp;
            then accept;
        }
    }
    policy-statement nlri2ted_igp {
        term 1 {
            from {
                traffic-engineering {
                    protocol isis-level-2;
                }
            }
            then accept;
        }
    }
    policy-statement payload_vpn_109 {
        term 1 {
            from {
```

```
                route-filter 109.0.0.1/16 orlonger;
            }
            then {
                community add color7000;
                next-hop 10.7.7.7;
                accept;
            }
        }
    }
    policy-statement payload_vpn_110 {
        term 1 {
            from {
                route-filter 10.110.0.1/16 orlonger;
            }
            then {
                community add color7001;
                next-hop 10.7.7.7;
                accept;
            }
        }
    }
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
    }
    policy-statement ted2nlri {
        term 1 {
            from protocol bgp-ls-epe;
            then accept;
        }
    }
    community color7000 members color:0:7000;
    community color7001 members color:0:7001;
    resolution-map map1 {
        mode ip-color;
    }
}
routing-options {
    router-id 10.7.7.7;
    autonomous-system 65300;
    static {
        route1 0.100.100.101/32 {
```

```
            next-hop 10.100.100.100;
            resolve;
        }
    }
    forwarding-table {
        export pplb;
    }
}
protocols {
    bgp {
        group ebgp1 {
            type external;
            multihop {
                ttl 100;
            }
            family inet {
                unicast;
            }
            family inet-vpn {
                unicast;
            }
            family traffic-engineering {
                unicast;
            }
            export [ nlri2bgp_epe payload_vpn_109 payload_vpn_110 ];
            neighbor 192.168.12.1 {
                peer-as 65200;
                egress-te-adj-segment epe_adj1_toR3 {
                    label 8173;
                    next-hop 192.168.12.1;
                    te-link-attribute {
                        te-metric 20;
                        igp-metric 10;
                        admin-group [ red brown ];
                    }
                }
            }
            neighbor 192.168.15.1 {
                peer-as 65200;
                egress-te-adj-segment epe_adj1_toR6 {
                    label 8176;
                    next-hop 192.168.15.1;
                    te-link-attribute {
```

```
                        te-metric 20;
                        igp-metric 10;
                        admin-group [ red brown ];
                    }
                }
            }
            neighbor 10.100.100.101 {
                local-address 10.7.7.71;
                peer-as 65100;
            }
            vpn-apply-export;
        }
        group to-CE1 {
            type external;
            local-address 192.168.50.1;
            neighbor 192.168.50.2 {
                family inet {
                    unicast;
                }
                family inet6 {
                    unicast;
                }
                peer-as 65007;
                local-as 65300;
            }
        }
    }
    isis {
        interface fxp0.0 {
            disable;
        }
        interface lo0.0 {
            passive;
        }
        level 1 disable;
        level 2 wide-metrics-only;
        traffic-engineering {
            l3-unicast-topology;
            advertisement always;
        }
    }
    mpls {
        traffic-engineering {
```

```
            database {
                import {
                    l3-unicast-topology {
                        bgp-link-state;
                    }
                    policy ted2nlri;
                }
                export {
                    policy nlri2ted_bgp;
                    l3-unicast-topology;
                }
            }
        }
        admin-groups {
            red 0;
            blue 1;
            green 2;
            yellow 3;
            orange 4;
            brown 5;
            black 6;
            pink 7;
        }
        label-range {
            static-label-range 1000 70000;
        }
        interface ge-0/0/1.0 {
            admin-group [ red brown ];
        }
        interface ge-0/0/0.0 {
            admin-group [ red brown ];
        }
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
    source-packet-routing {
        compute-profile compute1 {
            no-label-stack-compression;
        }
        source-routing-path computelsp1 {
            to 10.100.100.100;
```

```
            install 10.100.100.101;
            primary {
                p1 {
                    compute {
                        compute1;
                    }
                }
            }
        }
    }
```

Verification

IN THIS SECTION

- Verify the Express Segment | **1199**
- Verify the Express Segment Advertisements | **1201**
- Verify the TE Topology Information | **1208**

To confirm that the configuration is working properly, perform the following tasks:

Verify the Express Segment

Purpose

Verify that the express segments are created correctly.

Action

From operational mode, run the following commands:

- `show express-segments detail`—Verify whether the express segments are created.
- `show ted database topology-type express-segments detail`—Verify that the newly created express segments are inserted into the TE database.

- `show route table mpls.0 protocol express-segments`—Verify whether the forwarding entries have been created.

```
user@R1>show express-segments detail

Name: set1sr-10.3.3.3
  To: 10.3.3.3, Type: Dynamic (Set: set1sr)
  Label: 16 (Route installed in mpls.0, TED entry added)
  Status: Up (ElapsedTime: 5d 20:37:08)
  LinkAttributes:
    LocalID: 2147483649
    TE-Metric: 20, IGP-Metric: 20
    BW: 0bps
  UnderlayPaths: 1
    SRTE LSP: lsp1to3_sr
      TE-Metric: 0, IGP-Metric: 0
      BW: 0bps
```

On R1

```
user@R1>show ted database topology-type express-segments detail

TED database: 18 ISIS nodes 7 INET nodes 0 INET6 nodes
NodeID: R1.00(10.1.1.1)
  Type: Rtr, Age: 774 secs, LinkIn: 4, LinkOut: 6
  Protocol: EXPRESS-SEG(0)
    To: R3.00(10.3.3.3), Local: 10.1.1.1, Remote: 10.3.3.3
      Local interface index: 2147483649, Remote interface index: 0
      Link name: set1sr-10.3.3.3
NodeID: R3.00(10.3.3.3)
  Type: Rtr, Age: 580 secs, LinkIn: 4, LinkOut: 3
  Protocol: EXPRESS-SEG(0)
```

On R1

```
user@R1>show route table mpls.0 protocol express-segments

mpls.0: 33 destinations, 33 routes (33 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

16                 *[EXPRESS-SEG/6] 5d 14:13:37, metric 1
```

```
                    >  to 192.168.3.2 via ge-0/0/2.0, Swap 801003
                       to 192.168.21.2 via ge-0/0/2.1, Swap 801003
```

Meaning

- In the `show express-segments detail` output, you can see the name of the express segment (**set1sr-10.3.3.3**), express segment label (**16**), and the underlay LSP (**lsp1to3_sr**).
- In the `show ted database topology-type express-segments detail` output, you can see the express segment entries are inserted into the TE database. The express segments (virtual TE links) are dynamically created. The protocol used is **EXPRESS-SEG(0)**.
- In the `show route table mpls.0 protocol express-segments` output, you can see the express segment label (**16**). Because the express segment is a construct that relies on the underlay LSPs, the express segment label gets swapped to the underlay SR-TE labels (**801003**).

Verify the Express Segment Advertisements

Purpose

Verify that the originating node advertises express segments to its eBGP/iBGP LS neighbors.

Action

From operational mode, run the following commands:

- `show route table lsdist.0`—Verify that the express segments in the RIB BGP-LS are being advertised.
- `show route advertising-protocol bgp` *neighbor*—Verify that the express segments are sent to the eBGP/iBGP LS neighbors.

```
user@R1>show route table lsdist.0

lsdist.0: 23 destinations, 40 routes (23 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

NODE { AS:65200 IPv4:10.1.1.1 STATIC:0 }/1216
                   *[EXPRESS-SEG/6] 5d 14:50:56
                       Fictitious
NODE { AS:65200 IPv4:10.3.3.3 STATIC:0 }/1216
                   *[EXPRESS-SEG/6] 5d 14:50:56
                       Fictitious
```

```
NODE { AS:65100 IPv4:10.1.1.1 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 5d 14:51:38, localpref 100
                      AS path: 100 I, validation-state: unverified
                    >  to 192.168.1.1 via ge-0/0/0.0
NODE { AS:100 IPv10:4.4.4.4 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 5d 14:51:38, localpref 100
                      AS path: 100 I, validation-state: unverified
                    >  to 192.168.1.1 via ge-0/0/0.0
NODE { AS:65100 IPv4:10.100.100.100 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 5d 14:51:38, localpref 100
                      AS path: 100 I, validation-state: unverified
                    >  to 192.168.1.1 via ge-0/0/0.0
NODE { AS:65200 IPv4:10.1.1.1 BGP-LS-EPE:0 }/1216
                   *[BGP-LS-EPE/170] 5d 14:51:38
                       Fictitious
NODE { AS:65200 IPv4:10.3.3.3 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 5d 14:51:04, localpref 100, from 10.2.2.2
                      AS path: I, validation-state: unverified
                    >  to 192.168.3.2 via ge-0/0/2.0
                       to 192.168.21.2 via ge-0/0/2.1
                       to 192.168.5.2 via ge-0/0/4.0
                    [BGP/170] 5d 14:51:00, localpref 100, from 10.5.5.5
                      AS path: I, validation-state: unverified
                    >  to 192.168.3.2 via ge-0/0/2.0
                       to 192.168.21.2 via ge-0/0/2.1
                       to 192.168.5.2 via ge-0/0/4.0
NODE { AS:65200 IPv4:10.4.4.4 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 5d 14:50:53, localpref 100, from 10.2.2.2
                      AS path: I, validation-state: unverified
                    >  to 192.168.4.2 via ge-0/0/3.0
                       to 192.168.5.2 via ge-0/0/4.0
                    [BGP/170] 5d 14:50:53, localpref 100, from 10.5.5.5
                      AS path: I, validation-state: unverified
                    >  to 192.168.4.2 via ge-0/0/3.0
                       to 192.168.5.2 via ge-0/0/4.0
NODE { AS:65200 IPv4:10.6.6.6 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 5d 14:51:04, localpref 100, from 10.2.2.2
                      AS path: I, validation-state: unverified
                    >  to 192.168.3.2 via ge-0/0/2.0
                       to 192.168.21.2 via ge-0/0/2.1
                       to 192.168.5.2 via ge-0/0/4.0
                    [BGP/170] 5d 14:51:00, localpref 100, from 10.5.5.5
                      AS path: I, validation-state: unverified
```

```
                    >  to 192.168.3.2 via ge-0/0/2.0
                       to 192.168.21.2 via ge-0/0/2.1
                       to 192.168.5.2 via ge-0/0/4.0
NODE { AS:65200 IPv4:7.7.7.7 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 5d 14:51:04, localpref 100, from 10.2.2.2
                      AS path: I, validation-state: unverified
                    >  to 192.168.3.2 via ge-0/0/2.0
                       to 192.168.21.2 via ge-0/0/2.1
                       to 192.168.5.2 via ge-0/0/4.0
                    [BGP/170] 5d 14:51:00, localpref 100, from 10.5.5.5
                      AS path: I, validation-state: unverified
                    >  to 192.168.3.2 via ge-0/0/2.0
                       to 192.168.21.2 via ge-0/0/2.1
                       to 192.168.5.2 via ge-0/0/4.0
NODE { AS:65200 IPv4:10.100.100.100 BGP-LS-EPE:0 }/1216
                   *[BGP-LS-EPE/170] 5d 14:51:38
                       Fictitious
NODE { AS:65300 IPv4:10.3.3.3 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 5d 14:51:04, localpref 100, from 10.2.2.2
                      AS path: 300 I, validation-state: unverified
                    >  to 192.168.3.2 via ge-0/0/2.0
                       to 192.168.21.2 via ge-0/0/2.1
                       to 192.168.5.2 via ge-0/0/4.0
                    [BGP/170] 5d 14:51:00, localpref 100, from 10.5.5.5
                      AS path: 65300 I, validation-state: unverified
                    >  to 192.168.3.2 via ge-0/0/2.0
                       to 192.168.21.2 via ge-0/0/2.1
                       to 192.168.5.2 via ge-0/0/4.0
                    [BGP/170] 5d 14:51:38, localpref 100
                      AS path: 65100 65300 I, validation-state: unverified
                    >  to 192.168.1.1 via ge-0/0/0.0
NODE { AS:65300 IPv4:10.6.6.6 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 5d 14:51:04, localpref 100, from 10.2.2.2
                      AS path: 65300 I, validation-state: unverified
                    >  to 192.168.3.2 via ge-0/0/2.0
                       to 192.168.21.2 via ge-0/0/2.1
                       to 192.168.5.2 via ge-0/0/4.0
                    [BGP/170] 5d 14:51:00, localpref 100, from 10.5.5.5
                      AS path: 65300 I, validation-state: unverified
                    >  to 192.168.3.2 via ge-0/0/2.0
                       to 192.168.21.2 via ge-0/0/2.1
                       to 192.168.5.2 via ge-0/0/4.0
                    [BGP/170] 5d 14:51:38, localpref 100
```

```
                      AS path: 65100 65300 I, validation-state: unverified
                    >  to 192.168.1.1 via ge-0/0/0.0
NODE { AS:65300 IPv4:10.7.7.7 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 5d 14:51:04, localpref 100, from 10.2.2.2
                      AS path: 65300 I, validation-state: unverified
                    >  to 192.168.3.2 via ge-0/0/2.0
                       to 192.168.21.2 via ge-0/0/2.1
                       to 192.168.5.2 via ge-0/0/4.0
                    [BGP/170] 5d 14:51:00, localpref 100, from 10.5.5.5
                      AS path: 65300 I, validation-state: unverified
                    >  to 192.168.3.2 via ge-0/0/2.0
                       to 192.168.21.2 via ge-0/0/2.1
                       to 192.168.5.2 via ge-0/0/4.0
                    [BGP/170] 5d 14:51:38, localpref 100
                      AS path: 65100 65300 I, validation-state: unverified
                    >  to 192.168.1.1 via ge-0/0/0.0
LINK { Local { AS:65200 IPv4:10.1.1.1 }.{ IfIndex:2147483649 } Remote { AS:200
IPv4
                        :10.3.3.3 }.{ IfIndex:0 } STATIC:0 }/1216
                   *[EXPRESS-SEG/6] 5d 14:50:56
                       Fictitious
LINK { Local { AS:65100 IPv4:10.100.100.100 }.{ IfIndex:333 } Remote { AS:65200
IPv4
                        4:10.1.1.1 }.{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                   *[BGP/170] 5d 14:51:38, localpref 100
                      AS path: 65100 I, validation-state: unverified
                    >  to 192.168.1.1 via ge-0/0/0.0
LINK { Local { AS:65100 IPv4:10.100.100.100 }.{ IfIndex:336 } Remote { AS:65200
IPv4
                        4:10.4.4.4 }.{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                   *[BGP/170] 5d 14:51:38, localpref 100
                      AS path: 65100 I, validation-state: unverified
                    >  to 192.168.1.1 via ge-0/0/0.0
LINK { Local { AS:65200 IPv4:10.1.1.1 }.{ IfIndex:333 } Remote { AS:65100
IPv4:10.10
                               0.10.100 }.{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                   *[BGP-LS-EPE/170] 5d 14:51:38
                       Fictitious
LINK { Local { AS:65200 IPv4:10.3.3.3 }.{ IfIndex:342 } Remote { AS:65300
IPv4:10.7.7.
                                 7 }.{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                   *[BGP/170] 5d 14:51:04, localpref 100, from 10.2.2.2
                      AS path: I, validation-state: unverified
```

```
                    >  to 192.168.3.2 via ge-0/0/2.0
                       to 192.168.21.2 via ge-0/0/2.1
                       to 192.168.5.2 via ge-0/0/4.0
                    [BGP/170] 5d 14:51:00, localpref 100, from 10.5.5.5
                      AS path: I, validation-state: unverified
                    >  to 192.168.3.2 via ge-0/0/2.0
                       to 192.168.21.2 via ge-0/0/2.1
                       to 192.168.5.2 via ge-0/0/4.0
LINK { Local { AS:65200 IPv4:10.4.4.4 }.{ IfIndex:333 } Remote { AS:65100
IPv4:10.10
                               0.100.100 }.{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                  *[BGP/170] 5d 14:50:53, localpref 100, from 10.2.2.2
                      AS path: I, validation-state: unverified
                    >  to 192.168.4.2 via ge-0/0/3.0
                       to 192.168.5.2 via ge-0/0/4.0
                    [BGP/170] 5d 14:50:53, localpref 100, from 10.5.5.5
                      AS path: I, validation-state: unverified
                    >  to 192.168.4.2 via ge-0/0/3.0
                       to 192.168.5.2 via ge-0/0/4.0
LINK { Local { AS:65200 IPv4:10.6.6.6 }.{ IfIndex:341 } Remote { AS:65300
IPv4:10.7.7.
                                 7 }.{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                  *[BGP/170] 5d 14:51:04, localpref 100, from 10.2.2.2
                      AS path: I, validation-state: unverified
                    >  to 192.168.3.2 via ge-0/0/2.0
                       to 192.168.21.2 via ge-0/0/2.1
                       to 192.168.5.2 via ge-0/0/4.0
                    [BGP/170] 5d 14:51:00, localpref 100, from 10.5.5.5
                      AS path: I, validation-state: unverified
                    >  to 192.168.3.2 via ge-0/0/2.0
                       to 192.168.21.2 via ge-0/0/2.1
                       to 192.168.5.2 via ge-0/0/4.0
LINK { Local { AS:65300 IPv4:10.7.7.7 }.{ IfIndex:333 } Remote { AS:200
IPv4:10.3.3.
                                 3 }.{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                  *[BGP/170] 5d 14:51:04, localpref 100, from 10.2.2.2
                      AS path: 65300 I, validation-state: unverified
                    >  to 192.168.3.2 via ge-0/0/2.0
                       to 192.168.21.2 via ge-0/0/2.1
                       to 192.168.5.2 via ge-0/0/4.0
                    [BGP/170] 5d 14:51:00, localpref 100, from 10.5.5.5
                      AS path: 65300 I, validation-state: unverified
                    >  to 192.168.3.2 via ge-0/0/2.0
```

```
                        to 192.168.21.2 via ge-0/0/2.1
                        to 192.168.5.2 via ge-0/0/4.0
                     [BGP/170] 5d 14:51:38, localpref 100
                       AS path: 65100 65300 I, validation-state: unverified
                     >  to 192.168.1.1 via ge-0/0/0.0
LINK { Local { AS:65300 IPv4:10.7.7.7 }.{ IfIndex:336 } Remote { AS:65200
IPv4:10.6.6.
                                  6 }.{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                    *[BGP/170] 5d 14:51:04, localpref 100, from 10.2.2.2
                       AS path: 65300 I, validation-state: unverified
                     >  to 192.168.3.2 via ge-0/0/2.0
                        to 192.168.21.2 via ge-0/0/2.1
                        to 192.168.5.2 via ge-0/0/4.0
                     [BGP/170] 5d 14:51:00, localpref 100, from 10.5.5.5
                       AS path: 65300 I, validation-state: unverified
                     >  to 192.168.3.2 via ge-0/0/2.0
                        to 192.168.21.2 via ge-0/0/2.1
                        to 192.168.5.2 via ge-0/0/4.0
                     [BGP/170] 5d 14:51:38, localpref 100
                       AS path: 65100 65300 I, validation-state: unverified
                     >  to 192.168.1.1 via ge-0/0/0.0
```

On R1

```
user@R1>show route advertising-protocol bgp 10.2.2.2

lsdist.0: 23 destinations, 40 routes (23 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
  NODE { AS:65100 IPv4:10.1.1.1 BGP-LS-EPE:0 }/1216
*                         192.168.1.1                  100        65100 I
                Area border router: No
                External router: No
                Attached: No
                Overload: No
  Prefix                  Nexthop              MED     Lclpref    AS path
  NODE { AS:65100 IPv4:10.4.4.4 BGP-LS-EPE:0 }/1216
*                         192.168.1.1                  100        65100 I
                Area border router: No
                External router: No
                Attached: No
                Overload: No
```

```
  Prefix                  Nexthop              MED     Lclpref    AS path
  NODE { AS:65100 IPv4:10.100.100.100 BGP-LS-EPE:0 }/1216
*                         192.168.1.1                  100        65100 I
                Area border router: No
                External router: No
                Attached: No
                Overload: No
  Prefix                  Nexthop              MED     Lclpref    AS path
  NODE { AS:65200 IPv4:10.1.1.1 BGP-LS-EPE:0 }/1216
*                         Self                          100        I
                Area border router: No
                External router: No
                Attached: No
                Overload: No
  Prefix                  Nexthop              MED     Lclpref    AS path
  NODE { AS:65200 IPv4:10.100.100.100 BGP-LS-EPE:0 }/1216
*                         Self                          100        I
                Area border router: No
                External router: No
                Attached: No
                Overload: No
  Prefix                  Nexthop              MED     Lclpref    AS path
  LINK { Local { AS:65100 IPv4:10.100.100.100 }.{ IfIndex:333 } Remote { AS:65200
IPv4:10.1.1.1 }.{ IfIndex:0 } BGP-LS-EPE:0 }/1216
*                         192.168.1.1                  100        65100 I
                Color: 33
                Metric: 10
                TE Metric: 20
                Link name: epe_adj1_toR1
                Label: 7101, Flags: 0xd0, Weight: 0
  Prefix                  Nexthop              MED     Lclpref    AS path
  LINK { Local { AS:65100 IPv4:10.100.100.100 }.{ IfIndex:336 } Remote { AS:65200
IPv4:10.4.4.4 }.{ IfIndex:0 } BGP-LS-EPE:0 }/1216
*                         192.168.1.1                  100        65100 I
                Color: 33
                Metric: 10
                TE Metric: 20
                Link name: epe_adj1_toR4
                Label: 7104, Flags: 0xd0, Weight: 0
  Prefix                  Nexthop              MED     Lclpref    AS path
  LINK { Local { AS:65200 IPv4:10.1.1.1 }.{ IfIndex:333 } Remote { AS:65100
IPv4:10.100.100.100 }.{ IfIndex:0 } BGP-LS-EPE:0 }/1216
*                         Self                          100        I
```

```
                Color: 33
                Metric: 10
                TE Metric: 20
                Link name: epe_adj1_toR0
                Label: 8110, Flags: 0xd0, Weight: 0
```

Meaning

- In the `show route table lsdist.0` output, BGP advertises the routes in the routing table. The routing table is created from the TE database. You can see the express segments (**EXPRESS-SEG/6**) links and the EPE links (**BGP-LS-EPE:0 }/1216**).
- In the `show route advertising-protocol bgp 10.2.2.2` output, you can see what R1 is advertising to. The express segments are inserted into the TE database, which is copied to RIB. BGP-LS advertises the RIB to the peer router. On the peer, the received RIB information is copied into the local database. The policy in this example only advertises express segments and EPE segments.

Verify the TE Topology Information

Purpose

Verify that the ingress nodes receive TE topology information through eBGP/iBGP LS.

Action

From operational mode, run the following commands:

- `show route receive-protocol bgp` *neighbor*—Verify that the express segments are received from eBGP/iBGP LS neighbors.
- `show route table lsdist.0`—Verify that the express segments are in the BGP-LS RIB.
- `show ted database topology-type l3-unicast detail`—Verify that the express segments are imported into the ingress router's TE database.
- `show spring-traffic-engineering lsp`—Verify that the end-to-end SR policy has been successfully computed and installed.

On R0

```
user@R0>show route receive-protocol bgp 10.9.148.59
…
```

```
inet.0: 14 destinations, 14 routes (14 active, 0 holddown, 0 hidden)

inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

iso.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

mpls.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)

inet6.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)

lsdist.0: 32 destinations, 61 routes (32 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
  NODE { AS:65200 IPv4:10.1.1.1 STATIC:0 }/1216
*                         192.168.1.2                              65200 I
                Area border router: No
                External router: No
                Attached: No
                Overload: No
  Prefix                  Nexthop              MED     Lclpref    AS path
  NODE { AS:65200 IPv4:10.3.3.3 STATIC:0 }/1216
*                         192.168.1.2                              65200 I
                Area border router: No
                External router: No
                Attached: No
                Overload: No
  Prefix                  Nexthop              MED     Lclpref    AS path
  NODE { AS:65200 IPv4:10.3.3.3 BGP-LS-EPE:0 }/1216
                          192.168.1.2                              65200 I
                Area border router: No
                External router: No
                Attached: No
                Overload: No
  Prefix                  Nexthop              MED     Lclpref    AS path
  NODE { AS:65200 IPv4:10.4.4.4 BGP-LS-EPE:0 }/1216
*                         192.168.1.2                              65200 I
                Area border router: No
                External router: No
                Attached: No
                Overload: No
  Prefix                  Nexthop              MED     Lclpref    AS path
  NODE { AS:65200 IPv4:10.6.6.6 BGP-LS-EPE:0 }/1216
                          192.168.1.2                              65200 I
                Area border router: No
```

```
                External router: No
                Attached: No
                Overload: No
  Prefix                   Nexthop              MED     Lclpref    AS path
  NODE { AS:65200 IPv4:10.7.7.7 BGP-LS-EPE:0 }/1216
                           192.168.1.2                              65200 I
                Area border router: No
                External router: No
                Attached: No
                Overload: No
  Prefix                   Nexthop              MED     Lclpref    AS path
  NODE { AS:65300 IPv4:10.3.3.3 BGP-LS-EPE:0 }/1216
                           192.168.1.2                              65200 65300 I
                Area border router: No
                External router: No
                Attached: No
                Overload: No
  Prefix                   Nexthop              MED     Lclpref    AS path
  NODE { AS:65300 IPv4:10.6.6.6 BGP-LS-EPE:0 }/1216
                           192.168.1.2                              65200 65300 I
                Area border router: No
                External router: No
                Attached: No
                Overload: No
  Prefix                   Nexthop              MED     Lclpref    AS path
  NODE { AS:65300 IPv4:10.7.7.7 BGP-LS-EPE:0 }/1216
                           192.168.1.2                              65200 65300 I
                Area border router: No
                External router: No
                Attached: No
                Overload: No
  Prefix                   Nexthop              MED     Lclpref    AS path
  LINK { Local { AS:65200 IPv4:10.1.1.1 }.{ IfIndex:2147483649 } Remote { AS:65200
IPv4:10.3.3.3 }.{ IfIndex:0 } STATIC:0 }/1216
*                          192.168.1.2                              65200 I
                Metric: 20
                TE Metric: 20
                Link name: set1sr-10.3.3.3
                Label: 16, Flags: 0x60, Weight: 1
  Prefix                   Nexthop              MED     Lclpref    AS path
  LINK { Local { AS:65200 IPv4:10.3.3.3 }.{ IfIndex:342 } Remote { AS:65300 IPv4:10.7.7.7 }.
{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                           192.168.1.2                              65200 I
```

```
                Color: 33
                Metric: 10
                TE Metric: 20
                Link name: epe_adj1_toR7
                Label: 7137, Flags: 0xd0, Weight: 0
  Prefix                  Nexthop              MED     Lclpref    AS path
  LINK { Local { AS:65200 IPv4:10.4.4.4 }.{ IfIndex:333 } Remote { AS:65100
IPv4:10.100.100.100 }.{ IfIndex:0 } BGP-LS-EPE:0 }/1216
*                         192.168.1.2                               65200 I
                Color: 33
                Metric: 10
                TE Metric: 20
                Link name: epe_adj1_toR0
                Label: 8140, Flags: 0xd0, Weight: 0
  Prefix                  Nexthop              MED     Lclpref    AS path
  LINK { Local { AS:65200 IPv4:10.6.6.6 }.{ IfIndex:341 } Remote { AS:65300 IPv4:10.7.7.7 }.
{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                          192.168.1.2                               65200 I
                Color: 33
                Metric: 10
                TE Metric: 20
                Link name: epe_adj1_toR7
                Label: 7167, Flags: 0xd0, Weight: 0
  Prefix                  Nexthop              MED     Lclpref    AS path
  LINK { Local { AS:65300 IPv4:10.7.7.7 }.{ IfIndex:333 } Remote { AS:65200 IPv4:10.3.3.3 }.
{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                          192.168.1.2                               65200 65300 I
                Color: 33
                Metric: 10
                TE Metric: 20
                Link name: epe_adj1_toR3
                Label: 8173, Flags: 0xd0, Weight: 0
  Prefix                  Nexthop              MED     Lclpref    AS path
  LINK { Local { AS:65300 IPv4:10.7.7.7 }.{ IfIndex:336 } Remote { AS:65200 IPv4:10.6.6.6 }.
{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                          192.168.1.2                               65200 65300 I
                Color: 33
                Metric: 10
                TE Metric: 20
                Link name: epe_adj1_toR6
                Label: 8176, Flags: 0xd0, Weight: 0

lsdist.1: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
```

```
junos-rti-tc-<color>.inet.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
```

On R0

```
user@R0>show route table lsdist.0

lsdist.0: 32 destinations, 61 routes (32 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

NODE { AS:65100 ISO:0100.0a0a.0a0a.00 ISIS-L2:0 }/1216
                   *[IS-IS/18] 5d 18:02:43
                       Fictitious
NODE { AS:65200 IPv4:10.1.1.1 STATIC:0 }/1216
                   *[BGP/170] 5d 16:22:57, localpref 100
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0
                    [BGP/170] 5d 16:22:49, localpref 100, from 10.7.7.71
                      AS path: 65300 65200 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0, Push 7137, Push 16(top)
                       to 192.168.12.1192.168.2.2 via ge-0/0/2.0, Push 7167, Push 17(top)
NODE { AS:65200 IPv4:10.3.3.3 STATIC:0 }/1216
                   *[BGP/170] 5d 16:22:57, localpref 100
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0
                    [BGP/170] 5d 16:22:49, localpref 100, from 10.7.7.71
                      AS path: 65300 65200 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0, Push 7137, Push 16(top)
                       to 192.168.12.1 via ge-0/0/2.0, Push 7167, Push 17(top)
NODE { AS:65200 IPv10:4.4.4.4 STATIC:0 }/1216
                   *[BGP/170] 5d 17:35:34, localpref 100
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.12.1192.168.2.2 via ge-0/0/2.0
                    [BGP/170] 5d 16:26:54, localpref 100, from 10.7.7.71
                      AS path: 65300 65200 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0, Push 7137, Push 16(top)
                       to 192.168.12.1192.168.2.2 via ge-0/0/2.0, Push 7167, Push 17(top)
NODE { AS:65200 IPv4:10.6.6.6 STATIC:0 }/1216
                   *[BGP/170] 5d 17:35:34, localpref 100
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.12.1192.168.2.2 via ge-0/0/2.0
```

```
                     [BGP/170] 5d 16:26:54, localpref 100, from 7.7.7.71
                       AS path: 65300 65200 I, validation-state: unverified
                     >  to 192.168.1.2 via ge-0/0/0.0, Push 7137, Push 16(top)
                        to 192.168.12.1 via ge-0/0/2.0, Push 7167, Push 17(top)
NODE { AS:65100 IPv4:10.1.1.1 BGP-LS-EPE:0 }/1216
                    *[BGP-LS-EPE/170] 5d 16:23:39
                        Fictitious
NODE { AS:65100 IPv4:10.4.4.4 BGP-LS-EPE:0 }/1216
                    *[BGP-LS-EPE/170] 5d 17:39:46
                        Fictitious
NODE { AS:65100 IPv4:10.100.100.100 BGP-LS-EPE:0 }/1216
                    *[BGP-LS-EPE/170] 5d 18:02:07
                        Fictitious
NODE { AS:65200 IPv4:10.1.1.1 BGP-LS-EPE:0 }/1216
                    *[BGP/170] 5d 16:23:04, localpref 100
                       AS path: 65200 I, validation-state: unverified
                     >  to 192.168.12.1192.168.2.2 via ge-0/0/2.0
                     [BGP/170] 5d 16:23:04, localpref 100, from 10.7.7.71
                       AS path: 65300 65200 I, validation-state: unverified
                     >  to 192.168.1.2 via ge-0/0/0.0, Push 7137, Push 16(top)
                        to 192.168.12.1192.168.2.2 via ge-0/0/2.0, Push 7167, Push 17(top)
NODE { AS:65200 IPv4:10.3.3.3 BGP-LS-EPE:0 }/1216
                    *[BGP/170] 5d 16:26:56, localpref 100
                       AS path: 65200 I, validation-state: unverified
                     >  to 192.168.12.1192.168.2.2 via ge-0/0/2.0
                     [BGP/170] 5d 16:23:04, localpref 100
                       AS path: 65200 I, validation-state: unverified
                     >  to 192.168.1.2 via ge-0/0/0.0
                     [BGP/170] 5d 16:26:54, localpref 100, from 10.7.7.71
                       AS path: 65300 65200 I, validation-state: unverified
                     >  to 192.168.1.2 via ge-0/0/0.0, Push 7137, Push 16(top)
                        to 192.168.12.1192.168.2.2 via ge-0/0/2.0, Push 7167, Push 17(top)
NODE { AS:65200 IPv4:10.4.4.4 BGP-LS-EPE:0 }/1216
                    *[BGP/170] 5d 16:23:04, localpref 100
                       AS path: 65200 I, validation-state: unverified
                     >  to 192.168.1.2 via ge-0/0/0.0
                     [BGP/170] 5d 16:26:54, localpref 100, from 10.7.7.71
                       AS path: 65300 65200 I, validation-state: unverified
                     >  to 192.168.1.2 via ge-0/0/0.0, Push 7137, Push 16(top)
                        to 192.168.12.1192.168.2.2 via ge-0/0/2.0, Push 7167, Push 17(top)
NODE { AS:65200 IPv4:10.6.6.6 BGP-LS-EPE:0 }/1216
                    *[BGP/170] 5d 16:26:58, localpref 100
                       AS path: 65200 I, validation-state: unverified
```

```
                    >  to 192.168.12.1 via ge-0/0/2.0
                    [BGP/170] 5d 16:23:04, localpref 100
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0
                    [BGP/170] 5d 16:26:54, localpref 100, from 10.7.7.71
                      AS path: 65300 65200 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0, Push 7137, Push 16(top)
                       to 192.168.12.1 via ge-0/0/2.0, Push 7167, Push 17(top)
NODE { AS:65200 IPv4:10.7.7.7 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 5d 16:26:58, localpref 100
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.12.1192.168.2.2 via ge-0/0/2.0
                    [BGP/170] 5d 16:23:04, localpref 100
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0
                    [BGP/170] 5d 16:26:54, localpref 100, from 10.7.7.71
                      AS path: 65300 65200 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0, Push 7137, Push 16(top)
                       to 192.168.12.1192.168.2.2 via ge-0/0/2.0, Push 7167, Push 17(top)
NODE { AS:65200 IPv4:10.100.100.100 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 5d 16:23:03, localpref 100
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.12.1 via ge-0/0/2.0
                    [BGP/170] 5d 16:23:02, localpref 100, from 10.7.7.71
                      AS path: 65300 65200 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0, Push 7137, Push 16(top)
                       to 192.168.12.1192.168.2.2 via ge-0/0/2.0, Push 7167, Push 17(top)
NODE { AS:65300 IPv4:10.3.3.3 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 5d 16:26:54, localpref 100, from 10.7.7.71
                      AS path: 65300 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0, Push 7137, Push 16(top)
                       to 192.168.12.1via ge-0/0/2.0, Push 7167, Push 17(top)
                    [BGP/170] 5d 16:23:04, localpref 100
                      AS path: 65200 65300 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0
                    [BGP/170] 5d 16:26:56, localpref 100
                      AS path: 65200 65300 I, validation-state: unverified
                    >  to 192.168.12.1192.168.2.2 via ge-0/0/2.0
NODE { AS:65300 IPv4:10.6.6.6 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 5d 16:26:54, localpref 100, from 10.7.7.71
                      AS path: 65300 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0, Push 7137, Push 16(top)
                       to 192.168.12.1192.168.2.2 via ge-0/0/2.0, Push 7167, Push 17(top)
```

```
                    [BGP/170] 5d 16:23:04, localpref 100
                      AS path: 65200 65300 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0
                    [BGP/170] 5d 16:26:58, localpref 100
                      AS path: 65200 65300 I, validation-state: unverified
                    >  to 192.168.12.1192.168.2.2 via ge-0/0/2.0
NODE { AS:65300 IPv4:10.7.7.7 BGP-LS-EPE:0 }/1216
                   *[BGP/170] 5d 16:26:54, localpref 100, from 10.7.7.71
                      AS path: 65300 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0, Push 7137, Push 16(top)
                       to 192.168.12.1192.168.2.2 via ge-0/0/2.0, Push 7167, Push 17(top)
                    [BGP/170] 5d 16:23:04, localpref 100
                      AS path: 65200 65300 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0
                    [BGP/170] 5d 16:26:58, localpref 100
                      AS path: 65200 65300 I, validation-state: unverified
                    >  to 192.168.12.1192.168.2.2 via ge-0/0/2.0
LINK { Local { AS:65200 IPv4:10.1.1.1 }.{ IfIndex:2147483649 } Remote { AS:200
IPv4
                        :3.3.3.3 }.{ IfIndex:0 } STATIC:0 }/1216
                   *[BGP/170] 5d 16:22:57, localpref 100
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0
LINK { Local { AS:65200 IPv4:10.3.3.3 }.{ IfIndex:2147483649 } Remote { AS:200
IPv4
                        :10.1.1.1 }.{ IfIndex:0 } STATIC:0 }/1216
                   *[BGP/170] 5d 16:22:49, localpref 100, from 10.7.7.71
                      AS path: 65300 65200 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0, Push 7137, Push 16(top)
                       to 192.168.12.1 via ge-0/0/2.0, Push 7167, Push 17(top)
LINK { Local { AS:65200 IPv4:10.4.4.4 }.{ IfIndex:2147483649 } Remote { AS:65200
IPv4
                        :10.6.6.6 }.{ IfIndex:0 } STATIC:0 }/1216
                   *[BGP/170] 5d 16:27:54, localpref 100
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.12.1 via ge-0/0/2.0
LINK { Local { AS:65200 IPv4:10.6.6.6 }.{ IfIndex:2147483649 } Remote { AS:200
IPv4
                        :10.4.4.4 }.{ IfIndex:0 } STATIC:0 }/1216
                   *[BGP/170] 5d 16:26:54, localpref 100, from 10.7.7.71
                      AS path: 65300 65200 I, validation-state: unverified
                    >  to 192.168.1.2 via ge-0/0/0.0, Push 7137, Push 16(top)
                       to 192.168.12.1 via ge-0/0/2.0, Push 7167, Push 17(top)
```

```
LINK { Local { AS:65100 IPv4:10.100.100.100 }.{ IfIndex:333 } Remote { AS:65200
IPv
                    4:10.1.1.1 }.{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                *[BGP-LS-EPE/170] 5d 16:23:39
                    Fictitious
LINK { Local { AS:65100 IPv4:10.100.100.100 }.{ IfIndex:336 } Remote { AS:65200
IPv
                    4:10.4.4.4 }.{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                *[BGP-LS-EPE/170] 5d 17:39:46
                    Fictitious
LINK { Local { AS:65200 IPv4:10.1.1.1 }.{ IfIndex:333 } Remote { AS:65100
IPv4:10.100
                             0.10.100 }.{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                *[BGP/170] 5d 16:23:04, localpref 100
                  AS path: 65200 I, validation-state: unverified
                 >  to 192.168.12.1 via ge-0/0/2.0
                 [BGP/170] 5d 16:23:04, localpref 100, from 10.7.7.71
                  AS path: 65300 65200 I, validation-state: unverified
                 >  to 192.168.1.2 via ge-0/0/0.0, Push 7137, Push 16(top)
                    to 192.168.12.1 via ge-0/0/2.0, Push 7167, Push 17(top)
LINK { Local { AS:65200 IPv4:10.3.3.3 }.{ IfIndex:342 } Remote { AS:65300
IPv4:10.7.7.
                              7 }.{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                *[BGP/170] 5d 16:26:56, localpref 100
                  AS path: 65200 I, validation-state: unverified
                 >  to 192.168.12.1192.168.2.2 via ge-0/0/2.0
                 [BGP/170] 5d 16:23:04, localpref 100
                  AS path: 65200 I, validation-state: unverified
                 >  to 192.168.1.2 via ge-0/0/0.0
                 [BGP/170] 5d 16:26:54, localpref 100, from 10.7.7.71
                  AS path: 65300 65200 I, validation-state: unverified
                 >  to 192.168.1.2 via ge-0/0/0.0, Push 7137, Push 16(top)
                    to 192.168.12.1via ge-0/0/2.0, Push 7167, Push 17(top)
LINK { Local { AS:65200 IPv10:10.4.4.4 }.{ IfIndex:333 } Remote { AS:65100
IPv4:10.100
                             0.100.100 }.{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                *[BGP/170] 5d 16:23:04, localpref 100
                  AS path: 65200 I, validation-state: unverified
                 >  to 192.168.1.2 via ge-0/0/0.0
                 [BGP/170] 5d 16:26:54, localpref 100, from 10.7.7.71
                  AS path: 65300 65200 I, validation-state: unverified
                 >  to 192.168.1.2 via ge-0/0/0.0, Push 7137, Push 16(top)
                    to 192.168.12.1 via ge-0/0/2.0, Push 7167, Push 17(top)
```

```
LINK { Local { AS:65200 IPv4:10.6.6.6 }.{ IfIndex:341 } Remote { AS:65300
IPv4:10.7.7.
                              7 }.{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                    *[BGP/170] 5d 16:26:58, localpref 100
                       AS path: 65200 I, validation-state: unverified
                     >  to 192.168.12.1 via ge-0/0/2.0
                     [BGP/170] 5d 16:23:04, localpref 100
                       AS path: 65200 I, validation-state: unverified
                     >  to 192.168.1.2 via ge-0/0/0.0
                     [BGP/170] 5d 16:26:54, localpref 100, from 10.7.7.71
                       AS path: 65300 65200 I, validation-state: unverified
                     >  to 192.168.1.2 via ge-0/0/0.0, Push 7137, Push 16(top)
                        to 192.168.12.1192.168.2.2 via ge-0/0/2.0, Push 7167, Push 17(top)
LINK { Local { AS:65300 IPv4:10.7.7.7 }.{ IfIndex:333 } Remote { AS:65200
IPv4:10.3.3.
                              3 }.{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                    *[BGP/170] 5d 16:26:54, localpref 100, from 10.7.7.71
                       AS path: 65300 I, validation-state: unverified
                     >  to 192.168.1.2 via ge-0/0/0.0, Push 7137, Push 16(top)
                        to 192.168.12.1 via ge-0/0/2.0, Push 7167, Push 17(top)
                     [BGP/170] 5d 16:23:04, localpref 100
                       AS path: 65200 65300 I, validation-state: unverified
                     >  to 192.168.1.2 via ge-0/0/0.0
                     [BGP/170] 5d 16:26:56, localpref 100
                       AS path: 65200 65300 I, validation-state: unverified
                     >  to 192.168.12.1 via ge-0/0/2.0
LINK { Local { AS:65300 IPv4:10.7.7.7 }.{ IfIndex:336 } Remote { AS:65200
IPv4:10.6.6.
                              6 }.{ IfIndex:0 } BGP-LS-EPE:0 }/1216
                    *[BGP/170] 5d 16:26:54, localpref 100, from 10.7.7.71
                       AS path: 65300 I, validation-state: unverified
                     >  to 192.168.1.2 via ge-0/0/0.0, Push 7137, Push 16(top)
                        to 192.168.12.1192.168.2.2 via ge-0/0/2.0, Push 7167, Push 17(top)
                     [BGP/170] 5d 16:23:04, localpref 100
                       AS path: 65200 65300 I, validation-state: unverified
                     >  to 192.168.1.2 via ge-0/0/0.0
                     [BGP/170] 5d 16:26:58, localpref 100
                       AS path: 65200 65300 I, validation-state: unverified
                     >  to 192.168.12.1 via ge-0/0/2.0
PREFIX { Node { AS:65100 ISO:0100.0a0a.0a0a.00 } { IPv4:10.100.100.100/32 }
ISIS-
                         L2:0 }/1216
                    *[IS-IS/18] 5d 18:02:43
```

```
                       Fictitious
PREFIX { Node { AS:65100 ISO:0100.0a0a.0a0a.00 } { IPv4:10.100.100.101/32 }
ISIS-
                         L2:0 }/1216
                    *[IS-IS/18] 5d 18:02:43
                       Fictitious
P
```

On R0

```
user@R0>show ted database topology-type l3-unicast detail

TED database: 1 ISIS nodes 6 INET nodes 0 INET6 nodes
NodeID: R0.00(10.100.100.100)
  Type: Rtr, Age: 356 secs, LinkIn: 2, LinkOut: 2
  Protocol: Exported BGP(4)
  Protocol: BGP-LS-EPE(0)
    To: 10.4.4.4, Local: 192.168.2.1, Remote: 192.168.1.2
      Local interface index: 336, Remote interface index: 0
      Link name: epe_adj1_toR4
      Local bgp peer as: 65100, Remote bgp peer as: 65200
    To: 10.1.1.1, Local: 192.168.1.1, Remote: 192.168.1.2
      Local interface index: 333, Remote interface index: 0
      Link name: epe_adj1_toR1
      Local bgp peer as: 65100, Remote bgp peer as: 65200
  Protocol: IS-IS(2)
 10.100.100.100, 10.100.100.101
NodeID: 10.1.1.1
  Type: Rtr, Age: 491222 secs, LinkIn: 2, LinkOut: 2
  Protocol: Exported BGP(4)
    To: R0.00(10.100.100.100), Local: 0.0.0.0, Remote: 0.0.0.0
      Local interface index: 333, Remote interface index: 0
      Link name: epe_adj1_toR0
  Protocol: Exported STATIC(2)
    To: 10.3.3.3, Local: 0.0.0.0, Remote: 0.0.0.0
      Local interface index: 2147483649, Remote interface index: 0
      Link name: set1sr-10.3.3.3
  Protocol: BGP-LS-EPE(0)
NodeID: 10.3.3.3
  Type: Rtr, Age: 491420 secs, LinkIn: 2, LinkOut: 2
  Protocol: Exported BGP(4)
    To: 10.7.7.7, Local: 0.0.0.0, Remote: 0.0.0.0
```

```
      Local interface index: 342, Remote interface index: 0
      Link name: epe_adj1_toR7
  Protocol: Exported BGP(6)
  Protocol: Exported STATIC(2)
    To: 10.1.1.1, Local: 0.0.0.0, Remote: 0.0.0.0
      Local interface index: 2147483649, Remote interface index: 0
      Link name: set3sr-10.1.1.1
NodeID: 10.4.4.4
  Type: Rtr, Age: 495789 secs, LinkIn: 2, LinkOut: 2
  Protocol: Exported BGP(4)
    To: R0.00(10.100.100.100), Local: 0.0.0.0, Remote: 0.0.0.0
      Local interface index: 333, Remote interface index: 0
      Link name: epe_adj1_toR0
  Protocol: Exported STATIC(2)
    To: 10.6.6.6, Local: 0.0.0.0, Remote: 0.0.0.0
      Local interface index: 2147483649, Remote interface index: 0
      Link name: set4sr-10.6.6.6
  Protocol: BGP-LS-EPE(0)
NodeID: 10.6.6.6
  Type: Rtr, Age: 495537 secs, LinkIn: 2, LinkOut: 2
  Protocol: Exported BGP(4)
    To: 10.7.7.7, Local: 0.0.0.0, Remote: 0.0.0.0
      Local interface index: 341, Remote interface index: 0
      Link name: epe_adj1_toR7
  Protocol: Exported BGP(6)
  Protocol: Exported STATIC(2)
    To: 10.4.4.4, Local: 0.0.0.0, Remote: 0.0.0.0
      Local interface index: 2147483649, Remote interface index: 0
      Link name: set6sr-10.4.4.4
NodeID: 10.7.7.7
  Type: Rtr, Age: 491421 secs, LinkIn: 2, LinkOut: 2
  Protocol: Exported BGP(4)
  Protocol: Exported BGP(6)
    To: 10.6.6.6, Local: 0.0.0.0, Remote: 0.0.0.0
      Local interface index: 336, Remote interface index: 0
      Link name: epe_adj1_toR6
    To: 10.3.3.3, Local: 0.0.0.0, Remote: 0.0.0.0
      Local interface index: 333, Remote interface index: 0
      Link name: epe_adj1_toR3
```

On R0

```
user@R0>show spring-traffic-engineering lsp

To              State     LSPname
10.7.7.7         Up        computelsp1
10.7.7.7-7000<c> Up        ecomputelsp1


Total displayed LSPs: 2 (Up: 2, Down: 0)
```

On R0

```
user@R0>show spring-traffic-engineering lsp detail

Name: computelsp1
  Tunnel-source: Static configuration
  To: 10.7.7.7
  State: Up
    Path: p1
    Path Status: NA
    Outgoing interface: NA
    Auto-translate status: Disabled Auto-translate result: N/A
    Compute Status:Enabled , Compute Result:success , Compute-Profile Name:compute1
    Total number of computed paths: 2
    Computed-path-index: 1
      BFD status: N/A BFD name: N/A
      TE metric: 60, IGP metric: 40; Metric optimized by type: TE
      computed segments count: 3
        computed segment : 1 (computed-adjacency-segment):
          label: 7101
          source router-id: 10.100.100.100, destination router-id: 10.1.1.1
          source interface-address: 192.168.1.1, destination interface-address: 192.168.1.2
        computed segment : 2 (computed-adjacency-segment):
          label: 16
          source router-id: 10.1.1.1, destination router-id: 10.3.3.3
          source interface-address: 0.0.0.0, destination interface-address: 0.0.0.0
        computed segment : 3 (computed-adjacency-segment):
          label: 7137
          source router-id: 10.3.3.3, destination router-id: 10.7.7.7
          source interface-address: 0.0.0.0, destination interface-address: 0.0.0.0
```

```
    Computed-path-index: 2
      BFD status: N/A BFD name: N/A
      TE metric: 60, IGP metric: 40; Metric optimized by type: TE
      computed segments count: 3
        computed segment : 1 (computed-adjacency-segment):
          label: 7104
          source router-id: 10.100.100.100, destination router-id: 10.4.4.4
          source interface-address: 192.168.2.1, destination interface-address: 192.168.12.1
        computed segment : 2 (computed-adjacency-segment):
          label: 17
          source router-id: 10.4.4.4, destination router-id: 10.6.6.6
          source interface-address: 0.0.0.0, destination interface-address: 0.0.0.0
        computed segment : 3 (computed-adjacency-segment):
          label: 7167
          source router-id: 10.6.6.6, destination router-id: 10.7.7.7
          source interface-address: 0.0.0.0, destination interface-address: 0.0.0.0

Name: ecomputelsp1
  Tunnel-source: Static configuration
  To: 10.7.7.7-7000<c>
  State: Up
    Path: p1
    Path Status: NA
    Outgoing interface: NA
    Auto-translate status: Disabled Auto-translate result: N/A
    Compute Status:Enabled , Compute Result:success , Compute-Profile Name:ecompute1
    Total number of computed paths: 2
    Computed-path-index: 1
      BFD status: N/A BFD name: N/A
      TE metric: 60, IGP metric: 40; Metric optimized by type: TE
      computed segments count: 3
        computed segment : 1 (computed-adjacency-segment):
          label: 7101
          source router-id: 10.100.100.100, destination router-id: 10.1.1.1
          source interface-address: 192.168.1.1, destination interface-address: 192.168.1.2
        computed segment : 2 (computed-adjacency-segment):
          label: 16
          source router-id: 10.1.1.1, destination router-id: 10.3.3.3
          source interface-address: 0.0.0.0, destination interface-address: 0.0.0.0
        computed segment : 3 (computed-adjacency-segment):
          label: 7137
          source router-id: 10.3.3.3, destination router-id: 10.7.7.7
          source interface-address: 0.0.0.0, destination interface-address: 0.0.0.0
```

```
    Computed-path-index: 2
      BFD status: N/A BFD name: N/A
      TE metric: 60, IGP metric: 40; Metric optimized by type: TE
      computed segments count: 3
        computed segment : 1 (computed-adjacency-segment):
          label: 7104
          source router-id: 10.100.100.100, destination router-id: 10.4.4.4
          source interface-address: 192.168.2.1, destination interface-address: 192.168.12.1
        computed segment : 2 (computed-adjacency-segment):
          label: 17
          source router-id: 10.4.4.4, destination router-id: 10.6.6.6
          source interface-address: 0.0.0.0, destination interface-address: 0.0.0.0
        computed segment : 3 (computed-adjacency-segment):
          label: 7167
          source router-id: 10.6.6.6, destination router-id: 10.7.7.7
          source interface-address: 0.0.0.0, destination interface-address: 0.0.0.0
Total displayed LSPs: 2 (Up: 2, Down: 0)
```

Meaning

- In the `show route receive-protocol bgp 192.168.1.2` output, it shows the routes that have been received by the ingress router (R0) from the BGP neighbor, which describes the express segment (virtual TE links).
- In the `show route table lsdist.0` output, it shows the routes that have been received by the ingress router (R0) and whether they are inserted into the **lsdist.0** RIB. It also shows whether the **lsdist.0** RIB is copied into the local TE database.
- In the `show ted database topology-type l3-unicast detail` output, the routes are copied into the local TE database. The **set1sr-10.3.3.3** is an express segment with end point as 3.3.3.3 and is successfully created on R1. R1 has advertised the express segment and R0 has inserted it into the local TE database. You can also see the EPE segments (**epe_adj1_toR7**).
- In the `show spring-traffic-engineering lsp` output, you can see that the SR policies are up. It shows that you are now able to compute a multi-domain end-to-end (R0 to R7) SR policy.
- In the `show spring-traffic-engineering lsp detail` output, you can see the labels that are selected. In the **computelsp1** LSP, the label **7101** is an EPE segment, **16** is the express segment, and **7137** is also an EPE segment. It shows that you are now able to compute a multi-domain end-to-end (R0 to R7) SR policy.

5

PART

MPLS Signalling Protocols

- RSVP | **1224**
- LDP | **1316**

CHAPTER 10

RSVP

IN THIS CHAPTER

- RSVP Overview | 1224
- RSVP Configuration | 1240

RSVP Overview

IN THIS SECTION

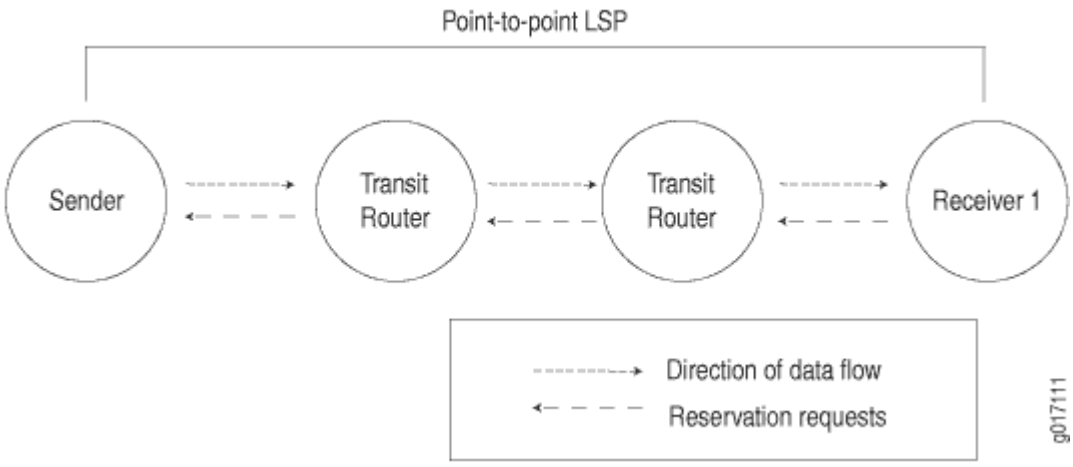
- RSVP Overview | 1225
- RSVP Operation Overview | 1225
- Understanding the RSVP Signaling Protocol | 1226
- RSVP-TE protocol extensions for FRR | 1229
- Junos OS RSVP Protocol Implementation | 1231
- RSVP Authentication | 1232
- RSVP and IGP Hello Packets and Timers | 1232
- RSVP Message Types | 1233
- Understanding RSVP Automatic Mesh | 1235
- RSVP Reservation Styles | 1236
- RSVP Refresh Reduction | 1237
- MTU Signaling in RSVP | 1237
- How the Correct MTU Is Signaled in RSVP | 1238
- Determining an Outgoing MTU Value | 1239
- MTU Signaling in RSVP Limitations | 1239

RSVP Overview

The RSVP protocol is used by routers to deliver quality-of-service (QoS) requests to all nodes along data flow path(s) and to establish and maintain state for the requested service. RSVP requests generally result in resource reservations in each node along the data path. RSVP has the following attributes:

- Makes resource reservations for unidirectional data flows.
- Allows the receiver of a data flow to initiate and maintain the resource reservation used for that flow, as shown in Figure 68 on page 1225.
- Maintains a soft state in routers and hosts, providing graceful support for dynamic membership changes and automatic adaptation to routing changes.
- Depends upon present and future routing protocols, but is not a routing protocol itself.
- Provides several reservation models or styles to fit a variety of applications.
- Supports both IPv4 and IPv6 packets that can be sent over RSVP-signaled LSPs.

Figure 68: RSVP Reservation Request and Data Flow



RSVP Operation Overview

RSVP creates independent sessions to handle each data flow. A session is identified by a combination of the destination address, an optional destination port, and a protocol. Within a session, there can be one or more senders. Each sender is identified by a combination of its source address and source port. An out-of-band mechanism, such as a session announcement protocol or human communication, is used to communicate the session identifier to all senders and receivers.

A typical RSVP session involves the following sequence of events:

1. A potential sender starts sending RSVP path messages to the session address.
2. A receiver, wanting to join the session, registers itself if necessary. For example, a receiver in a multicast application would register itself with IGMP.
3. The receiver receives the path messages.
4. The receiver sends appropriate Resv messages toward the sender. These messages carry a flow descriptor, which is used by routers along the path to make reservations in their link-layer media.
5. The sender receives the Resv message and then starts sending application data.

This sequence of events is not necessarily strictly synchronized. For example, receivers can register themselves before receiving path messages from the sender, and application data can flow before the sender receives Resv messages. Application data that is delivered before the actual reservation contained in the Resv message typically is treated as best-effort, non-real-time traffic with no CoS guarantee.

Understanding the RSVP Signaling Protocol

IN THIS SECTION

- RSVP Fundamentals | **1227**
- Bandwidth Reservation Requirement | **1227**
- Explicit Route Objects | **1227**
- Constrained Shortest Path First | **1228**
- Link Coloring | **1229**

RSVP is a signaling protocol that handles bandwidth allocation and true traffic engineering across an MPLS network. Like LDP, RSVP uses discovery messages and advertisements to exchange LSP path information between all hosts. However, RSVP also includes a set of features that control the flow of traffic through an MPLS network. Whereas LDP is restricted to using the configured IGP's shortest path as the transit path through the network, RSVP uses a combination of the Constrained Shortest Path First (CSPF) algorithm and Explicit Route Objects (EROs) to determine how traffic is routed through the network.

Basic RSVP sessions are established in exactly the same way that LDP sessions are established. By configuring both MPLS and RSVP on the appropriate transit interfaces, you enable the exchange of RSVP packets and the establishment of LSPs. However, RSVP also lets you configure link authentication, explicit LSP paths, and link coloring.

This topic contains the following sections:

RSVP Fundamentals

RSVP uses unidirectional and simplex flows through the network to perform its function. The inbound router initiates an RSVP path message and sends it downstream to the outbound router. The path message contains information about the resources needed for the connection. Each router along the path begins to maintain information about a reservation.

When the path message reaches the outbound router, resource reservation begins. The outbound router sends a reservation message upstream to the inbound router. Each router along the path receives the reservation message and sends it upstream, following the path of the original path message. When the inbound router receives the reservation message, the unidirectional network path is established.

The established path remains open as long as the RSVP session is active. The session is maintained by the transmission of additional path and reservation messages that report the session state every 30 seconds. If a router does not receive the maintenance messages for 3 minutes, it terminates the RSVP session and reroutes the LSP through another active router.

Bandwidth Reservation Requirement

When a bandwidth reservation is configured, reservation messages propagate the bandwidth value throughout the LSP. Routers must reserve the bandwidth specified across the link for the particular LSP. If the total bandwidth reservation exceeds the available bandwidth for a particular LSP segment, the LSP is rerouted through another LSR. If no segments can support the bandwidth reservation, LSP setup fails and the RSVP session is not established.

Explicit Route Objects

Explicit Route Objects (EROs) limit LSP routing to a specified list of LSRs. By default, RSVP messages follow a path that is determined by the network IGP's shortest path. However, in the presence of a configured ERO, the RSVP messages follow the path specified.

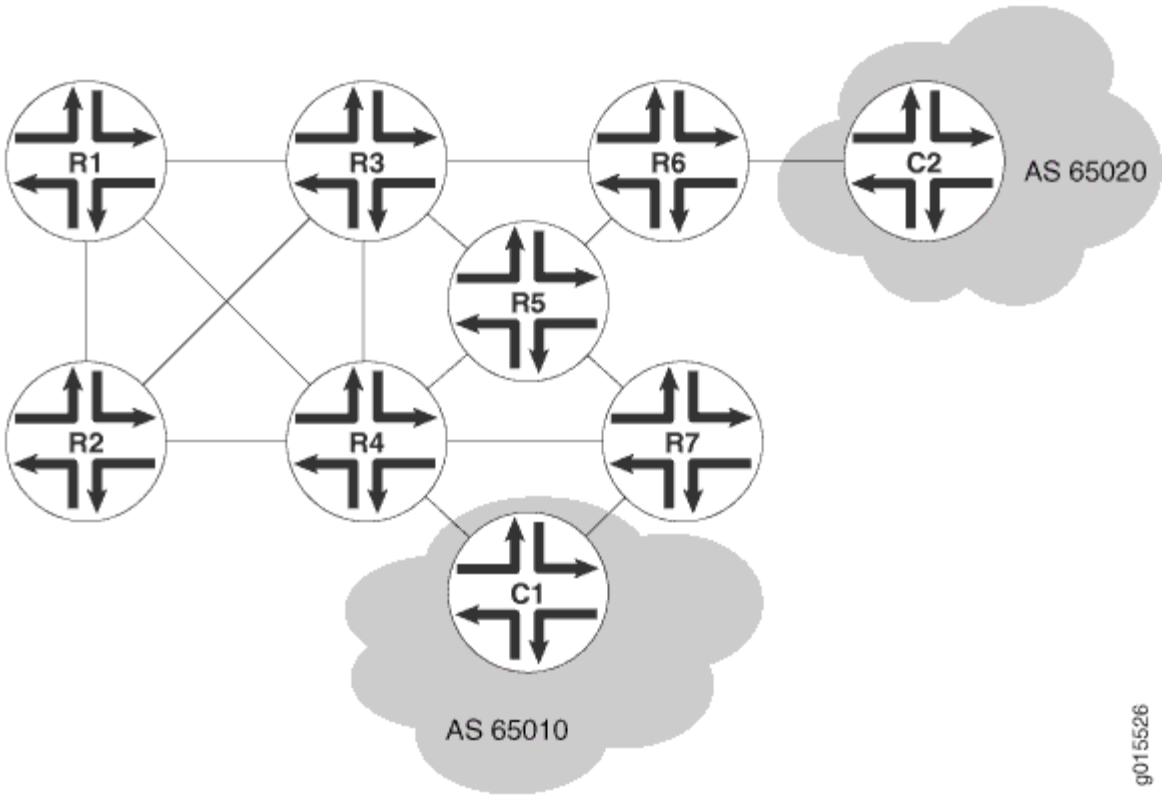
EROs consist of two types of instructions: loose hops and strict hops. When a loose hop is configured, it identifies one or more transit LSRs through which the LSP must be routed. The network IGP determines the exact route from the inbound router to the first loose hop, or from one loose hop to the next. The loose hop specifies only that a particular LSR be included in the LSP.

When a strict hop is configured, it identifies an exact path through which the LSP must be routed. Strict-hop EROs specify the exact order of the routers through which the RSVP messages are sent.

You can configure loose-hop and strict-hop EROs simultaneously. In this case, the IGP determines the route between loose hops, and the strict-hop configuration specifies the exact path for particular LSP path segments.

Figure 69 on page 1228 shows a typical RSVP-signaled LSP that uses EROs.

Figure 69: Typical RSVP-Signaled LSP with EROs



In the topology shown in Figure 69 on page 1228, traffic is routed from Host C1 to Host C2. The LSP can pass through Routers R4 or Router R7. To force the LSP to use R4, you can set up either a loose-hop or strict-hop ERO that specifies R4 as a hop in the LSP. To force a specific path through Router R4, R3, and R6, configure a strict-hop ERO through the three LSRs.

Constrained Shortest Path First

Whereas IGPs use the Shortest Path First (SPF) algorithm to determine how traffic is routed within a network, RSVP uses the Constrained Shortest Path First (CSPF) algorithm to calculate traffic paths that are subject to the following constraints:

- LSP attributes—Administrative groups such as link coloring, bandwidth requirements, and EROs
- Link attributes—Colors on a particular link and available bandwidth

These constraints are maintained in the traffic engineering database (TED). The database provides CSPF with up-to-date topology information, the current reservable bandwidth of links, and the link colors.

In determining which path to select, CSPF follows these rules:

- Computes LSPs one at a time, beginning with the highest priority LSP—the one with the lowest setup priority value. Among LSPs of equal priority, CSPF starts with those that have the highest bandwidth requirement.
- Prunes the traffic engineering database of links that are not full duplex and do not have sufficient reservable bandwidth.
- If the LSP configuration includes the `include` statement, prunes all links that do not share any included colors.
- If the LSP configuration includes the `exclude` statement, prunes all links that contain excluded colors. If a link does not have a color, it is accepted.
- Finds the shortest path toward the LSP's outbound router, taking into account any EROs. For example, if the path must pass through Router A, two separate SPF algorithms are computed: one from the inbound router to Router A and one from Router A to the outbound router.
- If several paths have equal cost, chooses the one with a last-hop address the same as the LSP's destination.
- If several equal-cost paths remain, selects the path with the least number of hops.
- If several equal-cost paths remain, applies CSPF load-balancing rules configured on the LSP.

Link Coloring

RSVP allows you to configure administrative groups for CSPF path selection. An administrative group is typically named with a color, assigned a numeric value, and applied to the RSVP interface for the appropriate link. Lower numbers indicate higher priority.

After configuring the administrative group, you can either exclude, include, or ignore links of that color in the TED:

- If you exclude a particular color, all segments with an administrative group of that color are excluded from CSPF path selection.
- If you include a particular color, only those segments with the appropriate color are selected.
- If you neither exclude nor include the color, the metrics associated with the administrative groups and applied on the particular segments determine the path cost for that segment.

The LSP with the lowest total path cost is selected into the TED.

RSVP-TE protocol extensions for FRR

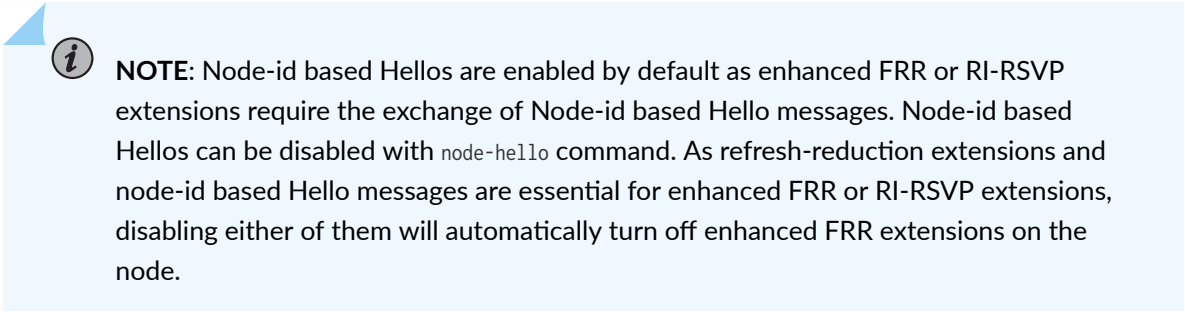
Starting with Junos OS Release 16.1, RSVP Traffic Engineering (TE) protocol extensions to support Refresh-interval Independent RSVP (RI-RSVP) defined RFC 8370 for fast reroute (FRR) facility

protection were introduced to allow greater scalability of label-switched paths (LSPs) faster convergence times and decrease RSVP signaling message overhead from periodic refreshes. Junos RSVP-TE runs in enhanced FRR aka RI-RSVP mode by default that includes protocol extensions to support RI-RSVP for FRR facility bypass originally specified in RFC 4090.

The RI-RSVP protocol extensions implemented in Junos are fully backward compatible. In mixed environments, where a subset of LSPs traverse nodes that do not include this feature, Junos RSVP-TE running in enhanced FRR mode will automatically turn off the new protocol extensions in its signaling exchanges with nodes that do not support the new extensions.

As part of enhanced FRR profile, a number of changes were made and new defaults adopted. These are listed here.

- RSVP-TE runs “enhanced” FRR, or RI-RSVP mode, by default, which includes enhancements to facilitate scaled up scenarios. These new protocol extensions can be disabled on a router with the *no-enhanced-frr-bypass* command.
- RSVP refresh reduction extensions defined in RFC 2961 are enabled by default. The user can disable them with the no-reliable command (see below).

  NOTE: Node-id based Hellos are enabled by default as enhanced FRR or RI-RSVP extensions require the exchange of Node-id based Hello messages. Node-id based Hellos can be disabled with `node-hello` command. As refresh-reduction extensions and node-id based Hello messages are essential for enhanced FRR or RI-RSVP extensions, disabling either of them will automatically turn off enhanced FRR extensions on the node.

- The default refresh time for RSVP messages has increased from 30 seconds to 20 minutes.
- The default value for *keep-multiplier*, which is 3, is applied to the new default refresh time.
- Local reversion is disabled by default. The existing CLI configuration for node Hellos, `[edit protocols rsvp node-hello]`, is still available but the action is redundant. If enabled, a message is displayed to indicate that the configuration is not necessary in conjunction with enhanced FRR.
- The existing commands to disable message reliability are now used to disable RSVP refresh reduction. To revert back to the previous default enabling refresh reduction, use the `delete` version of the following commands:
  - Set `no-reliable` on a given interface to disable FRR scalability enhancements automatically for all LSPs traversing the interface. Likewise, to run RSVP-TE without FRR facility protection enhancements, and without refresh-reduction, disable refresh reduction on each interface. Use one of the commands shown here:
    - `[edit protocols rsvp interface` *`name`* `no-reliable]`

    - [edit logical-systems *name* protocols rsvp interface *name* no-reliable]

- Graceful restart and nonstop active routing (NSR) are not supported while the LSP undergoes local repair or while the LSP is refreshed during backup LSP signaling. This limitation exists already in the implementation because GR or NSR switchover during FRR local-repair makes for multiple failure scenario.
- The following operational commands have been updated to include new information about the new procedures introduced for the RSVP TE protocol extensions for FRR facility protection.
    - show rsvp version displays whether enhanced FRR procedures are enabled.
    - show rsvp neighbor detail displays whether enhanced FRR procedures are enabled on the neighbor.
    - show rsvp interface detail displays conditional PathTear statistics.
    - show rsvp statistics displays sent and received statistics for conditional PathTear, along with other statistics.
    - show rsvp session extensive displays whether the node is a merge point, and if it is, shows the Point of Local Repair (PLR) address.
- The previous CLI configuration options for enabling or disabling message bundling have been deprecated (the existing configurations are accepted, but a warning is displayed in the show configuration output). These commands are the following:
    - [edit protocols rsvp interface *name* aggregate]
    - [edit logical-systems *name* protocols rsvp interface *name* aggregate]
    - [edit protocols rsvp interface *name* no-aggregate]
    - [edit logical-systems *name* protocols rsvp interface *name* no-aggregate]
- The following CLI configuration options have been made redundant by the current changes (the existing configurations are accepted, but a warning is displayed in the show configuration output):
    - [edit protocols rsvp interface *name* reliable]
    - [edit logical-systems *name* protocols rsvp interface *name* reliable]

Junos OS RSVP Protocol Implementation

The Junos implementation of RSVP supports RSVP version 1. The software includes support for all mandatory objects and RSVP message types, and supports message integrity and node authentications through the Integrity object.

The primary purpose of the Junos RSVP software is to support dynamic signaling within MPLS label-switched paths (LSPs). Supporting resource reservations over the Internet is only a secondary purpose of

the Junos OS implementation. Since supporting resource reservations is secondary, the Junos RSVP software does not support the following features:

- IP multicasting sessions.
- Traffic control. The software cannot make resource reservations for real-time video or audio sessions.

With regard to the protocol mechanism, packet processing, and RSVP objects supported, the Junos OS implementation of the software is interoperable with other RSVP implementations.

RSVP Authentication

The Junos OS supports both the RSVP authentication style described in RFC 2747 (allowing for multivendor compatibility) and the RSVP authentication style described in Internet draft draft-ietf-rsvp-md5-03.txt. The Junos OS uses the authentication style described in Internet draft draft-ietf-rsvp-md5-08.txt by default. If the router receives an RFC 2747-style RSVP authentication from a neighbor, it switches to this style of authentication for that neighbor. The RSVP authentication style for each neighboring router is determined separately.

RSVP and IGP Hello Packets and Timers

RSVP monitors the status of the interior gateway protocol (IGP) (IS-IS or OSPF) neighbors and relies on the IGP protocols to detect when a node fails. If an IGP protocol declares a neighbor down (because hello packets are no longer being received), RSVP also brings down that neighbor. However, the IGP protocols and RSVP still act independently when bringing a neighbor up.

In the Junos OS, RSVP typically relies on IGP hello packet detection to check for node failures. Configuring a short time for the IS-IS or OSPF hello timers allows these protocols to detect node failures quickly. When the node fails or a node failure is detected, a path error message is generated, and although the RSVP session goes down, the IGP neighbors remain up.

RSVP hellos can be relied on when the IGP does not recognize a particular neighbor (for example, if IGP is not enabled on the interface) or if the IGP is RIP (not IS-IS or OSPF). Also, the equipment of other vendors might be configured to monitor RSVP sessions based on RSVP hello packets. This equipment might also take an RSVP session down due to a loss of RSVP hello packets.

We do not recommend configuring a short RSVP hello timer. If quick discovery of a failed neighbor is needed, configure short IGP (OSPF or IS-IS) hello timers.

OSPF and IS-IS have infrastructure to manage rapid hello message sending and receiving reliably, even if the routing protocols or some other process are straining the processing capability of the router. Under the same circumstances, RSVP hellos might time out prematurely even though the neighbor is functioning normally.

RSVP Message Types

IN THIS SECTION

- Path Messages | 1233
- Resv Messages | 1233
- PathTear Messages | 1234
- ResvTear Messages | 1234
- PathErr Messages | 1234
- ResvErr Messages | 1234
- ResvConfirm Messages | 1234

RSVP uses the following types of messages to establish and remove paths for data flows, establish and remove reservation information, confirm the establishment of reservations, and report errors:

Path Messages

Each sender host transmits path messages downstream along the routes provided by the unicast and multicast routing protocols. Path messages follow the exact paths of application data, creating path states in the routers along the way, thus enabling routers to learn the previous-hop and next-hop node for the session. Path messages are sent periodically to refresh path states.

The refresh interval is controlled by a variable called the *refresh-time*, which is the periodical refresh timer expressed in seconds. A path state times out if a router does not receive a specified number of consecutive path messages. This number is specified by a variable called *keep-multiplier*. Path states are kept for ((*keep-multiplier* + 0.5) x 1.5 x *refresh-time*) seconds.

Resv Messages

Each receiver host sends reservation request (Resv) messages upstream toward senders and sender applications. Resv messages must follow exactly the reverse path of path messages. Resv messages create and maintain a reservation state in each router along the way.

Resv messages are sent periodically to refresh reservation states. The refresh interval is controlled by the same refresh time variable, and reservation states are kept for ((*keep-multiplier* + 0.5) x 1.5 x *refresh-time*) seconds.

PathTear Messages

PathTear messages remove (tear down) path states as well as dependent reservation states in any routers along a path. PathTear messages follow the same path as path messages. A PathTear typically is initiated by a sender application or by a router when its path state times out.

PathTear messages are not required, but they enhance network performance because they release network resources quickly. If PathTear messages are lost or not generated, path states eventually time out when they are not refreshed, and the resources associated with the path are released.

ResvTear Messages

ResvTear messages remove reservation states along a path. These messages travel upstream toward senders of the session. In a sense, ResvTear messages are the reverse of Resv messages. ResvTear messages typically are initiated by a receiver application or by a router when its reservation state times out.

ResvTear messages are not required, but they enhance network performance because they release network resources quickly. If ResvTear messages are lost or not generated, reservation states eventually time out when they are not refreshed, and the resources associated with the reservation are released.

PathErr Messages

When path errors occur (usually because of parameter problems in a path message), the router sends a unicast PathErr message to the sender that issued the path message. PathErr messages are advisory; these messages do not alter any path state along the way.

ResvErr Messages

When a reservation request fails, a ResvErr error message is delivered to all the receivers involved. ResvErr messages are advisory; these messages do not alter any reservation state along the way.

ResvConfirm Messages

Receivers can request confirmation of a reservation request, and this confirmation is sent with a ResvConfirm message. Because of the complex RSVP flow-merging rules, a confirmation message does not necessarily provide end-to-end confirmation of the entire path. Therefore, ResvConfirm messages are an indication, not a guarantee, of potential success.

Juniper Networks routers never request confirmation using the ResvConfirm message; however, a Juniper Networks router can send a ResvConfirm message if it receives a request from another vendor's equipment.

Understanding RSVP Automatic Mesh

When adding sites to BGP and MPLS VPNs that use RSVP signaling, more configuration is needed to add provider edge (PE) routers than is needed to add customer edge (CE) devices. RSVP automatic mesh helps to reduce this configuration burden.

Service providers often use BGP and MPLS VPNs to efficiently scale the network while delivering revenue-generating services. In these environments, BGP is used to distribute the VPN routing information across the service provider's network, while MPLS is used to forward that VPN traffic from one VPN site to another. BGP and MPLS VPNs are based on a peer model. To add a new CE device (site) to an existing VPN, you need to configure the CE router at the new site and the PE router connected to the CE router. You do not have to modify the configuration of all of the other PE routers participating in the VPN. The other PE routers automatically learn about the routes associated with the new site, a process called automatic discovery (AD).

The requirements are a bit different if you need to add a new PE router to the network. A BGP and MPLS VPN requires that the BGP session be fully meshed and that there also be a full mesh of PE router-to-PE router MPLS label-switched paths (LSPs) between all of the PE routers in the network. When you add a new PE router to the network, all of the existing PE routers must be reconfigured to peer with the new PE router. Much of the configuration effort can be reduced if you configure BGP route reflectors (mitigating the full mesh requirement for BGP) and if you configure (LDP) as the signaling protocol for MPLS.

However, if you need to add a new PE router to a network configured with a full mesh of RSVP-signaled LSPs, you must reconfigure each of the PE routers to have a peer relationship with the new PE router. You can configure RSVP automatic mesh to address this particular operational scenario. When you enable RSVP automatic mesh, RSVP LSPs are dynamically created between a new PE router and the existing PE routers, eliminating the need to reconfigure all of the PE routers manually. For dynamic LSP creation to function properly, BGP must be configured to exchange routes between all of the participating PE routers. If two BGP peers do not exchange routes, it is not possible to configure a dynamic LSP between them. The local router’s inet.3 routing table must include a labeled route to each potential IBGP next-hop (future potential PE routers or LSP destinations).

RSVP includes numerous capabilities that are not available in LDP, including fast reroute, end-point control, and link management. RSVP automatic mesh helps to reduce the operation and maintenance requirements for RSVP, making it possible to deploy RSVP in larger and more complicated networks.

Every PE router can reach every other PE router in the network because this information is distributed by the IGP. A PE router can set up a point-to-point RSVP LSP to any other PE router in the network as long as it knows that such an LSP is required. To build a full mesh of LSPs between the PE routers requires that each PE router know which of the other PE routers make up the full mesh.

NOTE: In Junos OS, RSVP automatic mesh is configured using the `rsvp-te` configuration statement at the `[edit routing-options dynamic-tunnels dynamic-tunnel-name]` hierarchy level. The `rsvp-te` configuration statement is also available for use in routing instances as a provider-tunnel option. When implemented as a provider-tunnel option, `rsvp-te` is used to configure the RSVP point-to-multipoint LSPs for multiprotocol BGP multicast VPNs, not to configure RSVP automatic mesh.

RSVP Reservation Styles

A reservation request includes options for specifying the reservation style. The reservation styles define how reservations for different senders within the same session are treated and how senders are selected.

Two options specify how reservations for different senders within the same session are treated:

- Distinct reservation—Each receiver establishes its own reservation with each upstream sender.
- Shared reservation—All receivers make a single reservation that is shared among many senders.

Two options specify how senders are selected:

- Explicit sender—List all selected senders.
- Wildcard sender—Select all senders, which then participate in the session.

The following reservation styles, formed by a combination of these four options, currently are defined:

- Fixed filter (FF)—This reservation style consists of distinct reservations among explicit senders. Examples of applications that use fixed-filter-style reservations are video applications and unicast applications, which both require flows that have a separate reservation for each sender. The fixed filter reservation style is enabled on RSVP LSPs by default.
- Wildcard filter (WF)—This reservation style consists of shared reservations among wildcard senders. This type of reservation reserves bandwidth for any and all senders, and propagates upstream toward all senders, automatically extending to new senders as they appear. A sample application for wildcard filter reservations is an audio application in which each sender transmits a distinct data stream. Typically, only a few senders are transmitting at any one time. Such a flow does not require a separate reservation for each sender; a single reservation is sufficient.
- Shared explicit (SE)—This reservation style consists of shared reservations among explicit senders. This type of reservation reserves bandwidth for a limited group of senders. A sample application is an audio application similar to that described for wildcard filter reservations.

RSVP Refresh Reduction

RSVP relies on soft-state to maintain the path and reservation states on each router. If the corresponding refresh messages are not sent periodically, the states eventually time out and reservations are deleted. RSVP also sends its control messages as IP datagrams with no reliability guarantee. It relies on periodic refresh messages to handle the occasional loss of Path or Resv messages.

The RSVP refresh reduction extensions, based on RFC 2961, addresses the following problems that result from relying on periodic refresh messages to handle message loss:

- Scalability—The scaling problem arises from the periodic transmission and processing overhead of refresh messages, which increases as the number of RSVP sessions increases.
- Reliability and latency—The reliability and latency problem stems from the loss of nonrefresh RSVP messages or one-time RSVP messages such as PathTear or PathErr. The time to recover from such a loss is usually tied to refresh interval and the keepalive timer.

The RSVP refresh reduction capability is advertised by enabling the refresh reduction (RR) capable bit in the RSVP common header. This bit is only significant between RSVP neighbors.

RSVP refresh reduction includes the following features:

- RSVP message bundling using the bundle message
- RSVP Message ID to reduce message processing overhead
- Reliable delivery of RSVP messages using Message ID, Message Ack, and Message Nack
- Summary refresh to reduce the amount of information transmitted every refresh interval

The RSVP refresh reduction specification (RFC 2961) allows you to enable some or all of the above capabilities on a router. It also describes various procedures that a router can use to detect the refresh reduction capabilities of its neighbor.

The Junos OS supports all of the refresh reduction extensions, some of which can be selectively enabled or disabled. The Junos OS supports Message ID and therefore can perform reliable message delivery only for Path and Resv messages.

For information about how to configure RSVP refresh reduction, see "Configuring RSVP Refresh Reduction" on page 1244.

MTU Signaling in RSVP

The maximum transmission unit (MTU) is the largest size packet or frame, in bytes, that can be sent in a network. An MTU that is too large might cause retransmissions. Too small an MTU might cause the router to send and handle relatively more header overhead and acknowledgments. There are default values for MTUs associated with various protocols. You can also explicitly configure an MTU on an interface.

When an LSP is created across a set of links with different MTU sizes, the ingress router does not know what the smallest MTU is on the LSP path. By default, the MTU for an LSP is 1,500 bytes.

If this MTU is larger than the MTU of one of the intermediate links, traffic might be dropped, because MPLS packets cannot be fragmented. Also, the ingress router is not aware of this type of traffic loss, because the control plane for the LSP would still function normally.

To prevent this type of packet loss in MPLS LSPs, you can configure MTU signaling in RSVP. This feature is described in RFC 3209. Juniper Networks supports the Integrated Services object for MTU signaling in RSVP. The Integrated Services object is described in RFCs 2210 and 2215. MTU signaling in RSVP is disabled by default.

To avoid packet loss due to MTU mismatches, the ingress router needs to do the following:

- Signal the MTU on the RSVP LSP—To prevent packet loss from an MTU mismatch, the ingress router needs to know what the smallest MTU value is along the path taken by the LSP. Once this MTU value is obtained, the ingress router can assign it to the LSP.
- Fragment packets—Using the assigned MTU value, packets that exceed the size of the MTU can be fragmented into smaller packets on the ingress router before they are encapsulated in MPLS and sent over the RSVP-signaled LSP.

Once both MTU signaling and packet fragmentation have been enabled on an ingress router, any route resolving to an RSVP LSP on this router uses the signaled MTU value. For information about how to configure this feature, see "Configuring MTU Signaling in RSVP" on page 1269.

The following sections describe how MTU signaling in RSVP works:

- "How the Correct MTU Is Signaled in RSVP" on page 1238
- "Determining an Outgoing MTU Value" on page 1239
- "MTU Signaling in RSVP Limitations" on page 1239

How the Correct MTU Is Signaled in RSVP

How the correct MTU is signaled in RSVP varies depending on whether the network devices (for example, routers) explicitly support MTU signaling in RSVP or not.

If the network devices support MTU signaling in RSVP, the following occur when you enable MTU signaling:

- The MTU is signaled from the ingress router to the egress router by means of the Adspec object. Before forwarding this object, the ingress router enters the MTU value associated with the interface over which the path message is sent. At each hop in the path, the MTU value in the Adspec object is updated to the minimum of the received value and the value of the outgoing interface.

- The ingress router uses the traffic specification (Tspec) object to specify the parameters for the traffic it is going to send. The MTU value signaled for the Tspec object at the ingress router is the maximum MTU value (9192 bytes for M Series and T Series routers, 9500 bytes for PTX Series Packet Transport Routers). This value does not change en route to the egress router.
- When the Adspec object arrives at the egress router, the MTU value is correct for the path (meaning it is the smallest MTU value discovered). The egress router compares the MTU value in the Adspec object to the MTU value in the Tspec object. It signals the smaller MTU using the Flowspec object in the Resv message.
- When the Resv object arrives at the ingress router, the MTU value in this object is used as the MTU for the next hops that use the LSP.

In a network where there are devices that do not support MTU signaling in RSVP, you might have the following behaviors:

- If the egress router does not support MTU signaling in RSVP, the MTU is set to 1,500 bytes by default.
- A Juniper Networks transit router that does not support MTU signaling in RSVP sets an MTU value of 1,500 bytes in the Adspec object by default.

Determining an Outgoing MTU Value

The outgoing MTU value is the smaller of the values received in the Adspec object compared to the MTU value of the outgoing interface. The MTU value of the outgoing interface is determined as follows:

- If you configure an MTU value under the `[family mpls]` hierarchy level, this value is signaled.
- If you do not configure an MTU, the `inet` MTU is signaled.

MTU Signaling in RSVP Limitations

The following are limitations to MTU signaling in RSVP:

- Changes in the MTU value might cause a temporary loss of traffic in the following situations:
  - For link protection and node protection, the MTU of the bypass is only signaled at the time the bypass becomes active. During the time it takes for the new path MTU to be propagated, packet loss might occur because of an MTU mismatch.
  - For fast reroute, the MTU of the path is updated only after the detour becomes active, causing a delay in an update to the MTU at the ingress router. Until the MTU is updated, packet loss might occur if there is an MTU mismatch.

    In both cases, only packets that are larger than the detour or bypass MTU are lost.

- When an MTU is updated, it triggers a change in the next hop. Any change in the next hop causes the route statistics to be lost.
- The minimum MTU supported for MTU signaling in RSVP is 1,488 bytes. This value prevents a false or incorrectly configured value from being used.
- For single-hop LSPs, the MTU value displayed by the `show` commands is the RSVP-signaled value. However, this MPLS value is ignored and the correct IP value is used.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
16.1	Starting with Junos OS Release 16.1, RSVP Traffic Engineering (TE) protocol extensions to support Refresh-interval Independent RSVP (RI-RSVP) defined RFC 8370 for fast reroute (FRR) facility protection were introduced to allow greater scalability of label-switched paths (LSPs) faster convergence times and decrease RSVP signaling message overhead from periodic refreshes.

RELATED DOCUMENTATION

RSVP Configuration | **1240**

Basic MPLS Configuration | **46**

RSVP Configuration

IN THIS SECTION

- Minimum RSVP Configuration | **1242**
- Configuring RSVP and MPLS | **1242**
- Configuring RSVP Interfaces | **1244**
- Configuring RSVP Node-ID Hellos | **1250**
- Example: Configuring RSVP-Signaled LSPs | **1251**
- Example: Configuring RSVP Automatic Mesh | **1257**
- Configuring Hello Acknowledgments for Nonsession RSVP Neighbors | **1262**

- Switching LSPs Away from a Network Node | **1263**
- Configuring RSVP Setup Protection | **1264**
- Configuring Load Balancing Across RSVP LSPs | **1265**
- Configuring RSVP Automatic Mesh | **1266**
- Configuring Timers for RSVP Refresh Messages | **1267**
- Preempting RSVP Sessions | **1268**
- Configuring MTU Signaling in RSVP | **1269**
- Configuring Ultimate-Hop Popping for RSVP-signaled MPLS LSPs | **1270**
- Configuring RSVP to Pop the Label on the Ultimate-Hop Router | **1274**
- Enabling Ultimate-Hop Popping on Point-to-Multipoint LSPs | **1275**
- Tracing RSVP Protocol Traffic | **1276**
- RSVP Graceful Restart | **1279**
- RSVP Graceful Restart Terminology | **1280**
- RSVP Graceful Restart Operation | **1281**
- Processing the Restart Cap Object | **1282**
- Configuring RSVP Graceful Restart | **1282**
- RSVP LSP Tunnels Overview | **1284**
- Example: RSVP LSP Tunnel Configuration | **1286**
- Configuring Link Management Protocol Peers | **1311**
- Configuring Link Management Protocol Traffic Engineering Links | **1312**
- Configuring Peer Interfaces in OSPF and RSVP | **1312**
- Defining Label-Switched Paths for the FA-LSP | **1313**
- Establishing FA-LSP Path Information | **1313**
- Option: Tearing Down RSVP LSPs Gracefully | **1314**

Minimum RSVP Configuration

To enable RSVP on a single interface, include the `rsvp` statement and specify the interface using the `interface` statement. This is the minimum RSVP configuration. All other RSVP configuration statements are optional.

```
rsvp {
    interface interface-name;
}
```

You can include these statements at the following hierarchy levels:

- `[edit protocols]`
- `[edit logical-systems logical-system-name protocols]`

To enable RSVP on all interfaces, substitute `all` for the *interface-name* variable.

If you have configured interface properties on a group of interfaces and want to disable RSVP on one of the interfaces, include the `disable` statement:

```
interface interface-name {
    disable;
}
```

You can include this statement at the following hierarchy levels:

- `[edit protocols rsvp interface interface-name ]`
- `[edit logical-systems logical-system-name protocols rsvp interface interface-name ]`

Configuring RSVP and MPLS

IN THIS SECTION

- Example: Configuring RSVP and MPLS | **1243**

The primary purpose of the Junos RSVP software is to support dynamic signaling within label-switched paths (LSPs). When you enable both MPLS and RSVP on a router, MPLS becomes a client of RSVP. No additional configuration is required to bind MPLS and RSVP.

You can configure MPLS to set up signaled paths by using the `label-switched-path` statement at the `[edit protocols mpls]` hierarchy level. Each LSP translates into a request for RSVP to initiate an RSVP session. This request is passed through the internal interface between label switching and RSVP. After examining the request information, checking RSVP states, and checking the local routing tables, RSVP initiates one session for each LSP. The session is sourced from the local router and is destined for the target of the LSP.

When an RSVP session is successfully created, the LSP is set up along the paths created by the RSVP session. If the RSVP session is unsuccessful, RSVP notifies MPLS of its status. It is up to MPLS to initiate backup paths or continue retrying the initial path.

To pass label-switching signaling information, RSVP supports four additional objects: Label Request object, Label object, Explicit Route object, and Record Route object. For an LSP to be set up successfully, all routers along the path must support MPLS, RSVP, and the four objects. Of the four objects, the Record Route object is not mandatory.

To configure MPLS and make it a client of RSVP, do the following:

- Enable MPLS on all routers that will participate in the label switching (this is, on all routers that might be part of a label-switching path).
- Enable RSVP on all routers and on all router interfaces that form the LSP.
- Configure the routers at the beginning of the LSP.

You can configure RSVP label-switched paths (LSPs) to use a delay metric for computing the path. To configure, use the CLI options that we've introduced under the `[edit protocols mpls label-switched-path name]` hierarchy.

Example: Configuring RSVP and MPLS

The following shows a sample configuration for a router at the beginning of an LSP:

```
[edit]
protocols {
    mpls {
        label-switched-path sf-to-london {
            to 192.168.1.4;
        }
    }
    rsvp {
        interface so-0/0/0;
    }
}
```

The following shows a sample configuration for all the other routers that form the LSP:

```
[edit]
protocols {
    mpls {
        interface so-0/0/0;
    }
    rsvp {
        interface so-0/0/0;
    }
}
```

Configuring RSVP Interfaces

IN THIS SECTION

- Configuring RSVP Refresh Reduction | **1244**
- Configuring the RSVP Hello Interval | **1247**
- Configuring RSVP Authentication | **1248**
- Configuring the Bandwidth Subscription for Class Types | **1248**
- Configuring the RSVP Update Threshold on an Interface | **1248**
- Configuring RSVP for Unnumbered Interfaces | **1250**

The following sections describe how to configure RSVP interfaces:

Configuring RSVP Refresh Reduction

You can configure RSVP refresh reduction on each interface by including the following statements in the interface configuration:

- `aggregate` and `reliable`—Enable all RSVP refresh reduction features: RSVP message bundling, RSVP message ID, reliable message delivery, and summary refresh.

  In order to have refresh reduction and reliable delivery, you must include the `aggregate` and `reliable` statements.

- `no-aggregate`—Disable RSVP message bundling and summary refresh.

- no-reliable—Disable RSVP message ID, reliable message delivery, and summary refresh.

For more information on RSVP refresh reduction, see "RSVP Refresh Reduction" on page 1237.

If the no-reliable statement is configured on the router (reliable message delivery is disabled), the router accepts RSVP messages that include the Message ID object but ignores the Message ID object and continues performing standard message processing. No error is generated in this case, and RSVP operates normally.

However, not all combinations between two neighbors with different refresh reduction capabilities function correctly. For example, a router is configured with either the aggregate statement and no-reliable statement or with the reliable and no-aggregate statements. If an RSVP neighbor sends a Summary Refresh object to this router, no error is generated, but the Summary Refresh object cannot be processed. Consequently, RSVP states can time out on this router if the neighbor is relying only on Summary Refresh to refresh those RSVP states.

We recommend, unless there are specific requirements, that you configure RSVP refresh reduction in a similar manner on each RSVP neighbor.

To enable all RSVP refresh reduction features on an interface, include the aggregate statement:

```
aggregate;
```

You can include this statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name*]
- [edit logical-systems *logical-system-name* protocols rsvp interface *interface-name*]

To disable RSVP message bundling and summary refresh, include the no-aggregate statement:

```
no-aggregate;
```

You can include this statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name*]
- [edit logical-systems *logical-system-name* protocols rsvp interface *interface-name*]

To enable RSVP message ID and reliable message delivery on an interface, include the reliable statement:

```
reliable;
```

You can include this statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name*]
- [edit logical-systems *logical-system-name* protocols rsvp interface *interface-name*]

To disable RSVP message ID, reliable message delivery, and summary refresh, include the no-reliable statement:

```
no-reliable;
```

You can include this statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name*]
- [edit logical-systems *logical-system-name* protocols rsvp interface *interface-name*]

Determining the Refresh Reduction Capability of RSVP Neighbors

To determine the RSVP refresh reduction capability of an RSVP neighbor, you need the following information:

- The RR bit advertised by the neighbor
- The local configuration of RSVP refresh reduction
- The actual RSVP messages received from the neighbor

To obtain this information, you can issue a show rsvp neighbor detail command. Sample output follows:

```
user@host> show rsvp neighbor detail
RSVP neighbor: 6 learned
    Address: 192.168.224.178 via: fxp1.0 status: Up
      Last changed time: 10:06, Idle: 5 sec, Up cnt: 1, Down cnt: 0
      Message received: 36
      Hello: sent 69, received: 69, interval: 9 sec
      Remote instance: 0x60b8feba, Local instance: 0x74bc7a8d
      Refresh reduction:  not operational

    Address: 192.168.224.186 via: fxp2.0 status: Down
      Last changed time: 10:17, Idle: 40 sec, Up cnt: 0, Down cnt: 0
      Message received: 6
      Hello: sent 20, received: 0, interval: 9 sec
      Remote instance: 0x0, Local instance: 0x2ae1b339
```

```
    Refresh reduction:  incomplete
      Remote end: disabled, Ack-extension: enabled

Address: 192.168.224.188 via: fxp2.0 status: Up
  Last changed time: 4:15, Idle: 0 sec, Up cnt: 1, Down cnt: 0
  Message received: 55
  Hello: sent 47, received: 31, interval: 9 sec
  Remote instance: 0x6436a35b, Local instance: 0x663849f0
  Refresh reduction:  operational
    Remote end: enabled, Ack-extension: enabled
```

For more information on the `show rsvp neighbor detail` command.

Configuring the RSVP Hello Interval

RSVP monitors the status of the interior gateway protocol (IGP) (IS-IS or OSPF) neighbors and relies on the IGP protocols to detect when a node fails. If an IGP protocol declares a neighbor down (because hello packets are no longer being received), RSVP also brings down that neighbor. However, the IGP protocols and RSVP still act independently when bringing a neighbor up.

For Juniper Networks routers, configuring a shorter or longer RSVP hello interval has no impact on whether or not an RSVP session is brought down. RSVP sessions are kept up even if RSVP hello packets are no longer being received. RSVP sessions are maintained until either the router stops receiving IGP hello packets or the RSVP Path and Resv messages time out. However, starting from Junos OS Release 16.1, when RSVP hello messages time-out, the RSVP sessions are brought down.

The RSVP hello interval might also be impacted when another vendor's equipment brings down an RSVP session. For example, a neighboring non-Juniper Networks router might be configured to monitor RSVP hello packets.

To modify how often RSVP sends hello packets, include the `hello-interval` statement:

```
hello-interval seconds;
```

NOTE: Starting in release 16.1 Junos sends RSVP hello messages over a bypass LSP when one is available. See *no-node-hello-on-bypass* for information on how to revert to the historic behavior of sending hellos over the IGP next hop.

For a list of hierarchy levels at which you can include this statement, see the statement summary section.

Configuring RSVP Authentication

All RSVP protocol exchanges can be authenticated to guarantee that only trusted neighbors participate in setting up reservations. By default, RSVP authentication is disabled.

RSVP authentication uses a Hashed Message Authentication Code (HMAC)-MD5 message-based digest. This scheme produces a message digest based on a secret authentication key and the message contents. (The message contents also include a sequence number.) The computed digest is transmitted with RSVP messages. Once you have configured authentication, all received and transmitted RSVP messages with all neighbors are authenticated on this interface.

MD5 authentication provides protection against forgery and message modification. It also can prevent replay attacks. However, it does not provide confidentiality, because all messages are sent in clear text.

By default, authentication is disabled. To enable authentication, configure a key on each interface by including the `authentication-key` statement:

```
authentication-key key;
```

You can include this statement at the following hierarchy levels:

- `[edit protocols rsvp]`
- `edit logical-systems logical-system-name protocols rsvp`
- `[edit protocols rsvp interface interface-name]`
- `[edit logical-systems logical-system-name protocols rsvp interface interface-name]`

To authenticate node hellos or remote messages between the Point of Local Repair (PLR) and the Merge Point (MP), enable authentication-key at the [edit protocols rsvp] hierarchy level.

Configuring the Bandwidth Subscription for Class Types

By default, RSVP allows 100 percent of the bandwidth for a class type to be used for RSVP reservations. When you oversubscribe a class type for a multiclass LSP, the aggregate demand of all RSVP sessions is allowed to exceed the actual capacity of the class type.

For detailed instructions on how to configure the bandwidth subscription for class types, see "Configuring the Bandwidth Subscription Percentage for LSPs" on page 679.

Configuring the RSVP Update Threshold on an Interface

The interior gateway protocols (IGPs) maintain the traffic engineering database, but the current available bandwidth on the traffic engineering database links originates from RSVP. When a link's bandwidth

changes, RSVP informs the IGPs, which can then update the traffic engineering database and forward the new bandwidth information to all network nodes. The network nodes then know how much bandwidth is available on the traffic engineering database link (local or remote), and CSPF can correctly compute the paths.

However, IGP updates can consume excessive system resources. Depending on the number of nodes in a network, it might not be desirable to perform an IGP update for small changes in bandwidth. By configuring the `update-threshold` statement at the `[edit protocols rsvp]` hierarchy level, you can adjust the threshold at which a change in the reserved bandwidth triggers an IGP update.

You can configure a value of from 0.001 percent through 20 percent (the default is 10 percent) for when to trigger an IGP update. If the change in the reserved bandwidth is greater than or equal to the configured threshold percentage of the static bandwidth on that interface, then an IGP update occurs. For example, if you have configured the `update-threshold` statement to be 15 percent and the router discovers that the reserved bandwidth on a link has changed by 10 percent of the link bandwidth, RSVP does not trigger an IGP update. However, if the reserved bandwidth on a link changes by 20 percent of the link bandwidth, RSVP triggers an IGP update.

You can also configure the threshold as an absolute value using the `threshold-value` option under the `update-threshold` statement.

If the threshold-value is configured to greater than 20% of bandwidth on that link, the threshold-value is capped at 20% of bandwidth.

For instance, if bandwidth on an interface is 1Gbps, and the `threshold-value` is configured greater than 200Mbps, the `threshold-value` is capped at 200Mbps. The *threshold-percent* is displayed as 20.000% and the `threshold-value` as 200Mbps.

NOTE: The two options, *threshold-percent* and `threshold-value`, are mutually exclusive. You can configure only one option at a given point in time to generate an IGP update for lower bandwidth reservations. As a result, when one option is configured, the other option is calculated and displayed on the CLI.

For instance, on a link of 1Gbps, if the *threshold-percent* is configured to 5%, the `threshold-value` is calculated and displayed as 50Mbps. Similarly, if the `threshold-value` is configured to 50m, then the *threshold-percent* is calculated and displayed as 5%.

To adjust the threshold at which a change in the reserved bandwidth triggers an IGP update, include the *update-threshold* statement. Because of the update threshold, it is possible for Constrained Shortest Path First (CSPF) to compute a path using outdated traffic engineering database bandwidth information on a link. If RSVP attempts to establish an LSP over that path, it might find that there is insufficient bandwidth on that link. When this happens, RSVP triggers an IGP traffic engineering database update, flooding the updated bandwidth information on the network. CSPF can then recompute the path by using the updated bandwidth information, and attempt to find a different path, avoiding the congested link. Note that this functionality is the default and does not need any additional configuration.

You can configure the `rsvp-error-hold-time` statement at the `[edit protocols mpls]` hierarchy level or the `[edit logical-systems` *`logical-system-name`* `protocols mpls]` hierarchy level to improve the accuracy of the traffic engineering database (including the accuracy of bandwidth estimates for LSPs) using information provided by PathErr messages. See "Improving Traffic Engineering Database Accuracy with RSVP PathErr Messages" on page 1769.

Configuring RSVP for Unnumbered Interfaces

The Junos OS supports RSVP traffic engineering over unnumbered interfaces. Traffic engineering information about unnumbered links is carried in the IGP traffic engineering extensions for OSPF and IS-IS as described in RFC 4203, *OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)*, and RFC 4205, *Intermediate System to Intermediate System (IS-IS) Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)*. Unnumbered links can also be specified in the MPLS traffic engineering signaling as described in RFC 3477, *Signalling Unnumbered Links in Resource ReSerVation Protocol - Traffic Engineering (RSVP-TE)*. This feature allows you avoid having to configure IP addresses for each interface participating in the RSVP-signaled network.

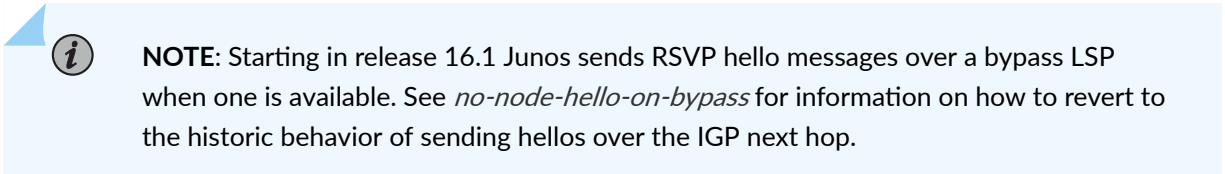
To configure RSVP for unnumbered interfaces, you must configure the router with a router ID using the `router-id` statement specified at the `[edit routing-options]` hierarchy level. The router ID must be available for routing (you can typically use the loopback address). The RSVP control messages for the unnumbered links are sent using the router ID address (rather than a randomly selected address).

To configure link protection and fast reroute on a router with unnumbered interfaces enabled, you must configure at least two addresses. We recommend that you configure a secondary interface on the loopback in addition to configuring the router ID.

Configuring RSVP Node-ID Hellos

You can configure node-ID based RSVP hellos to ensure that Juniper Networks routers can interoperate with the equipment of other vendors. By default, Junos OS uses interface-based RSVP hellos. Node-ID based RSVP hellos are specified in RFC 4558, *Node-ID Based Resource Reservation Protocol (RSVP) Hello: A Clarification Statement*. RSVP node-ID hellos are useful if you have configured BFD to detect problems over RSVP interfaces, allowing you to disable interface hellos for these interfaces. You can also use node-ID hellos for graceful restart procedures.

Node-ID hellos can be enabled globally for all RSVP neighbors. By default, node-ID hello support is disabled. If you have not enabled RSVP node IDs on the router, the Junos OS does not accept any node-ID hello packets.



NOTE: Starting in release 16.1 Junos sends RSVP hello messages over a bypass LSP when one is available. See *no-node-hello-on-bypass* for information on how to revert to the historic behavior of sending hellos over the IGP next hop.

To enable RSVP node-ID hellos globally on the router, include the *node-hello* statement at the following hierarchy levels:

- `[edit protocols rsvp]`
- `[edit logical-systems logical-systems-name protocols rsvp]`

You can also explicitly disable RSVP interface hellos globally. This type of configuration might be necessary in networks where the Juniper Networks router has numerous RSVP connections with equipment from other vendors. However, if you disable RSVP interface hellos globally, you can also configure a hello interval on an RSVP interface using the *hello-interval* statement. This configuration disables RSVP interface hellos globally, but enables RSVP interface hellos on the specified interface (the RSVP interface you configure the `hello-interval` statement on). This configuration might be necessary in a heterogeneous network in which some devices support RSVP node-ID hellos and other devices support RSVP interface hellos.

To disable RSVP interface hellos globally on the router, include the *no-interface-hello* statement at the following hierarchy levels:

- `[edit protocols rsvp]`
- `[edit logical-systems logical-systems-name protocols rsvp]`

Example: Configuring RSVP-Signaled LSPs

IN THIS SECTION

- Requirements | **1252**
- Overview and Topology | **1252**
- Configuration | **1253**
- Verification | **1255**

This example shows how to create an LSP between routers in an IP network using RSVP as the signaling protocol.

Requirements

Before you begin, delete security services from the device. See Example: Deleting Security Services.

Overview and Topology

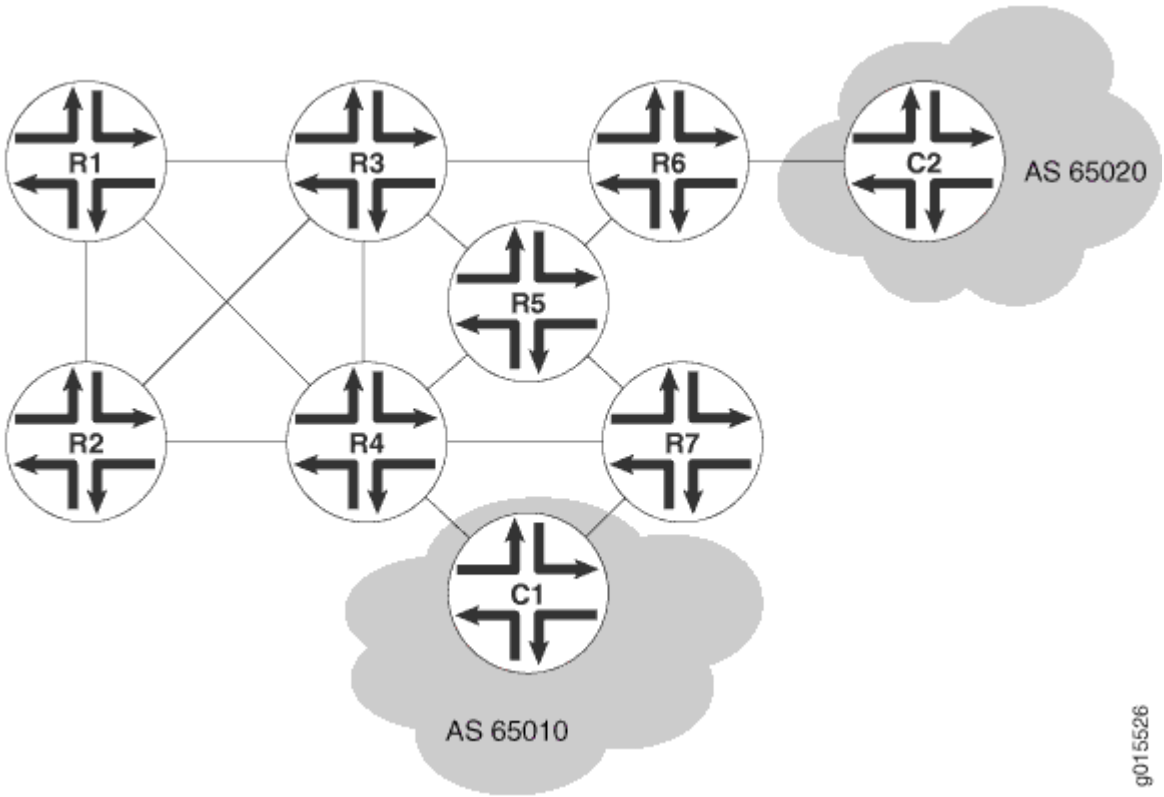
IN THIS SECTION

- Topology | 1252

Using RSVP as a signaling protocol, you can create LSPs between routers in an IP network. In this example, you configure a sample network as shown in Figure 70 on page 1252.

Topology

Figure 70: Typical RSVP-Signaled LSP



To establish an LSP between routers, you must individually enable the MPLS family and configure RSVP on each of the transit interfaces in the MPLS network. This example shows how to enable MPLS and configure RSVP on the ge-0/0/0 transit interface. Additionally, you must enable the MPLS process on all of the MPLS interfaces in the network.

This example shows how to define an LSP from R1 to R7 on the ingress router (R1) using R7's loopback address (10.0.9.7). The configuration reserves 10 Mbps of bandwidth. Additionally, the configuration disables the CSPF algorithm, ensuring that Hosts C1 and C2 use the RSVP-signaled LSP that correspond to the network IGP's shortest path.

Configuration

IN THIS SECTION

- Procedure | **1253**

Procedure

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the `[edit]` hierarchy level.

```
set interfaces ge-0/0/0 unit 0 family mpls
set protocols rsvp interface ge-0/0/0.0
set protocols mpls label-switched-path r1-r7 to 10.0.9.7
set protocols mpls label-switched-path r1-r7 bandwidth 10m
set protocols mpls interface all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure RSVP:

1. Enable the MPLS family on all transit interfaces in the MPLS network.

```
[edit]
user@host# set interfaces ge-0/0/0 unit 0 family mpls
```

2. Enable RSVP on each transit interface in the MPLS network.

```
[edit]
user @host# set protocols rsvp interface ge-0/0/0
```

3. Enable the MPLS process on the transit interface in the MPLS network.

```
[edit]
user@host# set protocols mpls interface ge-0/0/0
```

4. Define the LSP on the ingress router.

```
[edit protocols mpls]
user@host# set label-switched-path r1-r7 to 10.0.9.7
```

5. Reserve 10 Mbps of bandwidth on the LSP.

```
[edit protocols mpls]
user @host# set label-switched-path r1-r7 bandwidth 10m
```

Results

Confirm your configuration by entering the `show` command from configuration mode. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this `show` command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
user@host# show
...
    interfaces {
    ge-0/0/0 {
```

```
        family mpls;
        }
    }
    }
    ...
    protocols {
        rsvp {
            interface ge-0/0/0.0;
        }
        mpls {
            label-switched-path r1-r7 {
                to 10.0.9.7;
                bandwidth 10m;
            }
            interface all;
        }
    }
...
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

IN THIS SECTION

- Verifying RSVP Neighbors | **1255**
- Verifying RSVP Sessions | **1256**
- Verifying the Presence of RSVP-Signaled LSPs | **1257**

To confirm that the configuration is working properly, perform these tasks:

Verifying RSVP Neighbors

Purpose

Verify that each device shows the appropriate RSVP neighbors—for example, that Router R1 in Figure 70 on page 1252 lists both Router R3 and Router R2 as RSVP neighbors.

Action

From the CLI, enter the `show rsvp neighbor` command.

```
user@r1> show rsvp neighbor
RSVP neighbor: 2 learned
Address            Idle Up/Dn LastChange HelloInt HelloTx/Rx
10.0.6.2              0  3/2       13:01        3   366/349
10.0.3.3              0  1/0       22:49        3   448/448
```

The output shows the IP addresses of the neighboring routers. Verify that each neighboring RSVP router loopback address is listed.

Verifying RSVP Sessions

Purpose

Verify that an RSVP session has been established between all RSVP neighbors. Also, verify that the bandwidth reservation value is active.

Action

From the CLI, enter the `show rsvp session detail` command.

```
user@r1> show rsvp session detail
Ingress RSVP: 1 sessions

10.0.9.7
  From: 10.0.6.1, LSPstate: Up, ActiveRoute: 0
  LSPname: r1-r7, LSPpath: Primary
  Bidirectional, Upstream label in: -, Upstream label out: -
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 100000
  Resv style: 1 FF, Label in: -, Label out: 100000
  Time left:    -,  Since: Thu Jan 26 17:57:45 2002
  Tspec: rate 10Mbps size 10Mbps peak Infbps m 20 M 1500
  Port number: sender 3 receiver 17 protocol 0
  PATH rcvfrom: localclient
  PATH sentto: 10.0.4.13 (ge-0/0/1.0) 1467 pkts
  RESV rcvfrom: 10.0.4.13  (ge-0/0/1.0) 1467 pkts
```

```
  Record route: <self> 10.0.4.13  10.0.2.1  10.0.8.10
```

The output shows the detailed information, including session IDs, bandwidth reservation, and next-hop addresses, for each established RSVP session. Verify the following information:

- Each RSVP neighbor address has an entry for each neighbor, listed by loopback address.
- The state for each LSP session is **Up**.
- For **Tspec**, the appropriate bandwidth value, **10Mbps**, appears.

Verifying the Presence of RSVP-Signaled LSPs

Purpose

Verify that the routing table of the entry (ingress) router has a configured LSP to the loopback address of the other router. For example, verify that the **inet.3** routing table of the R1 entry router in Figure 70 on page 1252 has a configured LSP to the loopback address of Router R7.

Action

From the CLI, enter the `show route table inet.3` command.

```
user@r1> show route table inet.3
inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.9.7/32         *[RSVP/7] 00:05:29, metric 10
                    > to 10.0.4.17 via ge-0/0/0.0, label-switched-path r1-r7
```

The output shows the RSVP routes that exist in the **inet.3** routing table. Verify that an RSVP-signaled LSP is associated with the loopback address of the exit (egress) router, R7, in the MPLS network.

Example: Configuring RSVP Automatic Mesh

IN THIS SECTION

- Requirements | **1258**
- Overview | **1258**

Configuration | 1259
Verification | 1261

Service providers often use BGP and MPLS VPNs to efficiently scale the network while delivering revenue-generating services. In these environments, BGP is used to distribute the VPN routing information across the service provider's network, while MPLS is used to forward that VPN traffic from one VPN site to another.

When adding a new PE router that will participate in BGP and MPLS VPNs, all of the previously existing PE routers must be configured to peer with the new PE router for both BGP and MPLS. As each new PE router is added to the service provider's network, the configuration burden soon becomes too much to handle.

The configuration requirements for BGP peering can be reduced with the use of route reflectors. In RSVP signaled MPLS networks, RSVP automatic mesh can minimize the configuration burden for the MPLS portion of the network. Configuring `rsvp-te` on all PE routers allows RSVP to automatically create the needed LSPs when a new PE router is added.

Requirements

This example uses the following hardware and software components:

- A router running Junos OS Release 10.1 or later.
- A BGP and MPLS VPN using RSVP as the MPLS label-switched path (LSP) signaling protocol.

Overview

IN THIS SECTION

Topology | 1259

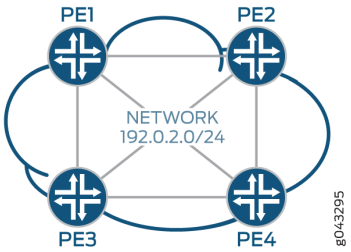
This example shows how to configure RSVP automatic mesh on a PE router using the `rsvp-te` configuration statement. In order for RSVP automatic mesh to function properly, all of the PE routers in the full mesh configuration must have the `rsvp-te` statement configured. This ensures that any new PE routers that are added later will also benefit from the automatic mesh feature, provided that they too are configured with the `rsvp-te` statement.

Given this requirement, this example only shows the configuration on the newly added PE router. It is assumed that RSVP automatic mesh has already been configured on the existing PE routers.

Topology

In Figure 71 on page 1259, there are three existing PE routers, PE1, PE2, and PE3, in the topology. PE4 has been added, and RSVP automatic mesh will be configured. The cloud represents the service provider network, and the network address, 192.0.2.0/24, is shown in the center of the figure.

Figure 71: Service Provider Network with PE Routers



Configuration

IN THIS SECTION

- CLI Quick Configuration | **1260**
- Configuring RSVP Automatic Mesh | **1260**
- Results | **1261**

Configuring RSVP automatic mesh involves performing these tasks:

- Enabling the `rsvp-te` configuration statement at the `[edit routing-options dynamic-tunnels` *`dynamic-tunnel-name`*`]` hierarchy level.
- Configuring the required `destination-networks` element.

  This configuration element specifies the IPv4 prefix range for the destination network. Only tunnels within the specified prefix range can be created.

- Configuring the required `label-switched-path-template` element.

  This configuration element takes either `default-template` or the name of a preconfigured LSP template as an argument. The `default-template` is a system-defined template that requires no user configuration.

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the `[edit]` hierarchy level.

PE4 Router

```
set routing-options dynamic-tunnels dt-1 rsvp-te rsvp-te-1 label-switched-path-template default-template
set routing-options dynamic-tunnels dt-1 rsvp-te rsvp-te-1 destination-networks 192.0.2.0/24
```

Configuring RSVP Automatic Mesh

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To enable RSVP automatic mesh:

1. Configure `rsvp-te` at the `[edit routing-options dynamic-tunnels]` hierarchy level.

```
[edit routing-options dynamic-tunnels]
user@PE4# set dt-1 rsvp-te rsvp-te-1 label-switched-path-template default-template
```

2. Configure `destination-networks` at the `[edit routing-options dynamic-tunnels]` hierarchy level.

```
[edit routing-options dynamic-tunnels]
user@PE4# set dt-1 rsvp-te rsvp-te-1 destination-networks 192.0.2.0/24
```

Results

Issue the `show` command from the `[edit routing-options dynamic-tunnels]` hierarchy level to see the results of your configuration:

```
[edit routing-options dynamic-tunnels]
user@PE4#show
dt-1 {
    rsvp-te rsvp-te-1 {
        label-switched-path-template {
            default-template;
        }
        destination-networks {
            192.0.2.0/24;
        }
    }
}
```

Verification

IN THIS SECTION

- Verifying the Existence of RSVP Automatic Mesh Tunnels on Router PE4 | **1261**
- Verifying the Existence of MPLS LSPs on Router PE4 | **1262**

Verifying the Existence of RSVP Automatic Mesh Tunnels on Router PE4

Purpose

To verify the operation of the newly configured PE4 router, issue the `show dynamic-tunnels database` command from operational mode. This command will show the destination network to which dynamic tunnels can be created.

Action

```
user@PE4> show dynamic-tunnels database
Table: inet.3
Destination-network: 192.0.2.0/24
```

Verifying the Existence of MPLS LSPs on Router PE4

Purpose

To verify the existence of MPLS LSPs on the PE4 router, issue the `show mpls lsp` command from operational mode. This command will show the state of the MPLS LSPs.

Action

```
user@PE4> show mpls lsp
```

```
Ingress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Egress LSP: 3 sessions
To              From            State   Rt Style Labelin Labelout LSPname
192.0.2.104     192.0.2.103     Up       0  1 FF       3        - PE2-PE4
192.0.2.104     192.0.2.102     Up       0  1 FF       3        - PE2-PE4
192.0.2.104     192.0.2.101     Up       0  1 FF       3        - PE1-PE4
Total 3 displayed, Up 3, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Configuring Hello Acknowledgments for Nonsession RSVP Neighbors

The `hello-acknowledgements` statement controls the hello acknowledgment behavior between RSVP neighbors regardless of whether or not they are in the same session.

Hello messages received from RSVP neighbors that are not part of a common RSVP session are discarded. If you configure the `hello-acknowledgements` statement at the `[edit protocols rsvp]` hierarchy level, hello messages from nonsession neighbors are acknowledged with a hello acknowledgment message. When hellos are received from nonsession neighbors, an RSVP neighbor relationship is created and

periodic hello messages can now be received from the nonsession neighbor. The `hello-acknowledgements` statement is disabled by default. Configuring this statement allows RSVP-capable routers to be discovered using hello packets and verifies that the interface is able to receive RSVP packets before sending any MPLS LSP setup messages.

Once you enable hello acknowledgments for nonsession RSVP neighbors, the router continues to acknowledge hello messages from any nonsession RSVP neighbors unless the interface itself goes down or you change the configuration. Interface-based neighbors are not automatically aged out.

```
hello-acknowledgements;
```

You can include this statement at the following hierarchy levels:

- `[edit protocols rsvp]`
- `[edit logical-systems` *`logical-system-name`* `protocols rsvp]`

Switching LSPs Away from a Network Node

You can configure the router to switch active LSPs away from a network node using a bypass LSP enabled for an interface. This feature might be used to maintain active networks when a device needs to be replaced without interrupting traffic transiting the network. The LSPs can be either static or dynamic.

1. You first need to configure either link or node protection for the traffic that needs to pass around the network device you intend to disable. To function properly, the bypass LSP must use a different logical interface than the protected LSP.
2. To prepare the router to begin switching traffic away from a network node, configure the `always-mark-connection-protection-tlv` statement:

   ```
   always-mark-connection-protection-tlv;
   ```

   The router then marks all OAM traffic transiting this interface in preparation for switching the traffic to an alternate path based on the OAM functionality.

   You can configure this statement at the following hierarchy levels:

   - `[edit protocols mpls interface` *`interface-name`*`]`
   - `[edit logical-systems` *`logical-system-name`* `protocols mpls interface` *`interface-name`*`]`
3. You then need to configure the `switch-away-lsps` statement to switch the traffic from the protected LSP to the bypass LSP, effectively bypassing the default downstream network device. The actual link itself is not brought down by this configuration.

To configure the router to switch traffic away from a network node, configure the `switch-away-lsps` statement:

```
switch-away-lsps;
```

You can configure this statement at the following hierarchy levels:

- `[edit protocols mpls interface` *`interface-name`*`]`
- `[edit logical-systems` *`logical-system-name`* `protocols mpls interface` *`interface-name`*`]`

Note the following limitations related to switching active LSPs away from a network node:

- The switch-away feature is supported on MX Series routers only.
- The switch-away feature is not supported for switching traffic from primary point-to-multipoint LSPs to bypass point-to-multipoint LSPs. If you configure the `switch-away-lsps` statement for a point-to-multipoint LSP, traffic is not switched to the bypass point-to-multipoint LSP.
- If you configure the switch-away feature on an interface along the path of a dynamic LSP, new dynamic LSPs cannot be established over that path. The switch-away feature prevents the make-before-break behavior of RSVP-signaled LSPs. The make-before-break behavior normally causes the router to first attempt to re-signal a dynamic LSP before tearing down the original.

Configuring RSVP Setup Protection

You can configure the facility-backup fast reroute mechanism to provide setup protection for LSPs which are in the process of being signaled. Both point-to-point LSPs and point-to-multipoint LSPs are supported. This feature is applicable in the following scenario:

1. A failed link or node is present on the strict explicit path of an LSP before the LSP is signaled.
2. There is also a bypass LSP protecting the link or node.
3. RSVP signals the LSP through the bypass LSP. The LSP appears as if it was originally set up along its primary path and then failed over to the bypass LSP because of the link or node failure.
4. When the link or node has recovered, the LSP can be automatically reverted to the primary path.

You should configure the `setup-protection` statement at the `[edit protocols rsvp]` on each of the routers along the LSP path on which you want to enable LSP setup protection. You should also configure IGP traffic engineering on all of the routers on the LSP path. You can issue a `show rsvp session` command to determine whether or not the LSP has setup protection enabled on a router acting as a point of local repair (PLR) or a merge point.

To enable RSVP setup protection, include the `setup-protection` statement

```
setup-protection;
```

You can include this statement at the following hierarchy levels:

- `[edit protocols rsvp]`
- `[edit logical-systems` *`logical-system-name`* `protocols rsvp]`

Configuring Load Balancing Across RSVP LSPs

By default, when you have configured several RSVP LSPs to the same egress router, the LSP with the lowest metric is selected and carries all traffic. If all of the LSPs have the same metric, one of the LSPs is selected at random and all traffic is forwarded over it.

Alternatively, you can load-balance traffic across all of the LSPs by enabling per-packet load balancing.

To enable per-packet load balancing on an ingress LSP, configure the `policy-statement` statement as follows:

```
[edit policy-options]
policy-statement policy-name {
    then {
        load-balance per-packet;
    }
    accept;
}
```

You then need to apply this statement as an export policy to the forwarding table.

Once per-packet load balancing is applied, traffic is distributed equally between the LSPs (by default).

You need to configure per-packet load balancing if you want to enable PFE fast reroute. To enable PFE fast reroute, include the `policy-statement` statement for per-packet load balancing shown in this section in the configuration of each of the routers where a reroute might take place. See also "Configuring Fast Reroute" on page 580.

You can also load-balance the traffic between the LSPs in proportion to the amount of bandwidth configured for each LSP. This capability can better distribute traffic in networks with asymmetric bandwidth capabilities across external links, since the configured bandwidth of an LSP typically reflects the traffic capacity of that LSP.

To configure RSVP LSP load balancing, include the load-balance statement with the bandwidth option:

```
load-balance {
    bandwidth;
}
```

You can configure this statement at the following hierarchy levels:

- [edit protocols rsvp]
- [edit logical-systems *logical-system-name* protocols rsvp]

Keep the following information in mind when you use the load-balance statement:

- If you configure the load-balance statement, the behavior of currently running LSPs is not altered. To force currently running LSPs to use the new behavior, you can issue a clear mpls lsp command.
- The load-balance statement only applies to ingress LSPs that have per-packet load balancing enabled.
- For Differentiated Services–aware traffic engineered LSPs, the bandwidth of an LSP is calculated by summing the bandwidth of all of the class types.

Configuring RSVP Automatic Mesh

You can configure RSVP to establish point-to-point label-switched paths (LSPs) automatically for any new PE router added to a full mesh of LSPs. To enable this feature, you must configure the rsvp-te statement on all of the PE routers in the full mesh.

NOTE: You cannot configure RSVP automatic mesh in conjunction with CCC. CCC cannot use the dynamically generated LSPs.

To configure RSVP automatic mesh, include the rsvp-te statement:

```
rsvp-te {
    destination-networks network-prefix;
    label-switched-path-template (Multicast) {
        default-template;
        template-name;
    }
}
```

You can configure these statements at the following hierarchy levels:

- [edit routing-options dynamic-tunnels *tunnel-name*]
- [edit logical-systems *logical-system-name* routing-options dynamic-tunnels *tunnel-name*]

You must also configure the following statements to enable RSVP automatic mesh:

- destination-networks—Specify the IP version 4 (IPv4) prefix range for the destination network. Dynamic tunnels within the specified IPv4 prefix range can be initiated.
- label-switched-path-template (Multicast)—You can configure either the default template explicitly using the default-template option, or you can configure an LSP template of your own using the *template-name* option. The LSP template acts as a model configuration for the dynamically generated LSPs.

Configuring Timers for RSVP Refresh Messages

RSVP uses two related timing parameters:

- refresh-time—The refresh time controls the interval between the generation of successive RSVP state refresh messages. The default value for the refresh-time (R) is 1200 seconds or 20 minutes as recommended in RFC 8370. If you configure the set protocols rsvp no-enhanced-frr-bypass statement, the refresh-time is set to 30 seconds. To avoid synchronization of refresh messages in the network, the refresh time for a state is randomly set to a value in the range 0.5R and 1.5R as specified in RFC 2205.

  Refresh messages include path and Resv messages. Refresh messages are sent periodically so that reservation states in neighboring nodes do not time out. Each path and Resv message carries the refresh timer value, and the receiving node extracts this value from the messages.

- keep-multiplier—The keep multiplier is a small, locally configured integer from 1 through 255. The default value is 3. It indicates the number of messages that can be lost before a particular state is declared stale and must be deleted. The keep multiplier directly affects the lifetime of an RSVP state.

To determine the lifetime of a reservation state, use the following formula:

```
lifetime = (keep-multiplier + 0.5) x (1.5 x refresh-time)
```

In the worst case, (*keep-multiplier* – 1) successive refresh messages must be lost before a reservation state is deleted.

By default, the refresh timer value is 1200 seconds. If you configure the no-enhanced-frr-bypass statement, the refresh-timer value is 30 seconds. To modify this value, include the refresh-time statement:

```
refresh-time seconds;
```

You can include this statement at the following hierarchy levels:

- [edit protocols rsvp]
- [edit logical-systems *logical-system-name* protocols rsvp]

The default value of the keep multiplier is 3. To modify this value, include the keep-multiplier statement:

```
keep-multiplier number;
```

You can include this statement at the following hierarchy levels:

- [edit protocols rsvp]
- [edit logical-systems *logical-system-name* protocols rsvp]

Preempting RSVP Sessions

Whenever bandwidth is insufficient to handle all RSVP sessions, you can control the preemption of RSVP sessions. By default, an RSVP session is preempted only by a new higher-priority session.

To always preempt a session when the bandwidth is insufficient, include the preemption statement with the aggressive option:

```
preemption aggressive;
```

You can include this statement at the following hierarchy levels:

- [edit protocols rsvp]
- [edit logical-systems *logical-system-name* protocols rsvp]

To disable RSVP session preemption, include the preemption statement with the disabled option:

```
preemption disabled;
```

To return to the default (that is, preempt a session only for a new higher-priority session), include the preemption statement with the normal option:

```
preemption normal;
```

You can include this statement at the following hierarchy levels:

- [edit protocols rsvp]
- [edit logical-systems *logical-system-name* protocols rsvp]

Configuring MTU Signaling in RSVP

IN THIS SECTION

- Enabling MTU Signaling in RSVP | **1269**
- Enabling Packet Fragmentation | **1270**

To configure maximum transmission unit (MTU) signaling in RSVP, you need to configure MPLS to allow IP packets to be fragmented before they are encapsulated in MPLS. You also need to configure MTU signaling in RSVP. For troubleshooting purposes, you can configure MTU signaling alone without enabling packet fragmentation.

To configure MTU signaling in RSVP, include the `path-mtu` statement:

```
path-mtu {
    allow-fragmentation;
    rsvp {
        mtu-signaling;
    }
}
```

You can include this statement at the following hierarchy levels:

- `[edit protocols mpls]`
- `[edit logical-systems` *`logical-system-name`* `protocols mpls]`

The following sections describe how to enable packet fragmentation and MTU signaling in RSVP:

Enabling MTU Signaling in RSVP

To enable MTU signaling in RSVP, include the `rsvp mtu-signaling` statement:

```
rsvp mtu-signaling;
```

You can include this statement at the following hierarchy levels:

- `[edit protocols mpls path-mtu]`
- `[edit logical-systems` *`logical-system-name`* `protocols mpls path-mtu]`

Once you have committed the configuration, changes in the MTU signaling behavior for RSVP take effect the next time the path is refreshed.

You can configure the `mtu-signaling` statement by itself at the `[edit protocols mpls path-mtu rsvp]` hierarchy level. This can be useful for troubleshooting. If you configure just the `mtu-signaling` statement, you can use the `show rsvp session detail` command to determine what the smallest MTU is on an LSP. The `show rsvp session detail` command displays the MTU value received and sent in the Adspec object.

Enabling Packet Fragmentation

To allow IP packets to be fragmented before they are encapsulated in MPLS, include the `allow-fragmentation` statement:

```
allow-fragmentation;
```

You can include this statement at the following hierarchy levels:

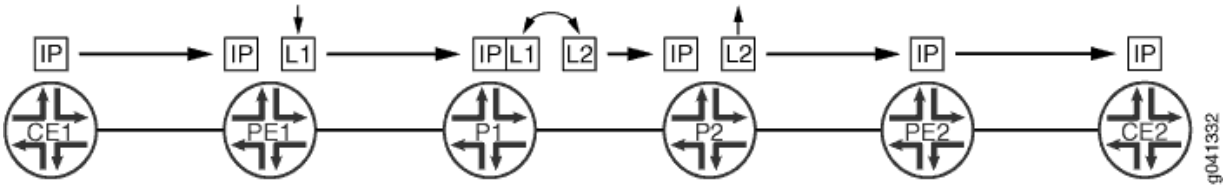
- `[edit protocols mpls path-mtu]`
- `[edit logical-systems` *`logical-system-name`* `protocols mpls path-mtu]`

> **NOTE**: Do not configure the `allow-fragmentation` statement alone. Always configure it in conjunction with the `mtu-signaling` statement.

Configuring Ultimate-Hop Popping for RSVP-signaled MPLS LSPs

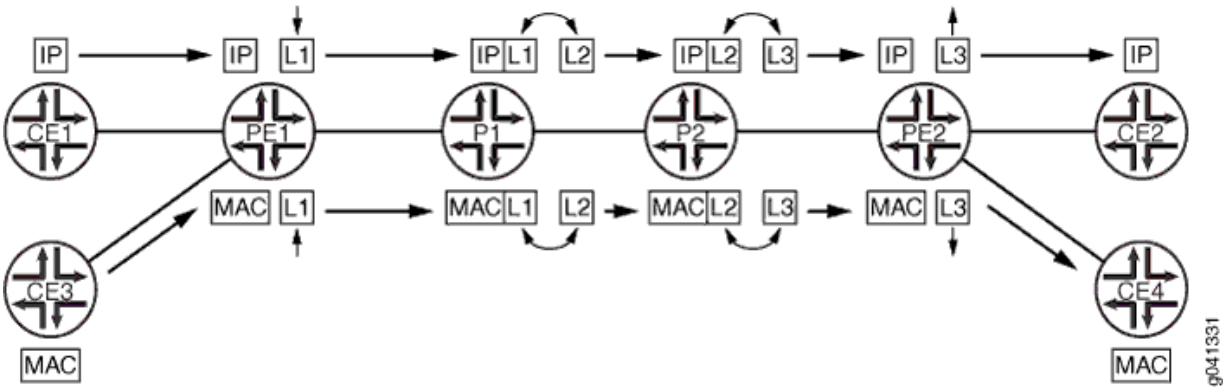
By default, RSVP-signaled LSPs use penultimate-hop popping (*PHP*). Figure 72 on page 1271 illustrates a penultimate-hop popping LSP between Router PE1 and Router PE2. Router CE1 forwards a packet to its next hop (Router PE1), which is also the LSP ingress. Router PE1 pushes label 1 on the packet and forwards the labeled packet to Router P1. Router P1 completes the standard MPLS label swapping operation, swapping label 1 for label 2, and forwards the packet to Router P2. Since Router P2 is the penultimate-hop router for the LSP to Router PE2, it first pops the label and then forwards the packet to Router PE2. When Router PE2 receives it, the packet can have a service label, an explicit-null label, or just be a plain IP or VPLS packet. Router PE2 forwards the unlabeled packet to Router CE2.

Figure 72: Penultimate-Hop Popping for an LSP



You can also configure ultimate-hop popping (*UHP*) (as shown in Figure 73 on page 1271) for RSVP-signaled LSPs. Some network applications can require that packets arrive at the egress router (Router PE2) with a non-null outer label. For an ultimate- hop popping LSP, the penultimate router (Router P2 in Figure 73 on page 1271) performs the standard MPLS label swapping operation (in this example, label 2 for label 3) before forwarding the packet to egress Router PE2. Router PE2 pops the outer label and performs a second lookup of the packet address to determine the end destination. It then forwards the packet to the appropriate destination (either Router CE2 or Router CE4).

Figure 73: Ultimate-Hop Popping for an LSP



The following network applications require that you configure UHP LSPs:

- MPLS-TP for performance monitoring and in-band *OAM*
- Edge protection virtual circuits

The following features do not support the UHP behavior:

- LDP-signaled LSPs
- Static LSPs
- Point-to-multipoint LSPs
- CCC
- `traceroute` command

For more information about UHP behavior, see Internet draft draft-ietf-mpls-rsvp-te-no-php-oob-mapping-01.txt, *Non PHP behavior and Out-of-Band Mapping for RSVP-TE LSPs*.

For point-to-point RSVP-signaled LSPs, UHP behavior is signaled from the LSP ingress. Based on the ingress router configuration, RSVP can signal the UHP LSP with the non-PHP flag set. RSVP PATH messages carry the two flags in the LSP-ATTRIBUTES object. When the egress router receives the PATH message, it assigns a non-null label to the LSP. RSVP also creates and installs two routes in the mpls.0 routing table. S refers to the S bit of the MPLS label, which indicates whether or not the bottom of the label stack has been reached.

- Route S=0—Indicates that there are more labels in the stack. The next hop for this route points to the mpls.0 routing table, triggering a chained MPLS label lookup to discover the remaining MPLS labels in the stack.

- Route S=1—Indicates that there are no more labels. The next hop points to the inet.0 routing table if the platform supports chained and multi-family lookup. Alternatively, the label route can point to a *VT* interface to initiate IP forwarding.

If you enable UHP LSPs, MPLS applications such as Layer 3 VPNs, VPLS, Layer 2 VPNs, and Layer 2 circuits can use the UHP LSPs. The following explains how UHP LSPs affect the different types of MPLS applications:

- Layer 2 VPNs and Layer 2 circuits—A packet arrives at the PE router (egress of the UHP LSP) with two labels. The outer label (S=0) is the UHP label, and the inner label (S=1) is the *VC* label. A lookup based on the transport label results in a table handle for the mpls.0 routing table. There is an additional route in the mpls.0 routing table corresponding to the inner label. A lookup based on the inner label results in the CE router next hop.

- Layer 3 VPN—A packet arrives at the PE router (egress of the UHP LSP) with two labels. The outer label (S=0) is the UHP label, and the inner label is the VPN label (S=1). A lookup based on the transport label results in the table handle for the mpls.0 routing table. There are two cases in this scenario. By default, Layer 3 VPNs advertise the per-next hop label. A lookup based on the inner label results in the next hop toward the CE router. However, if you have configured the `vrf-table-label` statement for the Layer 3 VPN routing instance, the inner *LSI* label points to the VRF routing table. An IP lookup is also completed for the VRF routing table.

  NOTE: UHP for Layer 3 VPNs configured with the `vrf-table-label` statement is supported on MX Series 5G Universal Routing Platforms only.

- VPLS—A packet arrives at the PE router (egress of the UHP LSP) with two labels. The outer label is the transport label (S=0) and the inner label is the VPLS label (S=1). A lookup based on the transport label results in the table handle for the mpls.0 routing table. A lookup based on the inner label in mpls.0 routing table results in the LSI tunnel interface of the VPLS routing instance if tunnel-services

is not configured (or a VT interface not available). MX 3D Series routers support chained lookup and multi-family lookup.

> **NOTE**: UHP for VPLS configured with the `no-tunnel-service` statement is supported on MX 3D Series routers only.

- IPv4 over MPLS—A packet arrives at the PE router (egress of the UHP LSP) with one label (S=1). A lookup based on this label returns a VT tunnel interface. Another IP lookup is completed on the VT interface to determine where to forward the packet. If the routing platform supports multi-family and chained lookups (for example, MX 3D routers and PTX Series Packet Transport Routers), lookup based on label route (S=1) points to the inet.0 routing table.

- IPv6 over MPLS—For IPv6 tunneling over MPLS, PE routers advertise IPv6 routes to each other with a label value of 2. This is the explicit null label for IPv6. As a result, the forwarding next hops for IPv6 routes that are learned from remote PE routers normally push two labels. The inner label is 2 (it could be different if the advertising PE router is from another vendor), and the router label is the LSP label. Packets arrive at the PE router (egress of the UHP LSP) with two labels. The outer label is the transport label (S=0), and the inner label is the IPv6 explicit-null label (label 2). Lookup based on the inner label in the mpls.0 routing table redirects back to the mpls.0 routing table. On MX 3D Series routers, the inner label (label 2) is stripped off and an IPv6 lookup is done using the inet6.0 routing table.

- Enabling both PHP and UHP LSPs—You can configure both PHP and UHP LSPs over the same network paths. You can separate PHP and UHP traffic by selecting forwarding LSP next hops using a regular expression with the `install-nexthop` statement. You can also separate traffic by simply naming the LSPs appropriately.

The following statements enable ultimate-hop popping for an LSP. You can enable this feature on a specific LSP or for all of the ingress LSPs configured on the router. Configure these statements on the router at the LSP ingress.

1. To enable ultimate-hop popping, include the `ultimate-hop-popping` statement:

   ```
   ultimate-hop-popping;
   ```

   Include this statement at the `[edit protocols mpls label-switched-path` *`label-switched-path-name`*`]` hierarchy level to enable ultimate-hop popping on a specific LSP. Include this statement at the `[edit protocols mpls]` hierarchy level to enable ultimate-hop popping on all of the ingress LSPs configured on the router. You can also configure the `ultimate-hop-popping` statement under the equivalent `[edit logical-routers]` hierarchy levels.

> **NOTE**: When you enable ultimate-hop popping, RSVP attempts to resignal existing LSPs as ultimate-hop popping LSPs in a make-before-break fashion. If an egress router does not support ultimate-hop popping, the existing LSP is torn down (RSVP sends a *PathTear message* along an LSP's path, removing the path state and dependent reservation state and releasing the associated networking resources).
>
> If you disable ultimate-hop popping, RSVP resignals existing LSPs as penultimate-hop popping LSPs in a make-before-break fashion.

2. If you want to enable both ultimate-hop-popping and chained next hops on MX 3D Series routers only, you also need to configure the `enhanced-ip` option for the `network-services` statement:

```
network-services enhanced-ip;
```

   You configure this statement at the `[edit chassis]` hierarchy level. Once you have configured the `network-services` statement, you need to reboot the router to enable UHP behavior.

Configuring RSVP to Pop the Label on the Ultimate-Hop Router

You can control the label value advertised on the egress router of an LSP. The default advertised label is label 3 (Implicit Null label). If label 3 is advertised, the penultimate-hop router removes the label and sends the packet to the egress router. When ultimate-hop popping is enabled, label 0 (IP version 4 [IPv4] Explicit Null label) is advertised. Ultimate-hop popping ensures that any packets traversing an MPLS network include a label.

> **NOTE**: Juniper Networks routers queue packets based on the incoming label. Routers from other vendors might queue packets differently. Keep this in mind when working with networks containing routers from multiple vendors.

To configure ultimate-hop popping for RSVP, include the `explicit-null` statement:

```
explicit-null;
```

You can include this statement at the following hierarchy levels:

- `[edit protocols mpls]`
- `[edit logical-systems` *`logical-system-name`* `protocols mpls]`

Enabling Ultimate-Hop Popping on Point-to-Multipoint LSPs

By default, for both point-to-point and point-to-multipoint LSPs, penultimate-hop popping is used for MPLS traffic. MPLS labels are removed from packets on the router just before the egress router of the LSP. The plain IP packets are then forwarded to the egress router. For ultimate-hop popping, the egress router is responsible for both removing the MPLS label and processing the plain IP packet.

It can be beneficial to enable ultimate-hop popping on point-to-multipoint LSPs, particularly when transit traffic is traversing the same egress device. If you enable ultimate-hop popping, a single copy of traffic can be sent over the incoming link, saving significant bandwidth. By default, ultimate-hop popping is disabled.

You enable ultimate-hop popping for point-to-multipoint LSPs by configuring the `tunnel-services` statement. When you enable ultimate-hop popping, the Junos OS selects one of the available virtual loopback tunnel (VT) interfaces to loop back the packets to the PFE for IP forwarding. By default, the VT interface selection process is performed automatically. Bandwidth admission control is used to limit the number of LSPs that can be used on one VT interface. Once all the bandwidth is consumed on one interface, the Junos OS selects another VT interface with sufficient bandwidth for admission control.

If an LSP requires more bandwidth than is available from any of the VT interfaces, ultimate-hop popping cannot be enabled and penultimate-hop popping is enabled instead.

For ultimate-hop popping on point-to-multipoint LSPs to function properly, the egress router must have a PIC that provides tunnel services, such as the tunnel services PIC or the adaptive services PIC. Tunnel services are needed for popping the final MPLS label and for returning packets for IP address lookups.

You can explicitly configure which VT interfaces handle the RSVP traffic by including the `devices` option for the `tunnel-services` statement. The `devices` option allows you to specify which VT interfaces are to be used by RSVP. If you do not configure this option, all of the VT interfaces available to the router can be used.

To enable ultimate-hop popping for the egress point-to-multipoint LSPs on a router, configure the `tunnel-services` statement:

```
tunnel-services {
    devices device-names;
}
```

You can configure this statement at the following hierarchy levels:

- `[edit protocols rsvp]`
- `[edit logical-systems logical-system-name protocols rsvp]`

To enable ultimate-hop popping for egress point-to-multipoint LSPs, you must also configure the `interface` statement with the `all` option:

```
interface all;
```

You must configure this statement at the `[edit protocols rsvp]` hierarchy level.

Tracing RSVP Protocol Traffic

IN THIS SECTION

- Examples: Tracing RSVP Protocol Traffic | **1277**

To trace RSVP protocol traffic, include the `traceoptions` statement:

```
traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
}
```

You can include this statement at the following hierarchy levels:

- `[edit protocols rsvp]`
- `[edit logical-systems logical-system-name protocols rsvp]`

You can specify the following RSVP-specific flags in the RSVP `traceoptions` statement:

Use the `file` statement to specify the name of the file that receives the output of the tracing operation. All files are placed in the directory `/var/log`. We recommend that you place RSVP tracing output in the file `rsvp-log`.

- `all`—All tracing operations.
- `error`—All detected error conditions
- `event`—RSVP-related events (helps to trace events related to RSVP graceful restart)
- `lmp`—RSVP-Link Management Protocol (LMP) interactions
- `packets`—All RSVP packets

- `path`—All path messages
- `pathtear`—PathTear messages
- `resv`—Resv messages
- `resvtear`—ResvTear messages
- `route`—Routing information
- `state`—Session state transitions, including when RSVP-signaled LSPs come up and go down.

NOTE: Use the `all` trace flag and the `detail` flag modifier with caution because these might cause the CPU to become very busy.

To view the log file generated when you enable RSVP traceoptions, issue the `show log` *`file-name`* command, where *`file-name`* is the file you specified using the `traceoptions` statement.

For general information about tracing and global tracing options, see the Junos OS Routing Protocols Library for Routing Devices.

Examples: Tracing RSVP Protocol Traffic

Trace RSVP path messages in detail:

```
[edit]
protocols {
    rsvp {
        traceoptions {
            file rsvp size 10m files 5;
            flag path;
        }
    }
}
```

Trace all RSVP messages:

```
[edit]
protocols {
    rsvp {
        traceoptions {
            file rsvp size 10m files 5;
```

```
            flag packets;
        }
    }
}
```

Trace all RSVP error conditions:

```
[edit]
protocols {
    rsvp {
        traceoptions {
            file rsvp size 10m files 5;
            flag error;
        }
    }
}
```

Trace RSVP state transitions:

```
[edit]
protocols {
    rsvp {
        traceoptions {
            file rsvp-data;
            flag state;
        }
    }
}
```

RSVP Log File Output

The following is sample output generated by issuing the `show log` *file-name* command on a router on which RSVP traceoptions have been enabled with the `state` flag configured. The RSVP-signaled LSP E-D is shown being torn down on Mar 11 14:04:36.707092. On Mar 11 14:05:30.101492, it is shown coming back up.

```
user@host> show log rsvp-data
Mar 11 13:58:51 trace_on: Tracing to "/var/log/E/rsvp-data" started
Mar 11 13:58:51.286413 rsvp_iflchange for vt ifl vt-1/2/0.69206016
Mar 11 13:58:51.286718 RSVP add interface vt-1/2/0.69206016, ifindex 101, ifaddr (null), family
1, is_appl_vt 0, already known
```

```
Mar 11 13:58:51.286818 RSVP Peer vt-1/2/0.69206016 TE-link __rpd:vt-1/2/0.69206016 Up
Mar 11 13:58:51.286978 RSVP add interface vt-1/2/0.69206016, ifindex 101, ifaddr (null), family
3, is_appl_vt 0, already known
Mar 11 13:58:51.287962 RSVP add interface lt-1/2/0.2, ifindex 113, ifaddr (null), family 2,
is_appl_vt 0, already known
Mar 11 13:58:51.288629 RSVP add interface lt-1/2/0.2, ifindex 113, ifaddr 10.0.0.2, family 1,
is_appl_vt 0, already known
Mar 11 13:58:51.288996 RSVP add interface lt-1/2/0.17, ifindex 114, ifaddr (null), family 2,
is_appl_vt 0, already known
Mar 11 13:58:51.289593 RSVP add interface lt-1/2/0.17, ifindex 114, ifaddr (null), family 3,
is_appl_vt 0, already known
Mar 11 13:58:51.289949 RSVP add interface lt-1/2/0.17, ifindex 114, ifaddr 10.0.0.17, family 1,
is_appl_vt 0, already known
Mar 11 13:58:51.290049 RSVP Peer lt-1/2/0.17 TE-link __rpd:lt-1/2/0.17 Up
Mar 11 13:59:05.042034 RSVP new bypass Bypass->10.0.0.18 on interface lt-1/2/0.17 to 10.0.0.18
avoid 0.0.0.0
Mar 11 14:04:36.707092 LSP "E-D" is Down (Reason: Reservation state deleted)
        Session: 192.168.0.4(port/tunnel ID 10321 Ext-ID 192.168.0.5) Proto 0 Sender:
192.168.0.5(port/lsp ID  1)
Mar 11 14:04:36.707661 RSVP delete resv state, session 192.168.0.4(port/tunnel ID 10321 Ext-ID
192.168.0.5) Proto 0
Mar 11 14:04:36.781185 RSVP delete path state, session 192.168.0.4(port/tunnel ID 10321 Ext-ID
192.168.0.5) Proto 0
Mar 11 14:04:36.781440 RSVP delete session 192.168.0.4(port/tunnel ID 10321 Ext-ID 192.168.0.5)
Proto 0
Mar 11 14:05:30.101492 RSVP new Session 192.168.0.4(port/tunnel ID 10321 Ext-ID 192.168.0.5)
Proto 0, session ID 3
Mar 11 14:05:30.101722 RSVP new path state, session 192.168.0.4(port/tunnel ID 10321 Ext-ID
192.168.0.5) Proto 0
Mar 11 14:05:30.179124 RSVP new resv state, session 192.168.0.4(port/tunnel ID 10321 Ext-ID
192.168.0.5) Proto 0
Mar 11 14:05:30.179395 RSVP PSB E-D, allocating psb resources for label 4294967295
Mar 11 14:05:30.180353 LSP "E-D" is Up
        Session: 192.168.0.4(port/tunnel ID 10321 Ext-ID 192.168.0.5) Proto 0 Sender:
192.168.0.5(port/lsp ID  2)
```

RSVP Graceful Restart

RSVP graceful restart allows a router undergoing a restart to inform its adjacent neighbors of its condition. The restarting router requests a grace period from the neighbor or peer, which can then cooperate with the restarting router. The restarting router can still forward MPLS traffic during the restart period; convergence in the network is not disrupted. The restart is not visible to the rest of the network, and the restarting router is not removed from the network topology. RSVP graceful restart can

be enabled on both transit routers and ingress routers. It is available for both point-to-point LSPs and point-to-multipoint LSPs.

RSVP graceful restart is described in the following sections:

- RSVP Graceful Restart Standard
- "RSVP Graceful Restart Terminology" on page 1280
- "RSVP Graceful Restart Operation" on page 1281
- "Processing the Restart Cap Object" on page 1282

RSVP Graceful Restart Terminology

IN THIS SECTION

- Restart time (in milliseconds) | **1280**
- Recovery time (in milliseconds) | **1280**

Restart time (in milliseconds)

The default value is 60,000 milliseconds (1 minute). The restart time is advertised in the hello message. The time indicates how long a neighbor should wait to receive a hello message from a restarting router before declaring that router dead and purging states.

The Junos OS can override a neighbor's advertised restart time if the time is greater than one-third the local restart time. For example, given the default restart time of 60 seconds, a router would wait 20 seconds or less to receive a hello message from a restarting neighbor. If the restart time is zero, the restarting neighbor can immediately be declared dead.

Recovery time (in milliseconds)

Applies only when the control channel is up (the hello exchange is complete) before the restart time. Applies only to nodal faults.

When a graceful restart is in progress, the time left to complete a recovery is advertised. At other times, this value is zero. The maximum advertised recovery time is 2 minutes (120,000 milliseconds).

During the recovery time, a restarting node attempts to recover its lost states with assistance from its neighbors. The neighbor of the restarting node must send the path messages with the recovery labels to the restarting node within a period of one-half the recovery time. The restarting node considers its graceful restart complete after its advertised recovery time.

RSVP Graceful Restart Operation

For RSVP graceful restart to function, the feature must be enabled on the global routing instance. RSVP graceful restart can be disabled at the protocol level (for RSVP alone) or at the global level for all protocols.

RSVP graceful restart requires the following of a restarting router and the router's neighbors:

- For the restarting router, RSVP graceful restart attempts to maintain the routes installed by RSVP and the allocated labels, so that traffic continues to be forwarded without disruption. RSVP graceful restart is done quickly enough to reduce or eliminate the impact on neighboring nodes.
- The neighboring routers must have RSVP graceful restart helper mode enabled, thus allowing them to assist a router attempting to restart RSVP.

An object called Restart Cap that is sent in RSVP hello messages advertises a node's restart capability. The neighboring node sends a Recover Label object to the restarting node to recover its forwarding state. This object is essentially the old label that the restarting node advertised before the node went down.

The following lists the RSVP graceful restart behaviors, which vary depending on the configuration and on which features are enabled:

- If you disable helper mode, the Junos OS does not attempt to help a neighbor restart RSVP. Any information that arrives with a Restart Cap object from a neighbor is ignored.
- When you enable graceful restart under the routing instance configuration, the router can restart gracefully with the help of its neighbors. RSVP advertises a Restart Cap object (RSVP RESTART) in hello messages in which restart and recovery times are specified (neither value is 0).
- If you explicitly disable RSVP graceful restart under the `[protocols rsvp]` hierarchy level, the Restart Cap object is advertised with restart and recovery times specified as 0. The restart of neighboring routers is supported (unless helper mode is disabled), but the router itself does not preserve the RSVP forwarding state and cannot recover its control state.
- If after a restart RSVP realizes that no forwarding state has been preserved, the Restart Cap object is advertised with restart and recovery times specified as 0.
- If graceful restart and helper mode are disabled, RSVP graceful restart is completely disabled. The router neither recognizes nor advertises the RSVP graceful restart objects.

You cannot explicitly configure values for the restart and recovery times.

Unlike other protocols, there is no way for RSVP to determine that it has completed a restart procedure, other than a fixed timeout. All RSVP graceful restart procedures are timer-based. A `show rsvp version` command might indicate that the restart is still in progress even if all RSVP sessions are back up and the routes are restored.

Processing the Restart Cap Object

The following assumptions are made about a neighbor based on the Restart Cap object (assuming that a control channel failure can be distinguished unambiguously from a node restart):

- A neighbor that does not advertise the Restart Cap object in its hello messages cannot assist a router with state or label recovery, nor can it perform an RSVP graceful restart.
- After a restart, a neighbor advertising a Restart Cap object with a restart time equal to any value and a recovery time equal to 0 has not preserved its forwarding state. When a recovery time equals 0, the neighbor is considered dead and any states related to this neighbor are purged, regardless of the value of the restart time.
- After a restart, a neighbor advertising its recovery time with a value other than 0 can keep or has kept the forwarding state. If the local router is helping its neighbor with restart or recovery procedures, it sends a Recover Label object to this neighbor.

Configuring RSVP Graceful Restart

IN THIS SECTION

- Enabling Graceful Restart for All Routing Protocols | **1283**
- Disabling Graceful Restart for RSVP | **1283**
- Disabling RSVP Helper Mode | **1283**
- Configuring the Maximum Helper Recovery Time | **1284**
- Configuring the Maximum Helper Restart Time | **1284**

The following RSVP graceful restart configurations are possible:

- Graceful restart and helper mode are both enabled (the default).
- Graceful restart is enabled but helper mode is disabled. A router configured in this way can restart gracefully, but cannot help a neighbor with its restart and recovery procedures.
- Graceful restart is disabled but helper mode is enabled. A router configured in this way cannot restart gracefully, but can help a restarting neighbor.
- Graceful restart and helper mode both are disabled. This configuration completely disables RSVP graceful restart (including restart and recovery procedures and helper mode). The router behaves like a router that does not support RSVP graceful restart.

NOTE: In order to turn on RSVP graceful restart, you must set the global graceful restart timer to at least 180 seconds.

The following sections describe how to configure RSVP graceful restart:

Enabling Graceful Restart for All Routing Protocols

To enable graceful restart for RSVP, you need to enable graceful restart for all the protocols that support graceful restart on the router. For more information about graceful restart, see the Junos OS Routing Protocols Library for Routing Devices.

To enable graceful restart on the router, include the `graceful-restart` statement:

```
graceful-restart;
```

You can include this statement at the following hierarchy levels:

- `[edit routing-options]`
- `[edit logical-systems` *`logical-system-name`* `routing-options]`

Disabling Graceful Restart for RSVP

By default, RSVP graceful restart and RSVP helper mode are enabled when you enable graceful restart. However, you can disable one or both of these capabilities.

To disable RSVP graceful restart and recovery, include the `disable` statement at the `[edit protocols rsvp graceful-restart]` hierarchy level:

```
disable;
```

Disabling RSVP Helper Mode

To disable RSVP helper mode, include the `helper-disable` statement at the `[edit protocols rsvp graceful-restart]` hierarchy level:

```
helper-disable;
```

Configuring the Maximum Helper Recovery Time

To configure the amount of time the router retains the state of its RSVP neighbors while they undergo a graceful restart, include the `maximum-helper-recovery-time` statement at the `[edit protocols rsvp graceful-restart]` hierarchy level. This value is applied to all neighboring routers, so it should be based on the time required by the slowest RSVP neighbor to recover.

```
maximum-helper-recovery-time seconds;
```

Configuring the Maximum Helper Restart Time

To configure the delay between when the router discovers that a neighboring router has gone down and when it declares the neighbor down, include the `maximum-helper-restart-time` statement at the `[edit protocols rsvp graceful-restart]` hierarchy level. This value is applied to all neighboring routers, so it should be based on the time required by the slowest RSVP neighbor to restart.

```
maximum-helper-restart-time seconds;
```

RSVP LSP Tunnels Overview

A Resource Reservation Protocol (RSVP) label-switched path (LSP) tunnel enables you to send RSVP LSPs inside other RSVP LSPs. This enables a network administrator to provide traffic engineering from one end of the network to the other. A useful application for this feature is to connect customer edge (CE) routers with provider edge (PE) routers by using an RSVP LSP, and then tunnel this edge LSP inside a second RSVP LSP traveling across the network core.

You should have a general understanding of MPLS and label switching concepts. For more information about MPLS, see the *Junos MPLS Applications Configuration Guide*.

An RSVP LSP tunnel adds the concept of a forwarding adjacency, similar to the one used for generalized Multiprotocol Label Switching (GMPLS). (For more information about GMPLS, see *Junos GMPLS User Guide*.

The forwarding adjacency creates a tunneled path for sending data between peer devices in an RSVP LSP network. Once a forwarding adjacency LSP (FA-LSP) has been established, other LSPs can be sent over the FA-LSP by using Constrained Shortest Path First (CSPF), Link Management Protocol (LMP), Open Shortest Path First (OSPF), and RSVP.

To enable an RSVP LSP tunnel, the Junos OS uses the following mechanisms:

- LMP—Originally designed for GMPLS, LMP establishes forwarding adjacencies between RSVP LSP tunnel peers, and maintains and allocates resources for traffic engineering links.

- OSPF extensions—OSPF was designed to route packets to physical and logical interfaces related to a *Physical Interface Card* (PIC). This protocol has been extended to route packets to virtual peer interfaces defined in an LMP configuration.
- RSVP-TE extensions—RSVP-TE was designed to signal the setup of packet LSPs to physical interfaces. The protocol has been extended to request path setup for packet LSPs traveling to virtual peer interfaces defined in an LMP configuration.

> **NOTE**: Beginning with Junos OS Release 15.1, multi-instance support is extended to MPLS RSVP-TE. This support is available only for virtual router instance type. A router can create and participate in multiple independent TE topology partitions, which allows each partitioned TE domain to scale independently. Multi-instance RSVP-TE provides the flexibility to hand pick the control-plane entities that need to be instance-aware, for example, a router can participate in multiple TE instances while still running a single BGP instance.

The Junos OS implementation of MPLS RSVP-TE is scaled to enhance the usability, visibility, configuration, and troubleshooting of LSPs in Junos OS Release 16.1.

These enhancements make the RSVP-TE configuration easier at scale by:

- Ensuring the LSP data-plane readiness during LSP resignaling before traffic traverses the LSP with the RSVP-TE LSP self-ping mechanism.

  An LSP should not start to carry traffic unless it is known to have been programmed in the data plane. Data exchange in the LSP data plane, such as LSP ping requests, happens at the ingress router before switching traffic to an LSP, or to its MBB instance. In large networks, this traffic can overwhelm an LSP egress router, as the egress LSP needs to respond to the LSP ping requests. The LSP self-ping mechanism enables the ingress LER to create LSP ping response messages and send them over the LSP data plane. On receiving these messages, the egress LER forwards them to the ingress, indicating the liveliness of the LSP data plane. This ensures that the LSP does not start to carry traffic before the data plane has been programmed.

- Removing the current hard limit of 64K LSPs on an ingress router and scaling the total number of LSPs with RSVP-TE signaled LSPs. There can be up to 64K LSPs configured on a per-egress basis. Earlier, this limit was the aggregate number of LSPs that could be configured on the ingress LER.
- Preventing abrupt tearing down of LSPs by the ingress router because of delay in signaling the LSP at the transit routers.
- Enabling a flexible view of LSP data-sets to facilitate LSP characteristic data visualization.

NOTE: Starting with Junos OS Release 17.4, a default timer of 1800 seconds for self-ping is introduced.

The following limitations exist for LSP hierarchies:

- Circuit cross-connect (CCC)-based LSPs are not supported.
- Graceful restart is not supported.
- Link protection is not available for FA-LSPs or at the egress point of the forwarding adjacency.
- Point-to-multipoint LSPs are not supported across FA-LSPs.

Example: RSVP LSP Tunnel Configuration

IN THIS SECTION

- Verifying Your Work | 1303

Figure 74: RSVP LSP Tunnel Topology Diagram

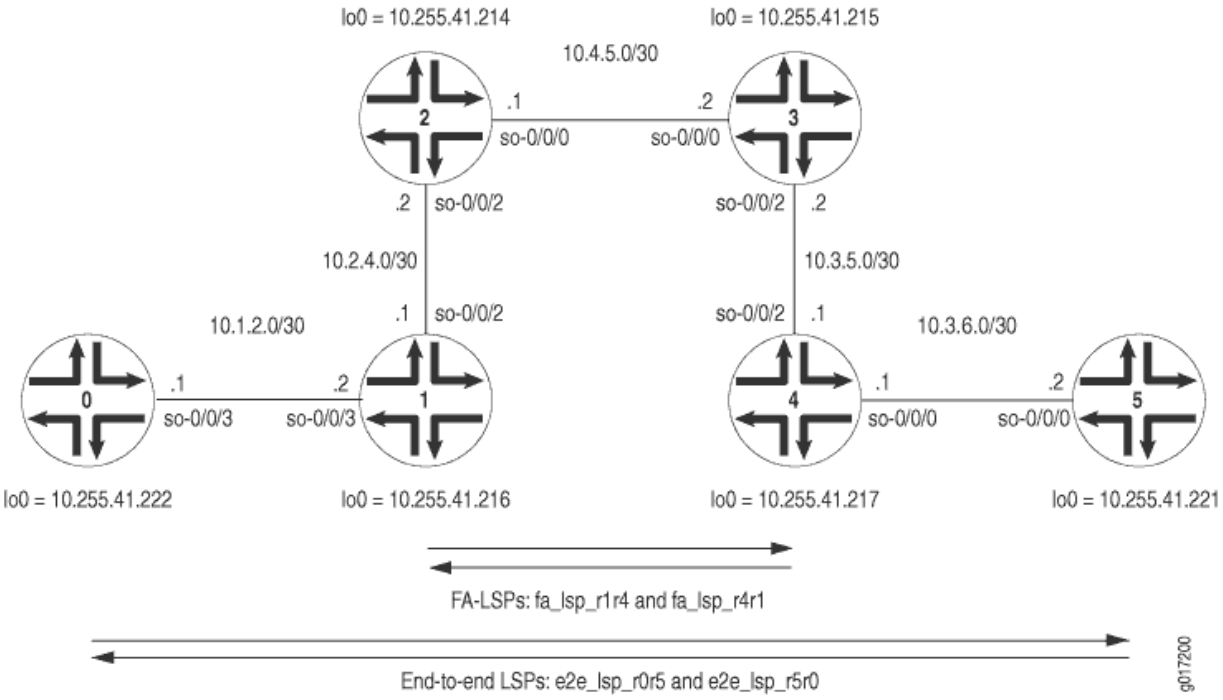


Figure 74 on page 1287 shows an end-to-end RSVP LSP called e2e_lsp_r0r5 that originates on Router 0 and terminates on Router 5. In transit, this LSP traverses the FA-LSP fa_lsp_r1r4. The return path is represented by the end-to-end RSVP LSP e2e_lsp_r5r0 that travels over the FA-LSP fa_lsp_r4r1.

On Router 0, configure the end-to-end RSVP LSP that travels to Router 5. Use a strict path that traverses Router 1 and the LMP traffic engineering link traveling from Router 1 to Router 4.

Router 0

```
[edit]
interfaces {
    so-0/0/3 {
        unit 0 {
            family inet {
                address 10.1.2.1/30;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.41.222/32;
            }
            family mpls;
        }
    }
}
routing-options {
    forwarding-table {
        export pplb;
    }
}
protocols {
    rsvp {
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
    mpls {
        admin-groups {
            fa 1;
```

```
            backup 2;
            other 3;
        }
                        label-switched-path e2e_lsp_r0r5 { # An end-to-end LSP traveling to
Router 5.
            to 10.255.41.221;
            bandwidth 30k;
                            primary path-fa; # Reference the requested path here.
        }
                        path path-fa { # Configure the strict path here.
            10.1.2.2 strict;
                            172.16.30.2 strict; # This traverses the TE link heading to
Router 4.
        }
        interface all;
        interface fxp0.0 {
            disable;
        }
        interface so-3/2/1.0 {
             admin-group other;
        }
        interface so-0/0/3.0 {
            admin-group other;
        }
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface fxp0.0 {
                disable;
            }
            interface all;
        }
    }
}
policy-options {
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
    }
}
```

On Router 1, configure an FA-LSP to reach Router 4. Establish an LMP traffic engineering link and LMP peer relationship with Router 4. Reference the FA-LSP in the traffic engineering link and add the peer interface into both OSPF and RSVP.

When the return path end-to-end LSP arrives at Router 1, the routing platform performs a routing lookup and can forward traffic to Router 0. Make sure you configure OSPF correctly between Routers 0 and 1.

Router 1

```
[edit]
interfaces {
    so-0/0/1 {
        unit 0 {
            family inet {
                address 10.2.3.1/30;
            }
            family mpls;
        }
    }
    so-0/0/2 {
        unit 0 {
            family inet {
                address 10.2.4.1/30;
            }
            family mpls;
        }
    }
    so-0/0/3 {
        unit 0 {
            family inet {
                address 10.1.2.2/30;
            }
            family mpls;
        }
    }
    fe-0/1/2 {
        unit 0 {
            family inet {
                address 10.2.5.1/30;
            }
            family mpls;
        }
```

```
    }
    at-1/0/0 {
        atm-options {
            vpi 1;
        }
         unit 0 {
            vci 1.100;
            family inet {
                address 10.2.3.5/30;
            }
            family mpls;
        }
    }
}
routing-options {
    forwarding-table {
        export [ pplb choose_lsp ];
    }
}
protocols {
    rsvp {
        interface all;
        interface fxp0.0 {
            disable;
        }
                        peer-interface r4; # Apply the LMP peer interface here.
    }
    mpls {
        admin-groups {
            fa 1;
            backup 2;
            other 3;
        }
                          label-switched-path fa_lsp_r1r4 { # Configure your FA-LSP to Router 4
here.
            to 10.255.41.217;
            bandwidth 400k;
                               primary path_r1r4; # Apply the FA-LSP path here.
        }
                          path path_r1r4 { # Configure the FA-LSP path here.
            10.2.4.2;
            10.4.5.2;
            10.3.5.1;
```

```
        }
        interface so-0/0/3.0 {
            admin-group other;
        }
        interface so-0/0/1.0 {
            admin-group fa;
        }
        interface at-1/0/0.0 {
            admin-group backup;
        }
        interface fe-0/1/2.0 {
            admin-group backup;
        }
        interface so-0/0/2.0 {
            admin-group fa;
        }
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface fxp0.0 {
                disable;
            }
            interface all;
                                peer-interface r4; # Apply the LMP peer interface here.
        }
    }
                    link-management { # Configure LMP statements here.
        te-link link_r1r4 { # Assign a name to the TE link here.
            local-address 172.16.30.1; # Configure a local address for the TE link.
            remote-address 172.16.30.2; # Configure a remote address for the TE link.
            te-metric 1; # Manually set a metric here if you are not relying on CSPF.
            label-switched-path fa_lsp_r1r4; # Reference the FA-LSP here.
        }
        peer r4 { # Configure LMP peers here.
            address 10.255.41.217; # Configure the loopback address of your peer here.
            te-link link_r1r4; # Apply the LMP TE link here.
        }
    }
}
policy-options {
    policy-statement choose_lsp {
        term A {
```

```
            from community choose_e2e_lsp;
            then {
                install-nexthop strict lsp e2e_lsp_r1r4;
                accept;
            }
        }
        term B {
            from community choose_fa_lsp;
            then {
                install-nexthop strict lsp fa_lsp_r1r4;
                accept;
            }
        }
    }
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
    }
    community choose_e2e_lsp members 1000:1000;
    community choose_fa_lsp members 2000:2000;
    community set_e2e_lsp members 1000:1000;
    community set_fa_lsp members 2000:2000;
}
```

On Router 2, configure OSPF, MPLS, and RSVP on all interfaces that transport the FA-LSPs across the core network.

Router 2

```
[edit]
interfaces {
    so-0/0/0 {
        unit 0 {
            family inet {
                address 10.4.5.1/30;
            }
            family mpls;
        }
    }
    so-0/0/1 {
        unit 0 {
```

```
            family inet {
                address 10.1.4.2/30;
            }
            family mpls;
        }
    }
    so-0/0/2 {
        unit 0 {
            family inet {
                address 10.2.4.2/30;
            }
            family mpls;
        }
    }
    fe-0/1/2 {
        unit 0 {
            family inet {
                address 10.3.4.2/30;
            }
            family mpls;
        }
    }
}
routing-options {
    forwarding-table {
        export pplb;
    }
}
              protocols { # OSPF, MPLS, and RSVP form the core backbone for the FA-LSPs.
                    rsvp {
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
                    mpls {
        admin-groups {
            fa 1;
            backup 2;
            other 3;
        }
        path path_r1 {
            10.2.4.1;
```

```
        }
        path path_r3r4 {
            10.4.5.2;
            10.3.5.1;
        }
        interface all;
        interface fxp0.0 {
            disable;
        }
        interface so-0/0/1.0 {
            admin-group other;
        }
        interface fe-0/1/2.0 {
            admin-group backup;
        }
        interface so-0/0/2.0 {
            admin-group fa;
        }
        interface so-0/0/0.0 {
            admin-group fa;
        }
    }
                    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface fxp0.0 {
                disable;
            }
            interface all;
        }
    }
}
policy-options {
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
    }
}
```

On Router 3, configure OSPF, MPLS, and RSVP on all interfaces that transport the FA-LSPs across the core network.

Router 3

```
[edit]
interfaces {
    so-0/0/0 {
        unit 0 {
            family inet {
                address 10.4.5.2/30;
            }
            family mpls;
        }
    }
    so-0/0/1 {
        unit 0 {
            family inet {
                address 10.5.6.1/30;
            }
            family mpls;
        }
    }
    so-0/0/2 {
        unit 0 {
            family inet {
                address 10.3.5.2/30;
            }
            family mpls;
        }
    }
    fe-0/1/2 {
        unit 0 {
            family inet {
                address 10.2.5.2/30;
            }
            family mpls;
        }
    }
}
routing-options {
    forwarding-table {
        export pplb;
    }
}
```

```
protocols { # OSPF, MPLS, and RSVP form the core backbone for the FA-LSPs.
    rsvp {
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
                    mpls {
        admin-groups {
            fa 1;
            backup 2;
            other 3;
        }
        path path_r4 {
            10.3.5.1;
        }
        path path_r2r1 {
            10.4.5.1;
            10.2.4.1;
        }
        interface all;
        interface fxp0.0 {
            disable;
        }
        interface so-0/0/2.0 {
            admin-group fa;
        }
        interface fe-0/1/2.0 {
            admin-group backup;
        }
        interface so-0/0/1.0 {
            admin-group other;
        }
        interface so-0/0/0.0 {
             admin-group fa;
        }
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface fxp0.0 {
                disable;
            }
```

```
            interface all;
        }
    }
}
policy-options {
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
    }
}
```

On Router 4, configure a return path FA-LSP to reach Router 1. Establish an LMP traffic engineering link and LMP peer relationship with Router 1. Reference the FA-LSP in the traffic engineering link and add the peer interface into both OSPF and RSVP.

When the initial end-to-end LSP arrives at Router 4, the routing platform performs a routing lookup and can forward traffic to Router 5. Make sure you configure OSPF correctly between Router 4 and Router 5.

Router 4

```
[edit]
interfaces {
    so-0/0/0 {
        unit 0 {
            family inet {
                address 10.3.6.1/30;
            }
            family mpls;
        }
    }
    so-0/0/1 {
        unit 0 {
            family inet {
                address 10.2.3.2/30;
            }
            family mpls;
        }
    }
    so-0/0/2 {
        unit 0 {
            family inet {
```

```
                address 10.3.5.1/30;
            }
            family mpls;
        }
    }
    fe-0/1/2 {
        unit 0 {
            family inet {
                address 10.3.4.1/30;
            }
            family mpls;
        }
    }
    at-1/0/0 {
        atm-options {
            vpi 1;
        }
        unit 0 {
            vci 1.100;
            family inet {
                address 10.2.3.6/30;
            }
            family mpls;
        }
    }
}
routing-options {
    forwarding-table {
        export [ pplb choose_lsp ];
    }
}
protocols {
    rsvp {
        interface all;
        interface fxp0.0 {
            disable;
        }
                        peer-interface r1; # Apply the LMP peer interface here.
    }
    mpls {
        admin-groups {
            fa 1;
            backup 2;
```

```
            other 3;
        }
                        label-switched-path fa_lsp_r4r1 { # Configure your FA-LSP here.
            to 10.255.41.216;
             bandwidth 400k;
                            primary path_r4r1; # Apply the FA-LSP path here.
        }
                        path path_r4r1 { # Configure the FA-LSP path here.
            10.3.5.2;
            10.4.5.1;
            10.2.4.1;
        }
        interface all;
        interface fxp0.0 {
            disable;
        }
        interface at-1/0/0.0 {
            admin-group backup;
        }
        interface so-0/0/2.0 {
            admin-group fa;
        }
        interface fe-0/1/2.0 {
            admin-group backup;
        }
        interface so-0/0/0.0 {
            admin-group other;
        }
        interface so-0/0/1.0 {
            admin-group fa;
        }
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface fxp0.0 {
                disable;
            }
            interface all;
                            peer-interface r1; # Apply the LMP peer interface here.
        }
    }
                    link-management { # Configure LMP statements here.
```

```
        te-link link_r4r1 { # Assign a name to the TE link here.
                            local-address 172.16.30.2; # Configure a local address for the TE
link.
            remote-address 172.16.30.1; # Configure a remote address for the TE link.
                            te-metric 1; # Manually set a metric here if you are not relying
on CSPF.
            label-switched-path fa_lsp_r4r1; # Reference the FA-LSP here.
        }
                        peer r1 { # Configure LMP peers here.
            address 10.255.41.216; # Configure the loopback address of your peer here.
                            te-link link_r4r1; # Apply the LMP TE link here.
        }
    }
}
policy-options {
    policy-statement choose_lsp {
        term A {
            from community choose_e2e_lsp;
            then {
                install-nexthop strict lsp e2e_lsp_r4r1;
                accept;
            }
        }
        term B {
            from community choose_fa_lsp;
            then {
                install-nexthop strict lsp fa_lsp_r4r1;
                accept;
            }
        }
    }
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
    }
    community choose_e2e_lsp members 1000:1000;
    community choose_fa_lsp members 2000:2000;
    community set_e2e_lsp members 1000:1000;
    community set_fa_lsp members 2000:2000;
}
```

On Router 5, configure the return path end-to-end RSVP LSP that travels to Router 0. Use a strict path that traverses Router 4 and the LMP traffic engineering link traveling from Router 4 to Router 1.

Router 5

```
[edit]
interfaces {
    so-0/0/2 {
        unit 0 {
            family inet {
                address 10.3.6.2/30;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.41.221/32;
            }
        }
    }
}
routing-options {
    forwarding-table {
        export pplb;
    }
}
protocols {
    rsvp {
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
    mpls {
        admin-groups {
            fa 1;
            backup 2;
            other 3;
        }
                        label-switched-path e2e_lsp_r5r0 { # An end-to-end LSP returning to
```

```
Router 0.
            to 10.255.41.222;
            bandwidth 30k;
                                primary path-fa; # Reference the requested path here.
        }
                            path path-fa { # Configure the strict path here.
            10.3.6.1 strict;
                                172.16.30.1 strict; # This traverses the TE link heading to
Router 1.
        }
        interface all;
        interface fxp0.0 {
            disable;
        }
        interface so-0/0/2.0 {
            admin-group other;
        }
        interface so-0/0/1.0 {
            admin-group other;
        }
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface fxp0.0 {
                disable;
            }
            interface all;
        }
    }
}
policy-options {
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
    }
}
```

Verifying Your Work

To verify that your RSVP LSP tunnel is working correctly, issue the following commands:

- show ted database (extensive)
- show rsvp session name (extensive)
- show link-management
- show link-management te-link name (detail)

To see these commands used with the configuration example, see the following sections:

Router 0

On Router 0, you can verify that the FA-LSPs appear as valid paths in the traffic engineering database. In this case, look for the paths from Router 1 (10.255.41.216) and Router 4 (10.255.41.217) that reference the LMP traffic engineering link addresses of 172.16.30.1 and 172.16.30.2. You can also issue the show rsvp session extensive command to look for the path of the end-to-end LSP as it travels to Router 5 over the FA-LSP.

```
user@router0> show ted database
TED database: 0 ISIS nodes 8 INET nodes
ID                          Type Age(s) LnkIn LnkOut Protocol
10.255.41.214               Rtr     486     4      4 OSPF(0.0.0.0)
    To: 10.255.41.222, Local: 10.1.4.2, Remote: 10.1.4.1
    To: 10.255.41.216, Local: 10.2.4.2, Remote: 10.2.4.1
    To: 10.255.41.215, Local: 10.4.5.1, Remote: 10.4.5.2
    To: 10.3.4.1-1, Local: 10.3.4.2, Remote: 0.0.0.0
ID                          Type Age(s) LnkIn LnkOut Protocol
10.255.41.215               Rtr     187     4      4 OSPF(0.0.0.0)
    To: 10.255.41.214, Local: 10.4.5.2, Remote: 10.4.5.1
    To: 10.255.41.217, Local: 10.3.5.2, Remote: 10.3.5.1
    To: 10.255.41.221, Local: 10.5.6.1, Remote: 10.5.6.2
    To: 10.2.5.1-1, Local: 10.2.5.2, Remote: 0.0.0.0
ID                          Type Age(s) LnkIn LnkOut Protocol
10.255.41.216                Rtr     396     6      6 OSPF(0.0.0.0)
    To: 10.255.41.222, Local: 10.1.2.2, Remote: 10.1.2.1
    To: 10.255.41.214, Local: 10.2.4.1, Remote: 10.2.4.2
    To: 10.255.41.217, Local: 10.2.3.1, Remote: 10.2.3.2
     To: 10.255.41.217, Local: 172.16.30.1, Remote: 172.16.30.2
    To: 10.255.41.217, Local: 10.2.3.5, Remote: 10.2.3.6
    To: 10.2.5.1-1, Local: 10.2.5.1, Remote: 0.0.0.0
ID                          Type Age(s) LnkIn LnkOut Protocol
10.255.41.217                Rtr     404     6      6 OSPF(0.0.0.0)
    To: 10.255.41.216, Local: 10.2.3.2, Remote: 10.2.3.1
```

```
     To: 10.255.41.216, Local: 172.16.30.2, Remote: 172.16.30.1
    To: 10.255.41.216, Local: 10.2.3.6, Remote: 10.2.3.5
    To: 10.255.41.215, Local: 10.3.5.1, Remote: 10.3.5.2
    To: 10.255.41.221, Local: 10.3.6.1, Remote: 10.3.6.2
    To: 10.3.4.1-1, Local: 10.3.4.1, Remote: 0.0.0.0
ID                          Type Age(s) LnkIn LnkOut Protocol
10.255.41.221               Rtr     481     2      2 OSPF(0.0.0.0)
    To: 10.255.41.215, Local: 10.5.6.2, Remote: 10.5.6.1
    To: 10.255.41.217, Local: 10.3.6.2, Remote: 10.3.6.1
ID                          Type Age(s) LnkIn LnkOut Protocol
10.255.41.222               Rtr    2883     2      2 OSPF(0.0.0.0)
    To: 10.255.41.216, Local: 10.1.2.1, Remote: 10.1.2.2
    To: 10.255.41.214, Local: 10.1.4.1, Remote: 10.1.4.2

user@router0>  show ted database 10.255.41.216 extensive
TED database: 0 ISIS nodes 8 INET nodes
NodeID: 10.255.41.216
  Type: Rtr, Age: 421 secs, LinkIn: 6, LinkOut: 6
  Protocol: OSPF(0.0.0.0)
    To: 10.255.41.222, Local: 10.1.2.2, Remote: 10.1.2.1
      Color: 0x8 other
      Metric: 1
      Static BW: 155.52Mbps
      Reservable BW: 155.52Mbps
      Available BW [priority] bps:
          [0] 155.4Mbps    [1] 155.4Mbps   [2] 155.4Mbps   [3] 155.4Mbps
          [4] 155.4Mbps    [5] 155.4Mbps   [6] 155.4Mbps   [7] 155.4Mbps
      Interface Switching Capability Descriptor(1):
        Switching type: Packet
        Encoding type: Packet
        Maximum LSP BW [priority] bps:
          [0] 155.4Mbps    [1] 155.4Mbps   [2] 155.4Mbps   [3] 155.4Mbps
          [4] 155.4Mbps    [5] 155.4Mbps   [6] 155.4Mbps   [7] 155.4Mbps
    To: 10.255.41.214, Local: 10.2.4.1, Remote: 10.2.4.2
      Color: 0x2 fa
      Metric: 1
      Static BW: 155.52Mbps
      Reservable BW: 155.52Mbps
      Available BW [priority] bps:
          [0] 155.12Mbps   [1] 155.12Mbps  [2] 155.12Mbps  [3] 155.12Mbps
          [4] 155.12Mbps   [5] 155.12Mbps  [6] 155.12Mbps  [7] 155.12Mbps
      Interface Switching Capability Descriptor(1):
        Switching type: Packet
```

```
      Encoding type: Packet
      Maximum LSP BW [priority] bps:
        [0] 155.12Mbps   [1] 155.12Mbps  [2] 155.12Mbps  [3] 155.12Mbps
        [4] 155.12Mbps   [5] 155.12Mbps  [6] 155.12Mbps  [7] 155.12Mbps
  To: 10.255.41.217, Local: 10.2.3.1, Remote: 10.2.3.2
    Color: 0x2 fa
    Metric: 1
    Static BW: 155.52Mbps
    Reservable BW: 155.52Mbps
    Available BW [priority] bps:
        [0] 155.52Mbps   [1] 155.52Mbps  [2] 155.52Mbps  [3] 155.52Mbps
        [4] 155.52Mbps   [5] 155.52Mbps  [6] 155.52Mbps  [7] 155.52Mbps
    Interface Switching Capability Descriptor(1):
      Switching type: Packet
      Encoding type: Packet
      Maximum LSP BW [priority] bps:
        [0] 155.52Mbps   [1] 155.52Mbps  [2] 155.52Mbps  [3] 155.52Mbps
        [4] 155.52Mbps   [5] 155.52Mbps  [6] 155.52Mbps  [7] 155.52Mbps
  To: 10.255.41.217,  Local: 172.16.30.1, Remote: 172.16.30.2
    Metric: 1
    Static BW: 400kbps
    Reservable BW: 400kbps
    Available BW [priority] bps:
        [0] 370kbps      [1] 370kbps     [2] 370kbps     [3] 370kbps
        [4] 370kbps      [5] 370kbps     [6] 370kbps     [7] 370kbps
    Interface Switching Capability Descriptor(1):
      Switching type: Packet
      Encoding type: Packet
      Maximum LSP BW [priority] bps:
        [0] 370kbps      [1] 370kbps     [2] 370kbps     [3] 370kbps
        [4] 370kbps      [5] 370kbps     [6] 370kbps     [7] 370kbps
  To: 10.255.41.217, Local: 10.2.3.5, Remote: 10.2.3.6
    Color: 0x4 backup
    Metric: 1
    Static BW: 155.52Mbps
    Reservable BW: 155.52Mbps
    Available BW [priority] bps:
        [0] 155.52Mbps   [1] 155.52Mbps  [2] 155.52Mbps  [3] 155.52Mbps
        [4] 155.52Mbps   [5] 155.52Mbps  [6] 155.52Mbps  [7] 155.52Mbps
    Interface Switching Capability Descriptor(1):
      Switching type: Packet
      Encoding type: Packet
      Maximum LSP BW [priority] bps:
```

```
          [0] 155.52Mbps   [1] 155.52Mbps  [2] 155.52Mbps  [3] 155.52Mbps
          [4] 155.52Mbps   [5] 155.52Mbps  [6] 155.52Mbps  [7] 155.52Mbps
    To: 10.2.5.1-1, Local: 10.2.5.1, Remote: 0.0.0.0
      Color: 0x4 backup
      Metric: 1
      Static BW: 100Mbps
      Reservable BW: 100Mbps
      Available BW [priority] bps:
          [0] 100Mbps      [1] 100Mbps     [2] 100Mbps     [3] 100Mbps
          [4] 100Mbps      [5] 100Mbps     [6] 100Mbps     [7] 100Mbps
      Interface Switching Capability Descriptor(1):
        Switching type: Packet
        Encoding type: Packet
        Maximum LSP BW [priority] bps:
          [0] 100Mbps      [1] 100Mbps     [2] 100Mbps     [3] 100Mbps
          [4] 100Mbps      [5] 100Mbps     [6] 100Mbps     [7] 100Mbps

user@router0>  show ted database 10.255.41.217 extensive
TED database: 0 ISIS nodes 8 INET nodes
NodeID: 10.255.41.217
  Type: Rtr, Age: 473 secs, LinkIn: 6, LinkOut: 6
  Protocol: OSPF(0.0.0.0)
    To: 10.255.41.216, Local: 10.2.3.2, Remote: 10.2.3.1
      Color: 0x2 fa
      Metric: 1
      Static BW: 155.52Mbps
      Reservable BW: 155.52Mbps
      Available BW [priority] bps:
          [0] 155.52Mbps   [1] 155.52Mbps  [2] 155.52Mbps  [3] 155.52Mbps
          [4] 155.52Mbps   [5] 155.52Mbps  [6] 155.52Mbps  [7] 155.52Mbps
      Interface Switching Capability Descriptor(1):
        Switching type: Packet
        Encoding type: Packet
        Maximum LSP BW [priority] bps:
          [0] 155.52Mbps   [1] 155.52Mbps  [2] 155.52Mbps  [3] 155.52Mbps
          [4] 155.52Mbps   [5] 155.52Mbps  [6] 155.52Mbps  [7] 155.52Mbps
    To: 10.255.41.216,  Local: 172.16.30.2, Remote: 172.16.30.1
      Metric: 1
      Static BW: 400kbps
      Reservable BW: 400kbps
      Available BW [priority] bps:
          [0] 370kbps      [1] 370kbps     [2] 370kbps     [3] 370kbps
          [4] 370kbps      [5] 370kbps     [6] 370kbps     [7] 370kbps
```

```
    Interface Switching Capability Descriptor(1):
      Switching type: Packet
      Encoding type: Packet
      Maximum LSP BW [priority] bps:
        [0] 370kbps      [1] 370kbps     [2] 370kbps     [3] 370kbps
        [4] 370kbps      [5] 370kbps     [6] 370kbps     [7] 370kbps
  To: 10.255.41.216, Local: 10.2.3.6, Remote: 10.2.3.5
    Color: 0x4 backup
    Metric: 1
    Static BW: 155.52Mbps
    Reservable BW: 155.52Mbps
    Available BW [priority] bps:
        [0] 155.52Mbps   [1] 155.52Mbps  [2] 155.52Mbps  [3] 155.52Mbps
        [4] 155.52Mbps   [5] 155.52Mbps  [6] 155.52Mbps  [7] 155.52Mbps
    Interface Switching Capability Descriptor(1):
      Switching type: Packet
      Encoding type: Packet
      Maximum LSP BW [priority] bps:
        [0] 155.52Mbps   [1] 155.52Mbps  [2] 155.52Mbps  [3] 155.52Mbps
        [4] 155.52Mbps   [5] 155.52Mbps  [6] 155.52Mbps  [7] 155.52Mbps
  To: 10.255.41.215, Local: 10.3.5.1, Remote: 10.3.5.2
    Color: 0x2 fa
    Metric: 1
    Static BW: 155.52Mbps
    Reservable BW: 155.52Mbps
    Available BW [priority] bps:
        [0] 155.12Mbps   [1] 155.12Mbps  [2] 155.12Mbps  [3] 155.12Mbps
        [4] 155.12Mbps   [5] 155.12Mbps  [6] 155.12Mbps  [7] 155.12Mbps
    Interface Switching Capability Descriptor(1):
      Switching type: Packet
      Encoding type: Packet
      Maximum LSP BW [priority] bps:
        [0] 155.12Mbps   [1] 155.12Mbps  [2] 155.12Mbps  [3] 155.12Mbps
        [4] 155.12Mbps   [5] 155.12Mbps  [6] 155.12Mbps  [7] 155.12Mbps
  To: 10.255.41.221, Local: 10.3.6.1, Remote: 10.3.6.2
    Color: 0x8 other
    Metric: 1
    Static BW: 155.52Mbps
    Reservable BW: 155.52Mbps
    Available BW [priority] bps:
        [0] 155.52Mbps   [1] 155.52Mbps  [2] 155.52Mbps  [3] 155.52Mbps
        [4] 155.52Mbps   [5] 155.52Mbps  [6] 155.52Mbps  [7] 155.52Mbps
    Interface Switching Capability Descriptor(1):
```

```
        Switching type: Packet
        Encoding type: Packet
        Maximum LSP BW [priority] bps:
          [0] 155.52Mbps   [1] 155.52Mbps  [2] 155.52Mbps  [3] 155.52Mbps
          [4] 155.52Mbps   [5] 155.52Mbps  [6] 155.52Mbps  [7] 155.52Mbps
    To: 10.3.4.1-1, Local: 10.3.4.1, Remote: 0.0.0.0
      Color: 0x4 backup
      Metric: 1
      Static BW: 100Mbps
      Reservable BW: 100Mbps
      Available BW [priority] bps:
          [0] 100Mbps      [1] 100Mbps     [2] 100Mbps     [3] 100Mbps
          [4] 100Mbps      [5] 100Mbps     [6] 100Mbps     [7] 100Mbps
      Interface Switching Capability Descriptor(1):
        Switching type: Packet
        Encoding type: Packet
        Maximum LSP BW [priority] bps:
          [0] 100Mbps      [1] 100Mbps     [2] 100Mbps     [3] 100Mbps
          [4] 100Mbps      [5] 100Mbps     [6] 100Mbps     [7] 100Mbps

user@router0>  show rsvp session name e2e_lsp_r0r5 extensive
Ingress RSVP: 1 sessions
10.255.41.221
  From: 10.255.41.222, LSPstate: Up, ActiveRoute: 2
  LSPname: e2e_lsp_r0r5, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 101584
  Resv style: 1 FF, Label in: -, Label out: 101584
  Time left:    -, Since: Wed Sep  7 19:02:56 2005
  Tspec: rate 30kbps size 30kbps peak Infbps m 20 M 1500
  Port number: sender 2 receiver 29458 protocol 0
  PATH rcvfrom: localclient
  Adspec: sent MTU 1500
  Path MTU: received 1500
  PATH sentto: 10.1.2.2 (so-0/0/3.0) 15 pkts
  RESV rcvfrom: 10.1.2.2 (so-0/0/3.0) 16 pkts
  Explct route: 10.1.2.2 172.16.30.2 10.3.6.2
   Record route: <self> 10.1.2.2 172.16.30.2 10.3.6.2
Total 1 displayed, Up 1, Down 0

Egress RSVP: 1 sessions
Total 0 displayed, Up 0, Down 0
```

```
Transit RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Router 1

On Router 1, verify that your LMP traffic engineering link configuration is working and that the end-to-end LSP is traversing the traffic engineering link by issuing the `show link-management` set of commands. You can also issue the `show rsvp session extensive` command to confirm that the FA-LSP is operational.

```
user@router1> show link-management
Peer name: r4 , System identifier: 10758
 State: Up, Control address: 10.255.41.217
   TE links:
   link_r1r4

 TE link name: link_r1r4, State: Up
  Local identifier: 16299, Remote identifier: 0, Local address: 172.16.30.1, Remote address:
172.16.30.2,
  Encoding: Packet, Switching: Packet, Minimum bandwidth: 0bps, Maximum bandwidth: 400kbps,
  Total bandwidth: 400kbps, Available bandwidth: 370kbps
    Name        State Local ID  Remote ID      Bandwidth Used  LSP-name
    fa_lsp_r1r4 Up       22642          0        400kbps  Yes  e2e_lsp_r0r5

user@router1>  show link-management te-link name link_r1r4 detail
 TE link name: link_r1r4, State: Up
  Local identifier: 16299, Remote identifier: 0, Local address: 172.16.30.1,  Remote address:
172.16.30.2,
  Encoding: Packet, Switching: Packet, Minimum bandwidth: 0bps, Maximum bandwidth: 400kbps,
  Total bandwidth: 400kbps, Available bandwidth: 370kbps
    Resource: fa_lsp_r1r4, Type: LSP, System identifier: 2147483683, State: Up, Local
identifier: 22642,
    Remote identifier: 0
    Total bandwidth: 400kbps, Unallocated bandwidth: 370kbps
    Traffic parameters: Encoding: Packet, Switching: Packet, Granularity: Unknown
     Number of allocations: 1, In use: Yes
      LSP name: e2e_lsp_r0r5,  Allocated bandwidth: 30kbps

user@router1>  show rsvp session name fa_lsp_r1r4 extensive
Ingress RSVP: 1 sessions
10.255.41.217
  From: 10.255.41.216, LSPstate: Up, ActiveRoute: 0
```

```
  LSPname: fa_lsp_r1r4, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 100816
  Resv style: 1 FF, Label in: -, Label out: 100816
  Time left:    -, Since: Wed Sep  7 19:02:33 2005
  Tspec: rate 400kbps size 400kbps peak Infbps m 20 M 1500
  Port number: sender 2 receiver 5933 protocol 0
  PATH rcvfrom: localclient
  Adspec: sent MTU 1500
  Path MTU: received 1500
  PATH sentto: 10.2.4.2 (so-0/0/2.0) 28 pkts
  RESV rcvfrom: 10.2.4.2 (so-0/0/2.0) 26 pkts
  Explct route: 10.2.4.2 10.4.5.2 10.3.5.1
  Record route: <self> 10.2.4.2 10.4.5.2 10.3.5.1
Total 1 displayed, Up 1, Down 0

Egress RSVP: 1 sessions
Total 0 displayed, Up 0, Down 0

Transit RSVP: 2 sessions
Total 0 displayed, Up 0, Down 0
```

Configuring Link Management Protocol Peers

After you set up traffic engineering links, configure LMP network peers with the `peer` *peer-name* statement at the `[edit protocols link-management]` hierarchy level. A peer is the network device with which your routing platform communicates and establishes an FA-LSP. Designate a peer name, configure the peer router ID as the address (often a loopback address), and apply the traffic engineering link to be associated with this peer. Remember to configure both sides of a peering relationship to enable bidirectional communication.

Unlike GMPLS, you must not configure a control channel for a peer. If you include a control channel, the commit operation fails.

```
[edit]
protocols {
    link-management {
        peer peer-name { # Configure the name of your network peer.
            address ip-address; # Include the router ID of the peer.
            te-link te-link-name; # Assign a TE link to this peer.
        }
```

```
    }
}
```

Configuring Link Management Protocol Traffic Engineering Links

To begin your RSVP LSP tunnel configuration, configure LMP traffic engineering links on both the ingress and egress routing platforms. Because traffic engineering links define a unidirectional connection between peer devices, you must configure traffic engineering links in both directions between peers to enable the bidirectional transport of packets.

To configure traffic engineering links in LMP, include the `te-link` *`te-link-name`* statement at the [`edit protocols link-management`] hierarchy level. Define the traffic engineering link options shown below, especially the label-switched path to be used as the FA-LSP to reach the peer. Optionally, you can specify the traffic engineering metric for the traffic engineering link (TE link). By default, the traffic engineering metric is derived from the CSPF computation.

```
[edit]
protocols {
    link-management {
        te-link te-link-name { # Name of the TE link.
            label-switched-path lsp-name; # LSP used for the forwarding adjacency.
            local-address ip-address; # Local IP address associated with the TE link.
            remote-address ip-address; # Remote IP address mapped to the TE link.
            te-metric value; # Traffic engineering metric used for the TE link.
        }
    }
}
```

Configuring Peer Interfaces in OSPF and RSVP

After you establish LMP peers, you must add peer interfaces to OSPF and RSVP. A peer interface is a virtual interface used to support the control adjacency between two peers.

The peer interface name must match the `peer` *`peer-name`* statement configured in LMP at the [`edit protocols link-management`] hierarchy level. Because actual protocol packets are sent and received by peer interfaces, the peer interfaces can be signaled and advertised to peers like any other physical interface configured for OSPF and RSVP. To configure OSPF routing for LMP peers, include the `peer-interface` statement at the [`edit protocols ospf area` *`area-number`*] hierarchy level. To configure RSVP signaling for LMP peers, include the `peer-interface` statement at the [`edit protocols rsvp`] hierarchy level.

```
[edit]
protocols {
```

```
    rsvp {
        peer-interface peer-name { # Configure the name of your LMP peer.
        }
        ospf {
            area                    area-number                 {
                peer-interface peer-name { # Configure the name of your LMP peer.
                }
            }
        }
    }
}
```

Defining Label-Switched Paths for the FA-LSP

Next, define your FA-LSP by including the `label-switched-path` statement at the `[edit protocols mpls]` hierarchy level. Include the router ID of the peer in the `to` statement at the `[edit protocols mpls label-switched-path]` hierarchy level. Because packet LSPs are unidirectional, you must create one FA-LSP to reach the peer and a second FA-LSP to return from the peer.

```
[edit]
protocols {
    mpls {
        label-switched-path lsp-name {
            from ip-address;
            to ip-address;
            primary path-name;
            secondary path-name;
            no-cspf; # This statement to disable CSPF is optional.
        }
    }
}
```

Establishing FA-LSP Path Information

When you configure explicit LSP paths for an FA-LSP, you must use the traffic engineering link remote address as your next-hop address. When CSPF is supported, you can use any path option you wish. However, when CSPF is disabled with the `no-cspf` statement at the `[edit protocols mpls label-switched-path lsp-name]` hierarchy level, you must use strict paths.

```
[edit]
protocols {
```

```
    mpls {
        path path-name {
                            next-hop-address (strict | loose);
        }
    }
}
```

NOTE: If the end-to-end LSP originates on the same routing platform as the FA-LSP, you must disable CSPF and use strict paths.

Option: Tearing Down RSVP LSPs Gracefully

You can tear down an RSVP LSP in a two-step process that gracefully withdraws the RSVP session used by the LSP. For all neighbors that support graceful teardown, a request for the teardown is sent by the routing platform to the destination endpoint for the LSP and all RSVP neighbors in the path. The request is included within the `ADMIN_STATUS` field of the RSVP packet. When neighbors receive the request, they prepare for the RSVP session to be withdrawn. A second message is sent by the routing platform to complete the teardown of the RSVP session. If a neighbor does not support graceful teardown, the request is handled as a standard session teardown rather than a graceful one.

To perform a graceful teardown of an RSVP session, issue the `clear rsvp session gracefully` command. Optionally, you can specify the source and destination address of the RSVP session, the LSP identifier of the RSVP sender, and the tunnel identifier of the RSVP session. To use these qualifiers, include the `connection-source`, `connection-destination`, `lsp-id`, and `tunnel-id` options when you issue the `clear rsvp session gracefully` command.

You can also configure the amount of time that the routing platform waits for neighbors to receive the graceful teardown request before initiating the actual teardown by including the `graceful-deletion-timeout` statement at the `[edit protocols rsvp]` hierarchy level. The default graceful deletion timeout value is 30 seconds, with a minimum value of 1 second and a maximum value of 300 seconds. To view the current value configured for graceful deletion timeout, issue the `show rsvp version` operational mode command.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
19.4R1	

16.1	However, starting from Junos OS Release 16.1, when RSVP hello messages time-out, the RSVP sessions are brought down.

RELATED DOCUMENTATION

Basic MPLS Configuration | 46

CHAPTER 11

LDP

IN THIS CHAPTER

- LDP Overview | **1316**
- LDP Configuration | **1362**
- Example: Configuring Multiple-Instance LDP | **1571**
- Tunneling LDP over SR-TE | **1598**
- Example: Tunneling LDP over SR-TE in IS-IS Network | **1602**
- Example: Tunneling LDP over SR-TE in OSPF Network | **1632**
- MPLS TTL Propagation Flexibility for LDP-signaled LSPs | **1661**
- Example: Configuring MPLS TTL Propagation for LDP-signaled LSPs | **1663**
- Understanding Multipoint LDP Recursive FEC | **1668**
- Example: Configuring Multipoint LDP Recursive FEC | **1671**

LDP Overview

IN THIS SECTION

- LDP Introduction | **1317**
- Understanding the LDP Signaling Protocol | **1317**
- Example: Configuring LDP-Signaled LSPs | **1318**
- Junos OS LDP Protocol Implementation | **1322**
- LDP Operation | **1322**
- LDP Message Types | **1322**
- Tunneling LDP LSPs in RSVP LSPs | **1324**
- Tunneling LDP LSPs in RSVP LSPs Overview | **1324**
- Tunneling LDP over SR-TE | **1325**

- Example: Tunneling LDP over SR-TE in IS-IS Network | **1329**
- Label Operations | **1359**
- LDP Session Protection | **1360**
- LDP Native IPv6 Support Overview | **1361**
- Longest Match Support for LDP Overview | **1362**

LDP Introduction

The Label Distribution Protocol (LDP) is a protocol for distributing labels in non-traffic-engineered applications. LDP allows routers to establish label-switched paths (LSPs) through a network by mapping network-layer routing information directly to data link layer-switched paths.

These LSPs might have an endpoint at a directly attached neighbor (comparable to IP hop-by-hop forwarding), or at a network egress node, enabling switching through all intermediary nodes. LSPs established by LDP can also traverse traffic-engineered LSPs created by RSVP.

LDP associates a forwarding equivalence class (FEC) with each LSP it creates. The FEC associated with an LSP specifies which packets are mapped to that LSP. LSPs are extended through a network as each router chooses the label advertised by the next hop for the FEC and splices it to the label it advertises to all other routers. This process forms a tree of LSPs that converge on the egress router.

Understanding the LDP Signaling Protocol

LDP is a signaling protocol that runs on a device configured for MPLS support. The successful configuration of both MPLS and LDP initiates the exchange of TCP packets across the LDP interfaces. The packets establish TCP-based LDP sessions for the exchange of MPLS information within the network. Enabling both MPLS and LDP on the appropriate interfaces is sufficient to establish LSPs.

LDP is a simple, fast-acting signaling protocol that automatically establishes LSP adjacencies within an MPLS network. Routers then share LSP updates such as hello packets and LSP advertisements across the adjacencies. Because LDP runs on top of an IGP such as IS-IS or OSPF, you must configure LDP and the IGP on the same set of interfaces. After both are configured, LDP begins transmitting and receiving LDP messages through all LDP-enabled interfaces. Because of LDP's simplicity, it cannot perform the true traffic engineering which RSVP can perform. LDP does not support bandwidth reservation or traffic constraints.

When you configure LDP on a *label-switching router* (LSR), the router begins sending LDP discovery messages out all LDP-enabled interfaces. When an adjacent LSR receives LDP discovery messages, it establishes an underlying TCP session. An LDP session is then created on top of the TCP session. The TCP three-way handshake ensures that the LDP session has bidirectional connectivity. After they establish the LDP session, the LDP neighbors maintain, and terminate, the session by exchanging

messages. LDP advertisement messages allow LSRs to exchange label information to determine the next hops within a particular LSP. Any topology changes, such as a router failure, generate LDP notifications that can terminate the LDP session or generate additional LDP advertisements to propagate an LSP change.

Starting in Junos OS Release 20.3R1, support for MPLS to provide LDP signaling protocol configuration with the control plane functionality.

Example: Configuring LDP-Signaled LSPs

IN THIS SECTION

- Requirements | **1318**
- Overview | **1318**
- Configuration | **1319**

This example shows how to create and configure LDP instances within an MPLS network.

Requirements

Before you begin:

- Configure network interfaces. See Interfaces User Guide for Security Devices.
- Configure an IGP across your network. (The LDP configuration is added to the existing IGP configuration and included in the MPLS configuration.)
- Configure a network to use LDP for LSP establishment by enabling MPLS on all transit interfaces in the MPLS network.

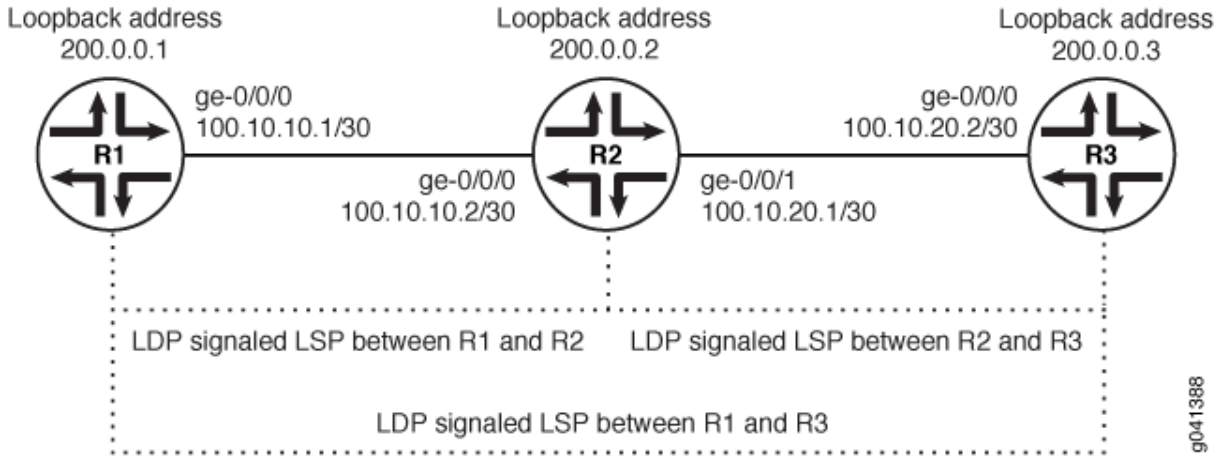
> **NOTE**: Because LDP runs on top of an IGP such as IS-IS or OSPF, you must configure LDP and the IGP on the same set of interfaces.

Overview

To configure LDP-signaled LSPs, you must enable the MPLS family on all transit interfaces in the MPLS network and include all the transit interfaces under the [`protocols mpls`] and [`protocols ldp`] hierarchy levels.

In this example, you enable the MPLS family and create an LDP instance on all the transit interfaces. Additionally, you enable the MPLS process on all the transit interfaces in the MPLS network. In this example, you configure a sample network as shown in Figure 75 on page 1319.

Figure 75: Typical LDP-Signaled LSP



Configuration

IN THIS SECTION

- Procedure | **1319**

Procedure

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level and then enter `commit` from configuration mode.

R1

```
set interfaces ge-0/0/0 unit 0 family mpls
set protocols mpls ge-0/0/0 unit 0
set protocols ldp interface ge-0/0/0 unit 0
```

R2

```
set interfaces ge-0/0/0 unit 0 family mpls
set protocols mpls ge-0/0/0 unit 0
set protocols ldp interface ge-0/0/0 unit 0
set interfaces ge-0/0/1 unit 0 family mpls
set protocols mpls ge-0/0/1 unit 0
set protocols ldp interface ge-0/0/1 unit 0
```

R3

```
set interfaces ge-0/0/0 unit 0 family mpls
set protocols mpls ge-0/0/0 unit 0
set protocols ldp interface ge-0/0/0 unit 0
```

Step-by-Step Procedure

To enable LDP instances within an MPLS network:

1. Enable the MPLS family on the transit interface on Router R1.

```
[edit]
user@R1# set interfaces ge-0/0/0 unit 0 family mpls
```

2. Enable the MPLS process on the transit interface.

```
[edit]
user@R1# set protocols mpls interface ge-0/0/0 unit 0
```

3. Create the LDP instance on the transit interface.

```
[edit]
user@R1# set protocols ldp interface ge-0/0/0 unit 0
```

Results

Confirm your configuration by entering the `show` command from configuration mode. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this `show` output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
user@R1# show
...
    interfaces {
        ge-0/0/0 {
            unit 0 {
                family inet {
                    address 10.100.37.20/24;
                }
                family mpls;
            }
        }
        }
    ...
        protocols {
            mpls {
                interface all;
            }
            ldp {
                interface ge-0/0/0.0;
            }
        }
```

If you are done configuring the device, enter the `commit` command from the configuration mode to activate the configuration.

Junos OS LDP Protocol Implementation

The Junos OS implementation of LDP supports LDP version 1. The Junos OS supports a simple mechanism for tunneling between routers in an interior gateway protocol (IGP), to eliminate the required distribution of external routes within the core. The Junos OS allows an MPLS tunnel next hop to all egress routers in the network, with only an IGP running in the core to distribute routes to egress routers. Edge routers run BGP but do not distribute external routes to the core. Instead, the recursive route lookup at the edge resolves to an LSP switched to the egress router. No external routes are necessary on the transit LDP routers.

LDP Operation

You must configure LDP for each interface on which you want LDP to run. LDP creates LSP trees rooted at each egress router for the router ID address that is the subsequent BGP next hop. The ingress point is at every router running LDP. This process provides an inet.3 route to every egress router. If BGP is running, it will attempt to resolve next hops by using the inet.3 table first, which binds most, if not all, of the BGP routes to MPLS tunnel next hops.

Two adjacent routers running LDP become neighbors. If the two routers are connected by more than one interface, they become neighbors on each interface. When LDP routers become neighbors, they establish an LDP session to exchange label information. If per-router labels are in use on both routers, only one LDP session is established between them, even if they are neighbors on multiple interfaces. For this reason, an LDP session is not related to a particular interface.

LDP operates in conjunction with a unicast routing protocol. LDP installs LSPs only when both LDP and the routing protocol are enabled. For this reason, you must enable both LDP and the routing protocol on the same set of interfaces. If this is not done, LSPs might not be established between each egress router and all ingress routers, which might result in loss of BGP-routed traffic.

You can apply policy filters to labels received from and distributed to other routers through LDP. Policy filters provide you with a mechanism to control the establishment of LSPs.

For LDP to run on an interface, MPLS must be enabled on a *logical interface* on that interface. For more information, see the Logical Interfaces.

LDP Message Types

IN THIS SECTION

- Discovery Messages | **1323**
- Session Messages | **1323**
- Advertisement Messages | **1323**

Notification Messages | 1324

LDP uses the message types described in the following sections to establish and remove mappings and to report errors. All LDP messages have a common structure that uses a type, length, and value (TLV) encoding scheme.

Discovery Messages

Discovery messages announce and maintain the presence of a router in a network. Routers indicate their presence in a network by sending hello messages periodically. Hello messages are transmitted as UDP packets to the LDP port at the group multicast address for all routers on the subnet.

LDP uses the following discovery procedures:

- Basic discovery—A router periodically sends LDP link hello messages through an interface. LDP link hello messages are sent as UDP packets addressed to the LDP discovery port. Receipt of an LDP link hello message on an interface identifies an adjacency with the LDP peer router.
- Extended discovery—LDP sessions between routers not directly connected are supported by LDP extended discovery. A router periodically sends LDP targeted hello messages to a specific address. Targeted hello messages are sent as UDP packets addressed to the LDP discovery port at the specific address. The targeted router decides whether to respond to or ignore the targeted hello message. A targeted router that chooses to respond does so by periodically sending targeted hello messages to the initiating router.

Session Messages

Session messages establish, maintain, and terminate sessions between LDP peers. When a router establishes a session with another router learned through the hello message, it uses the LDP initialization procedure over TCP transport. When the initialization procedure is completed successfully, the two routers are LDP peers and can exchange advertisement messages.

Advertisement Messages

Advertisement messages create, change, and delete label mappings for forwarding equivalence classes (FECs). Requesting a label or advertising a label mapping to a peer is a decision made by the local router. In general, the router requests a label mapping from a neighboring router when it needs one and advertises a label mapping to a neighboring router when it wants the neighbor to use a label.

Notification Messages

Notification messages provide advisory information and signal error information. LDP sends notification messages to report errors and other events of interest. There are two kinds of LDP notification messages:

- Error notifications, which signal fatal errors. If a router receives an error notification from a peer for an LDP session, it terminates the LDP session by closing the TCP transport connection for the session and discarding all label mappings learned through the session.
- Advisory notifications, which pass information to a router about the LDP session or the status of some previous message received from the peer.

Tunneling LDP LSPs in RSVP LSPs

You can tunnel LDP LSPs over RSVP LSPs. The following sections describe how tunneling of LDP LSPs in RSVP LSPs works:

- "Tunneling LDP LSPs in RSVP LSPs Overview" on page 1324
- "Label Operations" on page 1359

Tunneling LDP LSPs in RSVP LSPs Overview

IN THIS SECTION

- Benefits of Tunneling LDP LSPs in RSVP LSPs | **1325**

If you are using RSVP for traffic engineering, you can run LDP simultaneously to eliminate the distribution of external routes in the core. The LSPs established by LDP are tunneled through the LSPs established by RSVP. LDP effectively treats the traffic-engineered LSPs as single hops.

When you configure the router to run LDP across RSVP-established LSPs, LDP automatically establishes sessions with the router at the other end of the LSP. LDP control packets are routed hop-by-hop, rather than carried through the LSP. This routing allows you to use simplex (one-way) traffic-engineered LSPs. Traffic in the opposite direction flows through LDP-established LSPs that follow unicast routing rather than through traffic-engineered tunnels.

If you configure LDP over RSVP LSPs, you can still configure multiple OSPF areas and IS-IS levels in the traffic engineered core and in the surrounding LDP cloud.

Beginning with Junos OS Release 15.1, multi-instance support is extended to LDP over RSVP tunneling for a virtual router routing instance. This allows splitting of a single routing and MPLS domain into

multiple domains so that each domain can be scaled independently. BGP labeled unicast can be used to stitch these domains for service forwarding equivalence classes (FECs). Each domain uses intra-domain LDP-over-RSVP LSP for MPLS forwarding.

NOTE: With the introduction of the multi-instance support for LDP-over-RSVP LSPs, you cannot enable MPLS on an interface that is already assigned to another routing instance. Adding an interface that is part of another routing instance at the `[edit protocols mpls]` hierarchy level, throws a configuration error at the time of commit.

Benefits of Tunneling LDP LSPs in RSVP LSPs

Tunneling LDP LSPs in RSVP LSPs provides the following benefits:

- Provides convergence of different traffic types such as IPv4, IPv6, unicast, and multicast across Layer 2 and Layer 3 VPNs.
- Enables flexible access connectivity options that can accommodate multiple topologies, different protocols, and multiple administrative boundaries.
- Enables secure interworking among multiple providers.
- Enables provision of differentiated services on a per customer basis because RSVP-TE supports traffic engineering, bandwidth guarantees, and link and node redundancy capabilities.
- Reduces the number of LSPs required in the core, which reduces the resource requirements of the protocols and routers as well as reducing convergence time.
- Provides cost-efficient rollouts with minimal network disruption because the LSPs are built using point-to-point TE tunnels to directly attached neighbors. These TE tunnels only go to the next hop, not end to end. Then when LDP is run over those tunnels, the sessions are built to the directly connected neighbor. When there is a change in the network, such as adding a new node, the directly connected neighbors of the new node have RSVP and LDP sessions. Thus, the RSVP LSPs are only to the next hop, and LDP takes care of advertising labels for the new addresses.

Tunneling LDP over SR-TE

IN THIS SECTION

- Benefits of Tunneling LDP over SR-TE | **1326**
- Tunneling LDP over SR-TE Overview | **1326**

Learn about the benefits and get an overview of tunneling LDP over SR-TE.

Benefits of Tunneling LDP over SR-TE

- Enables seamless integration of LDP over SR-TE in the core network.
- Provides flexible connectivity options to accommodate multiple topologies, protocols, and domains.
- Enables interoperability between LDP and SR capable devices.
- Leverages SR-TE load sharing capabilities.
- Provides faster restoration of network connectivity using Topology Independent Loop-Free Alternate (TI-LFA) within the SR-TE domain. SR using TI-LFA routes the traffic instantly to a backup or an alternate path if the primary path fails or becomes unavailable.

Tunneling LDP over SR-TE Overview

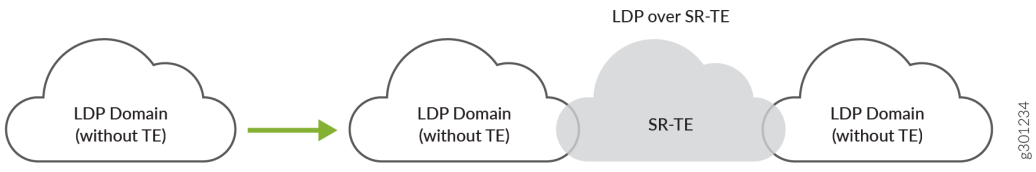
It's common for service providers to use the LDP signaling protocol with MPLS transport at the edges of their networks. LDP offers the advantage of being simple, but LDP lacks traffic engineering (TE) and sophisticated path repair capabilities that are often desired in the network's core. Many service providers are migrating from RSVP to segment routing traffic engineering (SR-TE) in the core. SR-TE is also referred to as source routing in packet networks (SPRING).

It's possible that the routers running LDP at the edge may not support SR capabilities. The service provider may wish to continue using LDP on these routers to avoid the need for an upgrade. In such scenarios, the LDP over SR-TE tunneling feature provides the ability to integrate routers that are not SR capable (running LDP) with routers that are SR capable (running SR-TE).

The LDP LSPs are tunneled through the SR-TE network, enabling interworking of LDP LSPs with SR-TE LSPs. For example, if you have LDP domains on the provider edge network and SR-TE in the core network, then you can connect the LDP domains over SR-TE, as shown in Figure 76 on page 1326.

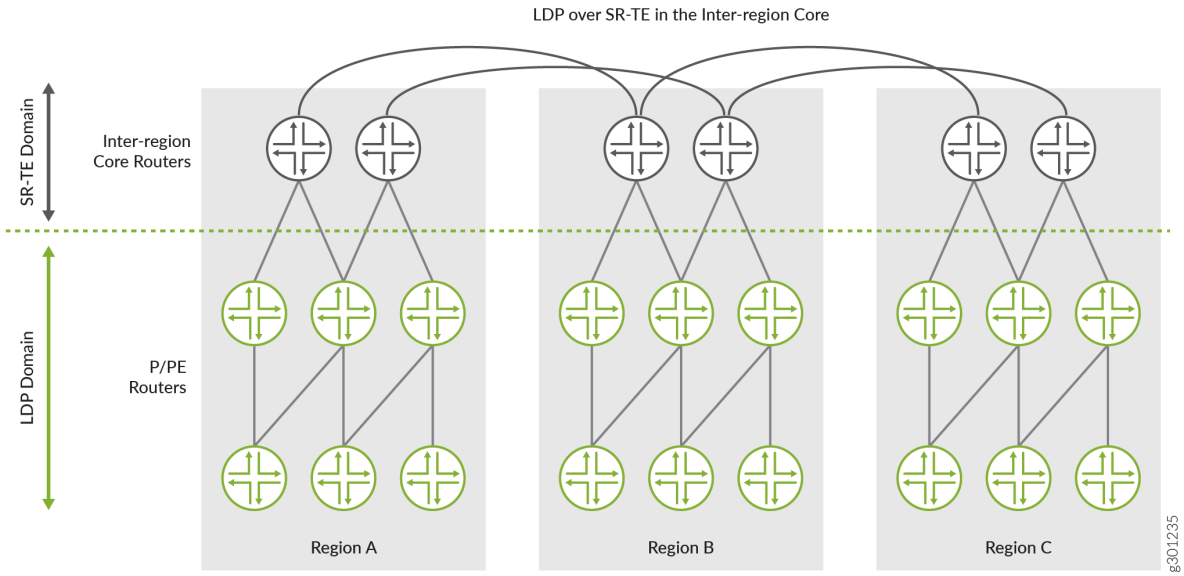
Tunneling LDP over SR-TE supports co-existence of both LDP LSPs and SR-TE LSPs.

Figure 76: Interconnect LDP Domains over SR-TE in the Core Network



You can also tunnel LDP over SR-TE between LDP domains connected to inter-region core networks. For example, if you have multiple regional LDP domains connected to the inter-region SR-TE core networks, you can tunnel LDP across the inter-region SR-TE core network, as shown in Figure 77 on page 1327.

Figure 77: LDP over SR-TE between Inter-region Core Networks



In Figure 77 on page 1327, you have three regional networks (A, B, and C) running LDP. These regional LDP domains are connected to their respective regional core networks running SR-TE. The regional SR-TE core networks are further interconnected to other regional SR-TE core networks (inter-region core network). You can tunnel LDP over these inter-region SR-TE core networks and deploy services, such as Layer 3 VPNs, seamlessly. This scenario could be used in a mobile backhaul network, where the core aggregation layer runs LDP tunneled over SR-TE while the access layer runs LDP only.

To enable LDP tunneling over SR-TE in IS-IS networks, you need to configure the following configuration statements:

- `ldp-tunneling` at the [`edit protocols source-packet-routing source-routing-path` *`source-routing-path-name`*] hierarchy level to enable LDP tunneling over SR-TE.
- `spring-te` at the [`edit protocols isis traffic-engineering tunnel-source-protocol`] hierarchy level selects LDP over SR-TE LSPs as the tunnel source protocol.

To enable LDP tunneling over SR-TE in OSPF networks, you need to configure the following configuration statements:

- `ldp-tunneling` at the [`edit protocols source-packet-routing source-routing-path` *`source-routing-path-name`*] hierarchy level to enable LDP tunneling over SR-TE.
- `spring-te` at the [`edit protocols ospf traffic-engineering tunnel-source-protocol`] hierarchy level selects LDP over SR-TE LSPs as the tunnel source protocol.

You can configure more than one tunnel source protocol for IGPs (IS-IS and OSPF) to create shortcut routes. When more than one tunnel source protocol is configured and if the tunnels from more than one protocol are available to a destination, the tunnel with the most preferred route is established. For example, if the core network has both RSVP LSPs and SR-TE LSPs and LDP tunneling is enabled for both RSVP and SR-TE LSPs, then the `tunnel-source-protocol` configuration selects the tunnel based on the preference value. The tunnel with the lowest preference value is most preferred. You can override this route preference with a specific protocol for all destinations by configuring the preference value, as shown in the following example:

```
[edit]
user@host#set protocols isis traffic-engineering tunnel-source-protocol spring-te preference 2
user@host#set protocols isis traffic-engineering tunnel-source-protocol rsvp preference 5
```

```
[edit]
user@host#set protocols ospf traffic-engineering tunnel-source-protocol spring-te preference 2
user@host#set protocols ospf traffic-engineering tunnel-source-protocol rsvp preference 5
```

In this example, you can see the preference value configured for the SR-TE tunnel source protocol is 2 and the preference value for RSVP tunnel source protocol is 5. In this case, the SR-TE tunnel are preferred because they have the lowest preference value as compared to RSVP tunnel source protocol.

NOTE: It is not mandatory to configure the tunnel source protocol preference value. If more than one tunnel source protocol has the same preference value, then the tunnel is established based on the preferred route to the destination.

The targeted LDP session is established and is triggered when the SR-TE LSP comes up. The LSP session remains established until the LDP tunneling (`ldp-tunneling`) configuration is removed, or the SR-TE LSP is removed from the configuration.

NOTE: Junos OS currently does not support LDP over colored SR-TE LSPs.

Example: Tunneling LDP over SR-TE in IS-IS Network

IN THIS SECTION

- Requirements | 1329
- Overview | 1329
- Configuration | 1330
- Verification | 1351

Use this example to learn how to tunnel LDP LSPs over SR-TE in your core network.

NOTE: Our content testing team has validated and updated this example.

Requirements

This example uses the following hardware and software components:

- MX Series routers as CE, PE, and core routers.
- Junos OS Release 20.3R1 or later running on all devices.
  - Updated and revalidated using vMX on Junos OS Release 21.1R1.

NOTE: Are you interested in getting hands-on experience on this feature?
Visit Juniper vLabs to reserve your pre-configured vLab Sandbox: Segment Routing - Basic and try it out for free!

Overview

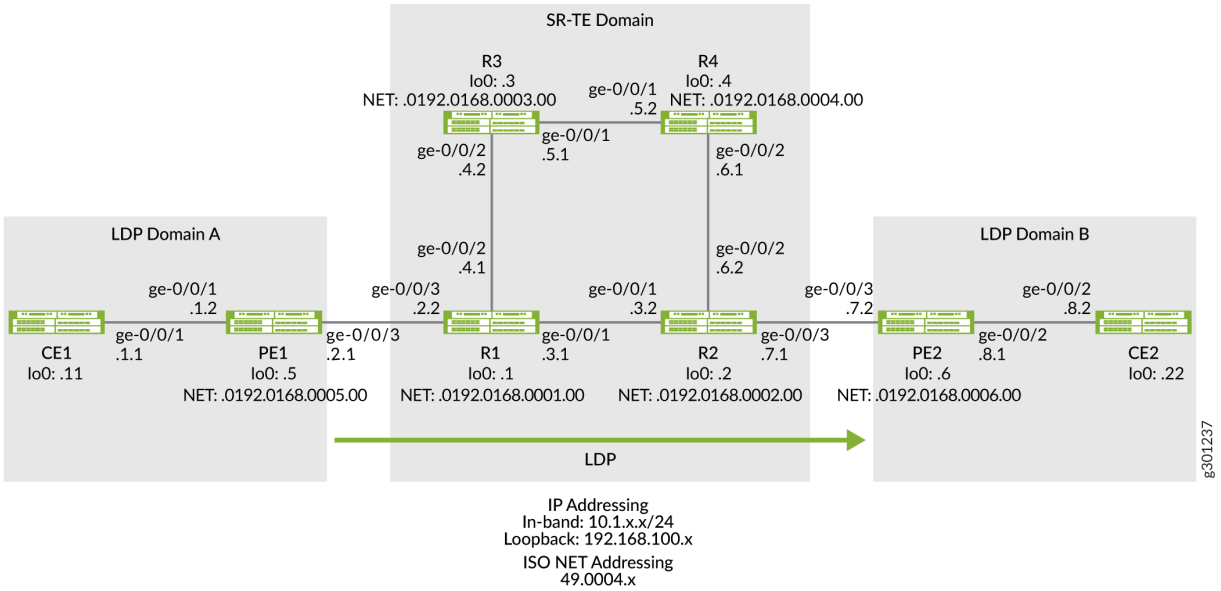
IN THIS SECTION

- Topology | 1330

The following topology (Figure 78 on page 1330) shows two LDP domains (LDP Domain A and LDP Domain B) connected to the SR-TE core network, which extends the LSP session over the core by tunneling them over SR-TE.

Topology

Figure 78: Tunneling LDP over SR-TE in the Core Network



Configuration

IN THIS SECTION

- CLI Quick Configuration | **1331**
- Configuring PE1 | **1338**
- Configuring R1 Device | **1344**

To tunnel LDP LSP over SR-TE in your core network, perform these tasks:

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

Device CE1

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/1 description CE1-to-PE1
set interfaces ge-0/0/1 unit 0 family inet address 10.1.1.1/24
set interfaces lo0 unit 0 family inet address 192.168.100.11/32
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set routing-options router-id 192.168.100.11
```

Device PE1

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/1 description PE1-to-CE1
set interfaces ge-0/0/1 unit 0 family inet address 10.1.1.2/24
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/3 description PE1-to-R1
set interfaces ge-0/0/3 unit 0 family inet address 10.1.2.1/24
set interfaces ge-0/0/3 unit 0 family iso
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.100.5/32
set interfaces lo0 unit 0 family iso address 49.0004.0192.0168.0005.00
set interfaces lo0 unit 0 family mpls
set policy-options policy-statement export_bgp term a from protocol bgp
set policy-options policy-statement export_bgp term a from protocol direct
set policy-options policy-statement export_bgp term a then accept
set routing-instances CE1_vpn1 instance-type vrf
set routing-instances CE1_vpn1 protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set routing-instances CE1_vpn1 protocols ospf export export_bgp
set routing-instances CE1_vpn1 interface ge-0/0/1.0
set routing-instances CE1_vpn1 route-distinguisher 192.168.100.5:1
set routing-instances CE1_vpn1 vrf-target target:100:4
set routing-instances CE1_vpn1 vrf-table-label
set protocols bgp group ibgp1 type internal
```

```
set protocols bgp group ibgp1 local-address 192.168.100.5
set protocols bgp group ibgp1 family inet unicast
set protocols bgp group ibgp1 family inet-vpn unicast
set protocols bgp group ibgp1 neighbor 192.168.100.6
set protocols isis interface ge-0/0/3.0 point-to-point
set protocols isis interface lo0.0 passive
set protocols ldp interface ge-0/0/3.0
set protocols ldp interface lo0.0
set protocols mpls interface ge-0/0/3.0
set protocols mpls interface lo0.0
set routing-options router-id 192.168.100.5
set routing-options autonomous-system 65410
```

Device R1

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/1 description R1-to-R2
set interfaces ge-0/0/1 unit 0 family inet address 10.1.3.1/24
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/2 description R1-to-R3
set interfaces ge-0/0/2 unit 0 family inet address 10.1.4.1/24
set interfaces ge-0/0/2 unit 0 family iso
set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/3 description R1-to-PE1
set interfaces ge-0/0/3 unit 0 family inet address 10.1.2.2/24
set interfaces ge-0/0/3 unit 0 family iso
set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
set interfaces lo0 unit 0 family inet address 192.168.100.1/32
set interfaces lo0 unit 0 family iso address 49.0004.0192.0168.0001.00
set interfaces lo0 unit 0 family mpls
set protocols isis interface ge-0/0/1.0 level 2 ipv4-adjacency-segment protected index 108
set protocols isis interface ge-0/0/1.0 level 2 ipv4-adjacency-segment unprotected index 110
set protocols isis interface ge-0/0/1.0 level 2 ipv6-adjacency-segment protected index 109
set protocols isis interface ge-0/0/1.0 level 2 ipv6-adjacency-segment unprotected index 111
set protocols isis interface ge-0/0/1.0 level 2 post-convergence-lfa
set protocols isis interface ge-0/0/1.0 point-to-point
set protocols isis interface ge-0/0/2.0 level 2 ipv4-adjacency-segment protected index 104
set protocols isis interface ge-0/0/2.0 level 2 ipv4-adjacency-segment unprotected index 106
set protocols isis interface ge-0/0/2.0 level 2 ipv6-adjacency-segment protected index 105
set protocols isis interface ge-0/0/2.0 level 2 ipv6-adjacency-segment unprotected index 107
```

```
set protocols isis interface ge-0/0/2.0 level 2 post-convergence-lfa
set protocols isis interface ge-0/0/2.0 point-to-point
set protocols isis interface ge-0/0/3.0 level 2 ipv4-adjacency-segment protected index 100
set protocols isis interface ge-0/0/3.0 level 2 ipv4-adjacency-segment unprotected index 102
set protocols isis interface ge-0/0/3.0 level 2 ipv6-adjacency-segment protected index 101
set protocols isis interface ge-0/0/3.0 level 2 ipv6-adjacency-segment unprotected index 103
set protocols isis interface ge-0/0/3.0 level 2 post-convergence-lfa
set protocols isis interface ge-0/0/3.0 point-to-point
set protocols isis interface lo0.0 passive
set protocols isis source-packet-routing srgb start-label 80000
set protocols isis source-packet-routing srgb index-range 50000
set protocols isis source-packet-routing node-segment ipv4-index 5001
set protocols isis source-packet-routing node-segment ipv6-index 5501
set protocols isis level 1 disable
set protocols isis backup-spf-options use-post-convergence-lfa
set protocols isis backup-spf-options use-source-packet-routing
set protocols isis traffic-engineering l3-unicast-topology
set protocols isis traffic-engineering credibility-protocol-preference
set protocols isis traffic-engineering tunnel-source-protocol spring-te
set protocols ldp auto-targeted-session
set protocols ldp preference 1
set protocols ldp interface ge-0/0/1.0
set protocols ldp interface ge-0/0/3.0
set protocols ldp interface lo0.0
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols mpls interface ge-0/0/3.0
set protocols mpls interface lo0.0
set protocols source-packet-routing segment-list seg1 inherit-label-nexthops
set protocols source-packet-routing segment-list seg1 auto-translate
set protocols source-packet-routing segment-list seg1 hop1 ip-address 10.1.4.2
set protocols source-packet-routing segment-list seg1 hop2 ip-address 10.1.5.2
set protocols source-packet-routing segment-list seg1 hop3 ip-address 10.1.6.2
set protocols source-packet-routing source-routing-path sr_static_r5 ldp-tunneling
set protocols source-packet-routing source-routing-path sr_static_r5 to 192.168.100.2
set protocols source-packet-routing source-routing-path sr_static_r5 binding-sid 1003001
set protocols source-packet-routing source-routing-path sr_static_r5 primary seg1
set routing-options router-id 192.168.100.1
```

Device R2

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/1 description R2-to-R1
set interfaces ge-0/0/1 unit 0 family inet address 10.1.3.2/24
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/2 description R2-to-R4
set interfaces ge-0/0/2 unit 0 family inet address 10.1.6.2/24
set interfaces ge-0/0/2 unit 0 family iso
set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/3 description R2-to-PE2
set interfaces ge-0/0/3 unit 0 family inet address 10.1.7.1/24
set interfaces ge-0/0/3 unit 0 family iso
set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
set interfaces lo0 unit 0 family inet address 192.168.100.2/32
set interfaces lo0 unit 0 family iso address 49.0004.0192.0168.0002.00
set interfaces lo0 unit 0 family mpls
set protocols isis interface ge-0/0/1.0 level 2 ipv4-adjacency-segment protected index 500
set protocols isis interface ge-0/0/1.0 level 2 ipv4-adjacency-segment unprotected index 502
set protocols isis interface ge-0/0/1.0 level 2 ipv6-adjacency-segment protected index 501
set protocols isis interface ge-0/0/1.0 level 2 ipv6-adjacency-segment unprotected index 503
set protocols isis interface ge-0/0/1.0 level 2 post-convergence-lfa
set protocols isis interface ge-0/0/1.0 point-to-point
set protocols isis interface ge-0/0/2.0 level 2 ipv4-adjacency-segment protected index 504
set protocols isis interface ge-0/0/2.0 level 2 ipv4-adjacency-segment unprotected index 506
set protocols isis interface ge-0/0/2.0 level 2 ipv6-adjacency-segment protected index 505
set protocols isis interface ge-0/0/2.0 level 2 ipv6-adjacency-segment unprotected index 507
set protocols isis interface ge-0/0/2.0 level 2 post-convergence-lfa
set protocols isis interface ge-0/0/2.0 point-to-point
set protocols isis interface ge-0/0/3.0 level 2 ipv4-adjacency-segment protected index 508
set protocols isis interface ge-0/0/3.0 level 2 ipv4-adjacency-segment unprotected index 510
set protocols isis interface ge-0/0/3.0 level 2 ipv6-adjacency-segment protected index 509
set protocols isis interface ge-0/0/3.0 level 2 ipv6-adjacency-segment unprotected index 511
set protocols isis interface ge-0/0/3.0 level 2 post-convergence-lfa
set protocols isis interface ge-0/0/3.0 point-to-point
set protocols isis interface lo0.0 passive
set protocols isis source-packet-routing srgb start-label 80000
set protocols isis source-packet-routing srgb index-range 50000
set protocols isis source-packet-routing node-segment ipv4-index 5005
set protocols isis source-packet-routing node-segment ipv6-index 5505
```

```
set protocols isis source-packet-routing traffic-statistics statistics-granularity per-interface
set protocols isis level 1 disable
set protocols isis backup-spf-options use-post-convergence-lfa
set protocols isis backup-spf-options use-source-packet-routing
set protocols isis traffic-engineering l3-unicast-topology
set protocols isis traffic-engineering credibility-protocol-preference
set protocols isis traffic-engineering tunnel-source-protocol spring-te
set protocols ldp interface ge-0/0/1.0
set protocols ldp interface ge-0/0/3.0
set protocols ldp interface lo0.0
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols mpls interface ge-0/0/3.0
set protocols mpls interface lo0.0
set protocols source-packet-routing segment-list seg1 inherit-label-nexthops
set protocols source-packet-routing segment-list seg1 auto-translate
set protocols source-packet-routing segment-list seg1 hop1 ip-address 10.1.6.1
set protocols source-packet-routing segment-list seg1 hop2 ip-address 10.1.5.1
set protocols source-packet-routing segment-list seg1 hop3 ip-address 10.1.4.1
set protocols source-packet-routing source-routing-path sr_static_r1 ldp-tunneling
set protocols source-packet-routing source-routing-path sr_static_r1 to 192.168.100.1
set protocols source-packet-routing source-routing-path sr_static_r1 binding-sid 1003001
set protocols source-packet-routing source-routing-path sr_static_r1 primary seg1
set routing-options router-id 192.168.100.2
```

Device R3

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/1 description R3-to-R4
set interfaces ge-0/0/1 unit 0 family inet address 10.1.5.1/24
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 description R3-to-R1
set interfaces ge-0/0/2 unit 0 family inet address 10.1.4.2/24
set interfaces ge-0/0/2 unit 0 family iso
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.100.3/32
set interfaces lo0 unit 0 family iso address 49.0004.0192.0168.0003.00
set interfaces lo0 unit 0 family mpls
set protocols isis interface ge-0/0/1.0 level 2 ipv4-adjacency-segment protected index 204
set protocols isis interface ge-0/0/1.0 level 2 ipv4-adjacency-segment unprotected index 206
```

```
set protocols isis interface ge-0/0/1.0 level 2 ipv6-adjacency-segment protected index 205
set protocols isis interface ge-0/0/1.0 level 2 ipv6-adjacency-segment unprotected index 207
set protocols isis interface ge-0/0/1.0 level 2 post-convergence-lfa
set protocols isis interface ge-0/0/1.0 point-to-point
set protocols isis interface ge-0/0/2.0 level 2 ipv4-adjacency-segment protected index 200
set protocols isis interface ge-0/0/2.0 level 2 ipv4-adjacency-segment unprotected index 202
set protocols isis interface ge-0/0/2.0 level 2 ipv6-adjacency-segment protected index 201
set protocols isis interface ge-0/0/2.0 level 2 ipv6-adjacency-segment unprotected index 203
set protocols isis interface ge-0/0/2.0 level 2 post-convergence-lfa
set protocols isis interface ge-0/0/2.0 point-to-point
set protocols isis interface lo0.0 passive
set protocols isis source-packet-routing srgb start-label 80000
set protocols isis source-packet-routing srgb index-range 50000
set protocols isis source-packet-routing node-segment ipv4-index 5003
set protocols isis source-packet-routing node-segment ipv6-index 5503
set protocols isis level 1 disable
set protocols isis backup-spf-options use-post-convergence-lfa
set protocols isis backup-spf-options use-source-packet-routing
set protocols isis traffic-engineering l3-unicast-topology
set protocols isis traffic-engineering credibility-protocol-preference
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols mpls interface lo0.0
set routing-options router-id 192.168.100.3
```

Device R4

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/1 description R4-to-R3
set interfaces ge-0/0/1 unit 0 family inet address 10.1.5.2/24
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 description R4-to-R2
set interfaces ge-0/0/2 unit 0 family inet address 10.1.6.1/24
set interfaces ge-0/0/2 unit 0 family iso
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.100.4/32
set interfaces lo0 unit 0 family iso address 49.0004.0192.0168.0004.00
set interfaces lo0 unit 0 family mpls
set protocols isis interface ge-0/0/1.0 level 2 ipv4-adjacency-segment protected index 300
set protocols isis interface ge-0/0/1.0 level 2 ipv4-adjacency-segment unprotected index 302
```

```
set protocols isis interface ge-0/0/1.0 level 2 ipv6-adjacency-segment protected index 301
set protocols isis interface ge-0/0/1.0 level 2 ipv6-adjacency-segment unprotected index 303
set protocols isis interface ge-0/0/1.0 level 2 post-convergence-lfa
set protocols isis interface ge-0/0/1.0 point-to-point
set protocols isis interface ge-0/0/2.0 level 2 ipv4-adjacency-segment protected index 304
set protocols isis interface ge-0/0/2.0 level 2 ipv4-adjacency-segment unprotected index 306
set protocols isis interface ge-0/0/2.0 level 2 ipv6-adjacency-segment protected index 305
set protocols isis interface ge-0/0/2.0 level 2 ipv6-adjacency-segment unprotected index 307
set protocols isis interface ge-0/0/2.0 level 2 post-convergence-lfa
set protocols isis interface ge-0/0/2.0 point-to-point
set protocols isis interface lo0.0 passive
set protocols isis source-packet-routing srgb start-label 80000
set protocols isis source-packet-routing srgb index-range 50000
set protocols isis source-packet-routing node-segment ipv4-index 5004
set protocols isis source-packet-routing node-segment ipv6-index 5504
set protocols isis source-packet-routing traffic-statistics statistics-granularity per-interface
set protocols isis level 1 disable
set protocols isis backup-spf-options use-post-convergence-lfa
set protocols isis backup-spf-options use-source-packet-routing
set protocols isis traffic-engineering l3-unicast-topology
set protocols isis traffic-engineering credibility-protocol-preference
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols mpls interface lo0.0
set routing-options router-id 192.168.100.4
```

Device PE2

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/2 description PE2-to-CE2
set interfaces ge-0/0/2 unit 0 family inet address 10.1.8.1/24
set interfaces ge-0/0/2 unit 0 family iso
set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/3 description PE2-to-R2
set interfaces ge-0/0/3 unit 0 family inet address 10.1.7.2/24
set interfaces ge-0/0/3 unit 0 family iso
set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
set interfaces lo0 unit 0 family inet address 192.168.100.6/32
set interfaces lo0 unit 0 family iso address 49.0004.0192.0168.0006.00
set interfaces lo0 unit 0 family mpls
set policy-options policy-statement export_bgp term a from protocol bgp
```

```
set policy-options policy-statement export_bgp term a from protocol direct
set policy-options policy-statement export_bgp term a then accept
set routing-instances CE2_vpn1 instance-type vrf
set routing-instances CE2_vpn1 protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set routing-instances CE2_vpn1 protocols ospf export export_bgp
set routing-instances CE2_vpn1 interface ge-0/0/2.0
set routing-instances CE2_vpn1 route-distinguisher 192.168.100.6:1
set routing-instances CE2_vpn1 vrf-target target:100:4
set routing-instances CE2_vpn1 vrf-table-label
set protocols bgp group ibgp1 type internal
set protocols bgp group ibgp1 local-address 192.168.100.6
set protocols bgp group ibgp1 family inet unicast
set protocols bgp group ibgp1 family inet-vpn unicast
set protocols bgp group ibgp1 neighbor 192.168.100.5
set protocols isis interface ge-0/0/3.0 point-to-point
set protocols isis interface lo0.0 passive
set protocols ldp interface ge-0/0/3.0
set protocols ldp interface lo0.0
set protocols mpls interface ge-0/0/3.0
set protocols mpls interface lo0.0
set routing-options router-id 192.168.100.6
set routing-options autonomous-system 65410
```

Device CE2

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/1 description CE2-to-PE2
set interfaces ge-0/0/2 unit 0 family inet address 10.1.8.2/24
set interfaces lo0 unit 0 family inet address 192.168.100.22/32
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set routing-options router-id 192.168.100.22
```

Configuring PE1

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the CLI User Guide.

To configure device PE1:

1. Configure the network services mode as Enhanced IP. Enhanced IP sets the router's network services to enhanced Internet Protocol and uses enhanced mode capabilities.

```
[edit chassis]
user@PE1#set network-services enhanced-ip
```

   After you configure the `enhanced-ip` statement and commit the configuration, the following warning message appears prompting you to reboot the router:

```
'chassis'
WARNING: Chassis configuration for network services has been changed. A system reboot is
mandatory. Please reboot the system NOW. Continuing without a reboot might result in
unexpected system behavior.
commit complete
```

   The reboot brings up the FPCs on the router.

2. Configure the device's interfaces.

```
[edit interfaces]
user@PE1#set ge-0/0/1 description PE1-to-CE1
user@PE1#set ge-0/0/1 unit 0 family inet address 10.1.1.2/24
user@PE1#set ge-0/0/1 unit 0 family iso

user@PE1#set ge-0/0/3 description PE1-to-R1
user@PE1#set ge-0/0/3 unit 0 family inet address 10.1.2.1/24
user@PE1#set ge-0/0/3 unit 0 family iso
user@PE1#set ge-0/0/3 unit 0 family mpls

user@PE1#set lo0 unit 0 family inet address 192.168.100.5/32
user@PE1#set lo0 unit 0 family iso address 49.0004.0192.0168.0005.00
user@PE1#set lo0 unit 0 family mpls
```

3. Configure policy options to export BGP routes to the CE router, which runs the OSPF protocol in this example.

```
[edit policy-options]
user@PE1#set policy-statement export_bgp term a from protocol bgp
```

```
user@PE1#set policy-statement export_bgp term a from protocol direct
user@PE1#set policy-statement export_bgp term a then accept
```

4. Configure a Layer 3 VPN routing instance to support the OSPF-based CE1 device.

```
[edit routing-instances]
user@PE1#set CE1_vpn1 instance-type vrf
user@PE1#set CE1_vpn1 protocols ospf area 0.0.0.0 interface ge-0/0/1.0
user@PE1#set CE1_vpn1 protocols ospf export export_bgp
user@PE1#set CE1_vpn1 interface ge-0/0/1.0
user@PE1#set CE1_vpn1 route-distinguisher 192.168.100.5:1
user@PE1#set CE1_vpn1 vrf-target target:100:4
user@PE1#set CE1_vpn1 vrf-table-label
```

5. Configure the router ID and autonomous system number for Device PE1.

```
[edit routing-options]
user@PE1#set router-id 192.168.100.5
userPE1# set autonomous-system 65410
```

6. Configure ISIS, LDP, and MPLS on the interfaces connected to the core network.

```
[edit protocols]
user@PE1#set isis interface ge-0/0/3.0 point-to-point
user@PE1#set isis interface lo0.0 passive

user@PE1#set ldp interface ge-0/0/3.0
user@PE1#set ldp interface lo0.0

user@PE1#set mpls interface ge-0/0/3.0
user@PE1#set mpls interface lo0.0
```

7. Configure BGP between the PE devices.

```
[edit protocols]
user@PE1#set bgp group ibgp1 type internal
user@PE1#set bgp group ibgp1 local-address 192.168.100.5
user@PE1#set bgp group ibgp1 family inet unicast
```

```
user@PE1#set bgp group ibgp1 family inet-vpn unicast
user@PE1#set bgp group ibgp1 neighbor 192.168.100.6
```

Results

From configuration mode, confirm your configuration by entering the `show chassis`, `show interfaces`, `show policy-options`, `show routing-instances`,`show routing-options`, and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@PE1#show chassis
network-services enhanced-ip;
```

```
user@PE1#show interfaces
ge-0/0/1 {
    description PE1-to-CE1;
    unit 0 {
        family inet {
            address 10.1.1.2/24;
        }
        family iso;
    }
}
ge-0/0/3 {
    description PE1-to-R1;
    unit 0 {
        family inet {
            address 10.1.2.1/24;
        }
        family iso;
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 192.168.100.5/32;
        }
```

```
        family iso {
            address 49.0004.0192.0168.0005.00;
        }
        family mpls;
    }
}
```

```
user@PE1#show policy-options
policy-statement export_bgp {
    term a {
        from protocol [ bgp direct ];
        then accept;
    }
}
```

```
user@PE1#show routing-instances
CE1_vpn1 {
    instance-type vrf;
    protocols {
        ospf {
            area 0.0.0.0 {
                interface ge-0/0/1.0;
            }
            export export_bgp;
        }
    }
    interface ge-0/0/1.0;
    route-distinguisher 192.168.100.5:1;
    vrf-target target:100:4;
    vrf-table-label;
}
```

```
user@PE1#show routing-options
```

```
router-id 192.168.100.5;
autonomous-system 65410;
```

```
user@PE1#show protocols
bgp {
    group ibgp1 {
        type internal;
        local-address 192.168.100.5;
        family inet {
            unicast;
        }
        family inet-vpn {
            unicast;
        }
        neighbor 192.168.100.6;
    }
}
isis {
    interface ge-0/0/3.0 {
        point-to-point;
    }
    interface lo0.0 {
        passive;
    }
}
ldp {
    interface ge-0/0/3.0;
    interface lo0.0;
}
mpls {
    interface ge-0/0/3.0;
    interface lo0.0;
}
```

Configuring R1 Device

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the CLI User Guide.

To configure device R1:

1. Configure the network services mode as Enhanced IP. Enhanced IP sets the router's network services to enhanced Internet Protocol and uses enhanced mode capabilities.

```
[edit chassis]
user@R1#set network-services enhanced-ip
```

   After you configure the `enhanced-ip` statement and commit the configuration, the following warning message appears prompting you to reboot the router:

```
'chassis'
WARNING: Chassis configuration for network services has been changed. A system reboot is
mandatory. Please reboot the system NOW. Continuing without a reboot might result in
unexpected system behavior.
commit complete
```

   The reboot brings up the FPCs on the router.

2. Configure the device's interfaces.

```
[edit interfaces]
user@R1#set ge-0/0/1 description R1-to-R2
user@R1#set ge-0/0/1 unit 0 family inet address 10.1.3.1/24
user@R1#set ge-0/0/1 unit 0 family iso
user@R1#set ge-0/0/1 unit 0 family mpls maximum-labels 8

user@R1#set ge-0/0/2 description R1-to-R3
user@R1#set ge-0/0/2 unit 0 family inet address 10.1.4.1/24
user@R1#set ge-0/0/2 unit 0 family iso
user@R1#set ge-0/0/2 unit 0 family mpls maximum-labels 8

user@R1#set ge-0/0/3 description R1-to-PE1
user@R1#set ge-0/0/3 unit 0 family inet address 10.1.2.2/24
```

```
user@R1#set ge-0/0/3 unit 0 family iso
user@R1#set ge-0/0/3 unit 0 family mpls maximum-labels 8

user@R1#set lo0 unit 0 family inet address 192.168.100.1/32
user@R1#set lo0 unit 0 family iso address 49.0004.0192.0168.0001.00
user@R1#set lo0 unit 0 family mpls
```

3. Configure routing options to identify the router in the domain.

```
[edit routing-options]
user@R1#set router-id 192.168.100.1
```

4. Configure ISIS adjacency SIDs on the interfaces and allocate SRGB labels to enable segment routing. The labels in the entire SRGB are available for ISIS. Prefix SIDs (and Node SIDs) are indexed from the SRGB.

```
[edit protocols]
user@R1#set isis interface ge-0/0/1.0 level 2 ipv4-adjacency-segment protected index 108
user@R1#set isis interface ge-0/0/1.0 level 2 ipv4-adjacency-segment unprotected index 110
user@R1#set isis interface ge-0/0/1.0 level 2 ipv6-adjacency-segment protected index 109
user@R1#set isis interface ge-0/0/1.0 level 2 ipv6-adjacency-segment unprotected index 111
user@R1#set isis interface ge-0/0/1.0 level 2 post-convergence-lfa
user@R1#set isis interface ge-0/0/1.0 point-to-point

user@R1#set isis interface ge-0/0/2.0 level 2 ipv4-adjacency-segment protected index 104
user@R1#set isis interface ge-0/0/2.0 level 2 ipv4-adjacency-segment unprotected index 106
user@R1#set isis interface ge-0/0/2.0 level 2 ipv6-adjacency-segment protected index 105
user@R1#set isis interface ge-0/0/2.0 level 2 ipv6-adjacency-segment unprotected index 107
user@R1#set isis interface ge-0/0/2.0 level 2 post-convergence-lfa
user@R1#set isis interface ge-0/0/2.0 point-to-point

user@R1#set isis interface ge-0/0/3.0 level 2 ipv4-adjacency-segment protected index 100
user@R1#set isis interface ge-0/0/3.0 level 2 ipv4-adjacency-segment unprotected index 102
user@R1#set isis interface ge-0/0/3.0 level 2 ipv6-adjacency-segment protected index 101
user@R1#set isis interface ge-0/0/3.0 level 2 ipv6-adjacency-segment unprotected index 103
user@R1#set isis interface ge-0/0/3.0 level 2 post-convergence-lfa
user@R1#set isis interface ge-0/0/3.0 point-to-point

user@R1#set isis interface lo0.0 passive

user@R1#set isis source-packet-routing srgb start-label 80000
```

```
user@R1#set isis source-packet-routing srgb index-range 50000
user@R1#set isis source-packet-routing node-segment ipv4-index 5001
user@R1#set isis source-packet-routing node-segment ipv6-index 5501
user@R1#set isis level 1 disable
```

5. Configure TI-LFA to enable protection against link and node failures. SR using TI-LFA provides faster restoration of network connectivity by routing the traffic instantly to a backup or an alternate path if the primary path fails or becomes unavailable.

```
[edit protocols]
user@R1#set isis backup-spf-options use-post-convergence-lfa
user@R1#set isis backup-spf-options use-source-packet-routing
```

6. Configure ISIS traffic engineering parameters.

```
[edit protocols]
user@R1#set isis traffic-engineering l3-unicast-topology
user@R1#set isis traffic-engineering credibility-protocol-preference
```

7. Enable LDP tunneling over SR-TE.

```
[edit protocols]
user@R1#set isis traffic-engineering tunnel-source-protocol spring-te
```

8. Configure MPLS and LDP protocols on the interfaces in the LDP domain to exchange labels in the LDP domain.

```
[edit protocols]
user@R1#set ldp preference 1
user@R1#set ldp interface ge-0/0/3.0
user@R1#set ldp interface lo0.0

user@R1#set mpls interface ge-0/0/1.0
user@R1#set mpls interface ge-0/0/2.0
user@R1#set mpls interface ge-0/0/3.0
user@R1#set mpls interface lo0.0
```

9. Enable targeted LDP session between the edge routers in the LDP domain.

```
[edit protocols]
user@R1#set ldp auto-targeted-session
```

10. Configure a segment list to route the traffic to a specific path.

```
[edit protocols]
user@R1#set source-packet-routing segment-list seg1 inherit-label-nexthops
user@R1#set source-packet-routing segment-list seg1 auto-translate
user@R1#set source-packet-routing segment-list seg1 hop1 ip-address 192.168.4.2
user@R1#set source-packet-routing segment-list seg1 hop2 ip-address 192.168.5.2
user@R1#set source-packet-routing segment-list seg1 hop3 ip-address 192.168.6.2
```

11. Configure SR-TE LSP to the remote edge routers to enable LDP tunneling over SR-TE.

```
[edit protocols]
user@R1#set source-packet-routing source-routing-path sr_static_r5 ldp-tunneling
user@R1#set source-packet-routing source-routing-path sr_static_r5 to 192.168.66.66
user@R1#set source-packet-routing source-routing-path sr_static_r5 binding-sid 1003001
user@R1#set source-packet-routing source-routing-path sr_static_r5 primary seg1
```

Results

From configuration mode, confirm your configuration by entering the `show chassis`, `show interfaces`, `show routing-options`, and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R1#show chassis
network-services enhanced-ip;
```

```
user@R1#show interfaces
ge-0/0/1 {
    description R1-to-R2;
    unit 0 {
        family inet {
            address 10.1.3.1/24;
```

```
        }
        family iso;
        family mpls {
            maximum-labels 8;
        }
    }
}
ge-0/0/2 {
    description R1-to-R3;
    unit 0 {
        family inet {
            address 10.1.4.1/24;
        }
        family iso;
        family mpls {
            maximum-labels 8;
        }
    }
}
ge-0/0/3 {
    description R1-to-PE1;
    unit 0 {
        family inet {
            address 10.1.2.2/24;
        }
        family iso;
        family mpls {
            maximum-labels 8;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 192.168.100.1/32;
        }
        family iso {
            address 49.0004.0192.0168.0001.00;
        }
        family mpls;
```

```
    }
}
```

```
user@R1#show protocols
isis {
    interface ge-0/0/1.0 {
        level 2 {
            ipv4-adjacency-segment {
                protected index 108;
                unprotected index 110;
            }
            ipv6-adjacency-segment {
                protected index 109;
                unprotected index 111;
            }
            post-convergence-lfa;
        }
        point-to-point;
    }
    interface ge-0/0/2.0 {
        level 2 {
            ipv4-adjacency-segment {
                protected index 104;
                unprotected index 106;
            }
            ipv6-adjacency-segment {
                protected index 105;
                unprotected index 107;
            }
            post-convergence-lfa;
        }
        point-to-point;
    }
    interface ge-0/0/3.0 {
        level 2 {
            ipv4-adjacency-segment {
                protected index 100;
                unprotected index 102;
            }
            ipv6-adjacency-segment {
                protected index 101;
```

```
                unprotected index 103;
            }
            post-convergence-lfa;
        }
        point-to-point;
    }
    interface lo0.0 {
        passive;
    }
    source-packet-routing {
        srgb start-label 80000 index-range 50000;
        node-segment {
            ipv4-index 5001;
            ipv6-index 5501;
        }
    }
    level 1 disable;
    backup-spf-options {
        use-post-convergence-lfa;
        use-source-packet-routing;
    }
    traffic-engineering {
        l3-unicast-topology;
        credibility-protocol-preference;
        tunnel-source-protocol {
            spring-te;
        }
    }
}
ldp {
    auto-targeted-session;
    preference 1;
    interface ge-0/0/3.0;
    interface lo0.0;
}
mpls {
    interface ge-0/0/1.0;
    interface ge-0/0/2.0;
    interface ge-0/0/3.0;
    interface lo0.0;
}
source-packet-routing {
    segment-list seg1 {
```

```
        inherit-label-nexthops;
        auto-translate;
        hop1 ip-address 10.1.4.2;
        hop2 ip-address 10.1.5.2;
        hop3 ip-address 10.1.6.2;
    }
    source-routing-path sr_static_r5 {
        ldp-tunneling;
        to 192.168.100.4;
        binding-sid 1003001;
        primary {
            seg1;
        }
    }
}
```

```
user@R1#show routing-options
router-id 192.168.100.1;
```

Verification

IN THIS SECTION

- Verifying LDP Tunneling over SR-TE | **1351**
- Verify LDP Forwarding to the Remote PE Device | **1353**
- Verifying the Advertised Label | **1356**

To confirm that the configuration is working properly, perform the following tasks:

Verifying LDP Tunneling over SR-TE

Purpose

Verify that the LDP over SR-TE tunnel is enabled and the LDP tunnel to the remote edge router is taking the right path.

Action

From operational mode, run the `show spring-traffic-engineering lsp detail` command.

On R1

```
user@R1>show spring-traffic-engineering lsp detail
Name: sr_static_r5
  Tunnel-source: Static configuration
  To: 192.168.100.2
  State: Up
  LDP-tunneling enabled
    Path: seg1
    Outgoing interface: NA
    Auto-translate status: Enabled Auto-translate result: Success
    Compute Status:Disabled , Compute Result:N/A , Compute-Profile Name:N/A
    BFD status: N/A BFD name: N/A
    ERO Valid: true
      SR-ERO hop count: 3
        Hop 1 (Strict):
          NAI: IPv4 Adjacency ID, 0.0.0.0 -> 10.1.4.2
          SID type: 20-bit label, Value: 80104
        Hop 2 (Strict):
          NAI: IPv4 Adjacency ID, 0.0.0.0 -> 10.1.5.2
          SID type: 20-bit label, Value: 80204
        Hop 3 (Strict):
          NAI: IPv4 Adjacency ID, 0.0.0.0 -> 10.1.6.2
          SID type: 20-bit label, Value: 80304


Total displayed LSPs: 1 (Up: 1, Down: 0)
```

On R2

```
user@R2>show spring-traffic-engineering lsp detail

Name: sr_static_r1
  Tunnel-source: Static configuration
  To: 192.168.100.1
  State: Up
  LDP-tunneling enabled
```

```
    Path: seg1
    Outgoing interface: NA
    Auto-translate status: Enabled Auto-translate result: Success
    Compute Status:Disabled , Compute Result:N/A , Compute-Profile Name:N/A
    BFD status: N/A BFD name: N/A
    ERO Valid: true
      SR-ERO hop count: 3
        Hop 1 (Strict):
          NAI: IPv4 Adjacency ID, 0.0.0.0 -> 10.1.6.1
          SID type: 20-bit label, Value: 80504
        Hop 2 (Strict):
          NAI: IPv4 Adjacency ID, 0.0.0.0 -> 10.1.5.1
          SID type: 20-bit label, Value: 80300
        Hop 3 (Strict):
          NAI: IPv4 Adjacency ID, 0.0.0.0 -> 10.1.4.1
          SID type: 20-bit label, Value: 80200


Total displayed LSPs: 1 (Up: 1, Down: 0)
```

Meaning

- On R1, the LDP tunnel is established with the remote edge router **192.168.100.2** in the SR-TE core network. You can also see the SID label values **80104, 80204, 80304** in the output.
- On R2, the LDP tunnel is established with the remote edge router **192.168.100.1** in the SR-TE core network. You can also see the SID label values **80504, 80300, 80200** in the output.

Verify LDP Forwarding to the Remote PE Device

Purpose

Verify that the route to the remote PE router uses LDP forwarding and is tunneled over SR-TE.

Action

From operational mode, run the `show route` *`destination-prefix`* command.

On R1

Verify that the route to the remote PE (**PE2**) router is through LDP over SR-TE tunnel.

```
user@R1>show route 192.168.100.6

inet.0: 24 destinations, 24 routes (24 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.100.6/32   *[IS-IS/18] 5d 13:10:09, metric 20
                    >  to 10.1.3.2 via ge-0/0/1.0

inet.3: 8 destinations, 13 routes (5 active, 0 holddown, 6 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.100.6/32   *[LDP/1] 5d 13:10:09, metric 1
                    >  to 10.1.4.2 via ge-0/0/2.0, Push 16, Push 80304, Push 80204(top)
                       to 10.1.3.2 via ge-0/0/1.0, Push 16, Push 80304, Push 80204, Push 85003,
Push 85004(top)
```

On R2

Verify that the route to the remote PE (**PE1**) router is through LDP over SR-TE tunnel.

```
user@R2>show route 192.168.100.5

inet.0: 24 destinations, 24 routes (24 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.100.5/32   *[IS-IS/18] 5d 13:20:15, metric 20
                    >  to 10.1.3.1 via ge-0/0/1.0

inet.3: 8 destinations, 13 routes (5 active, 0 holddown, 6 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.100.5/32   *[LDP/9] 5d 13:20:15, metric 1
                    >  to 10.1.6.1 via ge-0/0/2.0, Push 16, Push 80200, Push 80300(top)
                       to 10.1.3.1 via ge-0/0/1.0, Push 16, Push 80200, Push 80300, Push 85004,
Push 85003(top)
```

On PE1

Verify that the route to the remote PE (**PE2**) router is through a targeted LDP session to the remote PE.

```
user@PE1>show route 192.168.100.6

inet.0: 22 destinations, 22 routes (22 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.100.6/32   *[IS-IS/18] 1w3d 15:58:20, metric 30
                    >  to 10.1.2.2 via ge-0/0/3.0

inet.3: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.100.6/32   *[LDP/9] 1w0d 16:00:05, metric 1
                    >  to 10.1.2.2 via ge-0/0/3.0, Push 18
```

On PE2

Verify that the route to the remote PE (**PE1**) router is through a targeted LDP session to the remote PE.

```
user@PE2>show route 192.168.100.5

inet.0: 22 destinations, 22 routes (22 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.100.5/32   *[IS-IS/18] 1w3d 15:59:19, metric 30
                    >  to 10.1.7.1 via ge-0/0/3.0

inet.3: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.100.5/32   *[LDP/9] 1w0d 16:01:14, metric 1
                    >  to 10.1.7.1 via ge-0/0/3.0, Push 18
```

Meaning

- On R1, you can see the LDP label as **16** and the SR-TE label stacks as **80304, 80204, 85003, 85004**.
- On R2, you can see the LDP label as **16** and the SR-TE label stacks as **80200, 80300, 85004, 85003**.
- On PE1 and PE2, you can see the LDP label as **18** and **19**, respectively.

Verifying the Advertised Label

Purpose

Verify the labels advertised for the forwarding equivalence class (FEC).

Action

From operational mode, run the `show ldp database` command.

On R1

Verify the labels advertised towards the directly connected PE (PE1) and the labels received from remote edge router (R2).

```
user@R1>show ldp database

Input label database, 192.168.100.1:0--192.168.100.2:0
Labels received: 4
  Label     Prefix
     17      192.168.100.1/32
      3      192.168.100.2/32
     18      192.168.100.5/32
     16      192.168.100.6/32

Output label database, 192.168.100.1:0--192.168.100.2:0
Labels advertised: 4
  Label     Prefix
      3      192.168.100.1/32
     17      192.168.100.2/32
     16      192.168.100.5/32
     18      192.168.100.6/32

Input label database, 192.168.100.1:0--192.168.100.5:0
Labels received: 4
  Label     Prefix
     17      192.168.100.1/32
     18      192.168.100.2/32
      3      192.168.100.5/32
     19      192.168.100.6/32

Output label database, 192.168.100.1:0--192.168.100.5:0
```

```
Labels advertised: 4
  Label     Prefix
      3      192.168.100.1/32
     17      192.168.100.2/32
     16      192.168.100.5/32
     18      192.168.100.6/32
```

On R2

Verify the labels advertised towards the directly connected PE (PE2) and the labels received from remote edge router (R1).

```
user@R2>show ldp database

Input label database, 192.168.100.2:0--192.168.100.1:0
Labels received: 4
  Label     Prefix
      3      192.168.100.1/32
     17      192.168.100.2/32
     16      192.168.100.5/32
     18      192.168.100.6/32

Output label database, 192.168.100.2:0--192.168.100.1:0
Labels advertised: 4
  Label     Prefix
     17      192.168.100.1/32
      3      192.168.100.2/32
     18      192.168.100.5/32
     16      192.168.100.6/32

Input label database, 192.168.100.2:0--192.168.100.6:0
Labels received: 4
  Label     Prefix
     18      192.168.100.1/32
     17      192.168.100.2/32
     19      192.168.100.5/32
      3      192.168.100.6/32

Output label database, 192.168.100.2:0--192.168.100.6:0
Labels advertised: 4
  Label     Prefix
     17      192.168.100.1/32
```

```
       3      192.168.100.2/32
      18      192.168.100.5/32
      16      192.168.100.6/32
```

On PE1

Verify the label for the remote PE (PE2) device's loopback address is advertised by edge device R1 to the local PE (PE1) device.

```
user@PE1>show ldp database

Input label database, 192.168.100.5:0--192.168.100.1:0
Labels received: 4
  Label     Prefix
      3      192.168.100.1/32
     17      192.168.100.2/32
     16      192.168.100.5/32
     18      192.168.100.6/32

Output label database, 192.168.100.5:0--192.168.100.1:0
Labels advertised: 4
  Label     Prefix
     17      192.168.100.1/32
     18      192.168.100.2/32
      3      192.168.100.5/32
     19      192.168.100.6/32
```

On PE2

Verify the label for the remote PE (PE1) device's loopback address is advertised by edge device R2 to the local PE (PE2) device.

```
user@PE2>show ldp database

Input label database, 192.168.100.6:0--192.168.100.2:0
Labels received: 4
  Label     Prefix
     17      192.168.100.1/32
      3      192.168.100.2/32
     18      192.168.100.5/32
     16      192.168.100.6/32
```

```
Output label database, 192.168.100.6:0--192.168.100.2:0
Labels advertised: 4
  Label     Prefix
     18      192.168.100.1/32
     17      192.168.100.2/32
     19      192.168.100.5/32
      3      192.168.100.6/32
```

Meaning

- On R1, you can see label **18** is advertised towards the directly connected PE (PE1) and the label **19** is received from remote edge router (R2).
- On R2, you can see label **17** is advertised towards the directly connected PE (PE2) and the label **19** is received from remote edge router (R1).
- On PE1, you can see label **18** is received from the local edge router (R1).
- On PE2, you can see label **17** is received from the local edge router (R2).

SEE ALSO

vLab Sandbox: Segment Routing - Basic

Label Operations

Figure 79 on page 1360 depicts an LDP LSP being tunneled through an RSVP LSP. (For definitions of label operations, see "MPLS Label Overview" on page 524.) The shaded inner oval represents the RSVP domain, whereas the outer oval depicts the LDP domain. RSVP establishes an LSP through routers B, C, D, and E, with the sequence of labels L3, L4. LDP establishes an LSP through Routers A, B, E, F, and G, with the sequence of labels L1, L2, L5. LDP views the RSVP LSP between Routers B and E as a single hop.

When the packet arrives at Router A, it enters the LSP established by LDP, and a label (L1) is pushed onto the packet. When the packet arrives at Router B, the label (L1) is swapped with another label (L2). Because the packet is entering the traffic-engineered LSP established by RSVP, a second label (L3) is pushed onto the packet.

This outer label (L3) is swapped with a new label (L4) at the intermediate router (C) within the RSVP LSP tunnel, and when the penultimate router (D) is reached, the top label is popped. Router E swaps the label (L2) with a new label (L5), and the penultimate router for the LDP-established LSP (F) pops the last label.

Figure 79: Swap and Push When LDP LSPs Are Tunneled Through RSVP LSPs

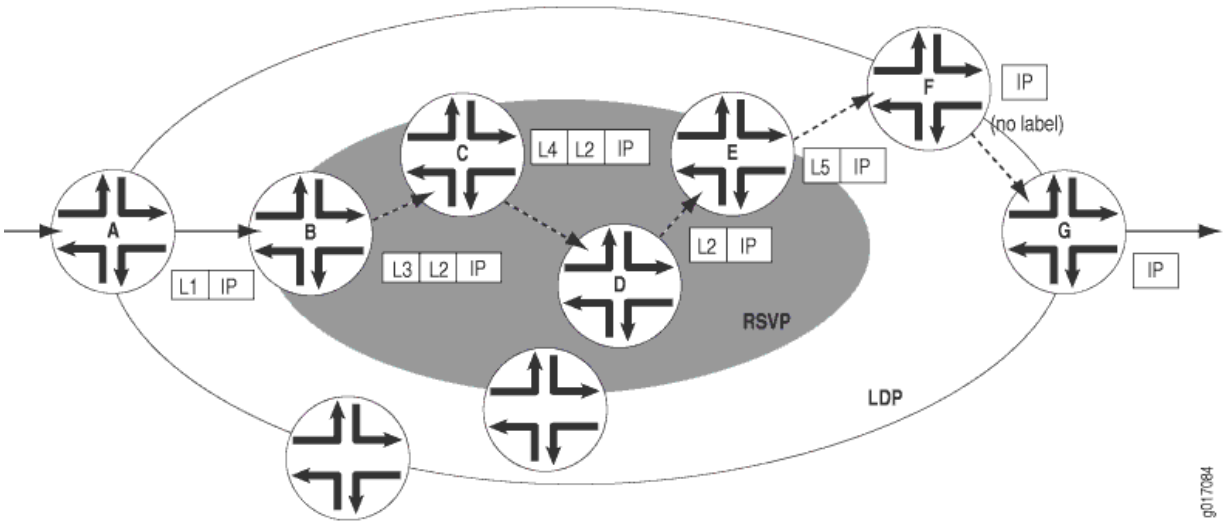
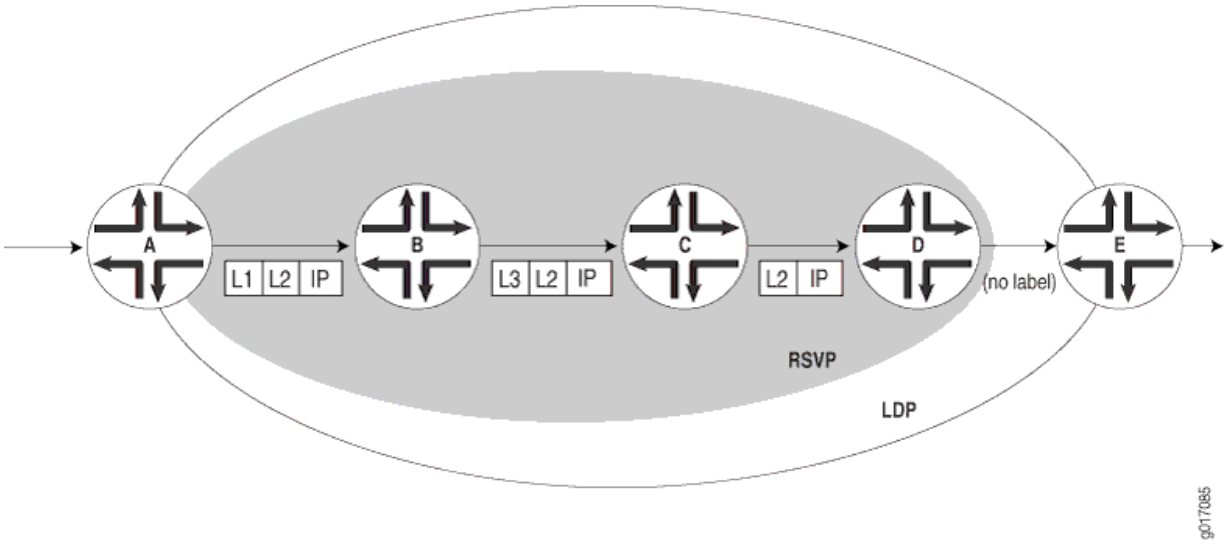


Figure 80 on page 1360 depicts a double push label operation (L1L2). A double push label operation is used when the ingress router (A) for both the LDP LSP and the RSVP LSP tunneled through it is the same device. Note that Router D is the penultimate hop for the LDP-established LSP, so L2 is popped from the packet by Router D.

Figure 80: Double Push When LDP LSPs Are Tunneled Through RSVP LSPs



LDP Session Protection

LDP session protection is based on the LDP targeted hello functionality defined in RFC 5036, *LDP Specification*, and is supported by the Junos OS as well as the LDP implementations of most other

vendors. It involves sending unicast User Datagram Protocol (UDP) hello packets to a remote neighbor address and receiving similar packets from the neighbor router.

If you configure LDP session protection on a router, the LDP sessions are maintained as follows:

1. An LDP session is established between a router and a remote neighboring router.
2. If all of the direct links between the routers go down, the LDP session remains up so long as there is IP connectivity between the routers based on another connection over the network.
3. When the direct link between the routers is reestablished, the LDP session is not restarted. The routers simply exchange LDP hellos with each other over the direct link. They can then begin forwarding LDP-signaled MPLS packets using the original LDP session.

By default, LDP targeted hellos are set to the remote neighbor so long as the LDP session is up, even if there are no more link neighbors to that router. You can also specify the duration you would like to maintain the remote neighbor connection in the absence of link neighbors. When the last link neighbor for a session goes down, the Junos OS starts an LDP session protection timer. If this timer expires before any of the link neighbors come back up, the remote neighbor connection is taken down and the LDP session is terminated. If you configure a different value for the timer while it is currently running, the Junos OS updates the timer to the specified value without disrupting the current state of the LDP session.

LDP Native IPv6 Support Overview

IPv6 connectivity often relies on tunneling IPv6 over an IPv4 MPLS core with IPv4-signaled MPLS label-switched paths (LSPs). This requires the IPv4-signaled LSPs to be configured statically or established dynamically by IPv6 provider edge routers. Because of the growing demand of IPv6, it has become imperative to deploy an IPv6 MPLS core with an IPv6-signaled LSP to provide IPv6 connectivity. In Junos OS, LDP is supported in an IPv6 network only, and in an IPv6/IPv4 dual-stack network as described in *RFC 7552*. Apart from providing a single session for both IPv4 and IPv6 networks, Junos OS LDP supports separate IPv4 sessions for IPv4 only, and IPv6 sessions for IPv6 only.

You can configure the address family as `inet` for IPv4 or `inet6` for IPv6, or both. If the family address is not configured, then the default address of family inet is enabled. When both IPv4 and IPv6 are configured, you can use the `transport-preference` statement to configure the prefered transport to be either `IPv4` or `IPv6`. Based on the preference, LDP attempts to establish a TCP connection using IPv4 or IPv6. By default, IPv6 is selected. The `dual-transport` statement allows Junos OS LDP to establish the TCP connection over IPv4 with IPv4 neighbors, and over IPv6 with IPv6 neighbors as a single-stack LSR. The `inet-lsr-id` and `inet6-lsr-id` IDs are the two LSR IDs that have to be configured to establish an LDP session over IPv4 and IPv6 TCP transport. These two IDs should be non-zero and must be configured with different values.

Longest Match Support for LDP Overview

LDP is often used to establish MPLS label-switched paths (LSPs) throughout a complete network domain using an IGP such as OSPF or IS-IS. In such a network, all links in the domain have IGP adjacencies as well as LDP adjacencies. LDP establishes the LSPs on the shortest path to a destination as determined by IGP. In Junos OS, the LDP implementation does an exact match lookup on the IP address of the forwarding equivalence class (FEC) in the routing information base (RIB) or IGP routes for label mapping. This exact mapping requires MPLS end-to-end LDP endpoint IP addresses to be configured in all the label edge routers (LERs). This defeats the purpose of IP hierarchical design or default routing in access devices. Configuring `longest-match` allows LDP to set up LSP based on the routes aggregated or summarized across OSPF areas or IS-IS levels in the inter-domain.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
22.4R1	Starting in Junos OS and Junos OS Evolved Release 22.4R1, you can tunnel LDP LSPs over Segment Routing Traffic Engineering (SR-TE) in OSPF networks.
20.3R1	Starting in Junos OS Release 20.3R1, support for MPLS to provide LDP signaling protocol configuration with the control plane functionality.
15.1	Beginning with Junos OS Release 15.1, multi-instance support is extended to LDP over RSVP tunneling for a virtual router routing instance.

RELATED DOCUMENTATION

Basic MPLS Configuration | 46

LDP Configuration

IN THIS SECTION

- Minimum LDP Configuration | 1364
- Enabling and Disabling LDP | 1364

- Configuring the LDP Timer for Hello Messages | **1365**
- Configuring the Delay Before LDP Neighbors Are Considered Down | **1366**
- Enabling Strict Targeted Hello Messages for LDP | **1368**
- Configuring the Interval for LDP Keepalive Messages | **1369**
- Configuring the LDP Keepalive Timeout | **1369**
- Configuring Longest Match for LDP | **1369**
- Example: Configuring Longest Match for LDP | **1370**
- Configuring LDP Route Preferences | **1391**
- LDP Graceful Restart | **1391**
- Configuring LDP Graceful Restart | **1392**
- Filtering Inbound LDP Label Bindings | **1395**
- Filtering Outbound LDP Label Bindings | **1398**
- Specifying the Transport Address Used by LDP | **1400**
- Control Transport Address Used for Targeted-LDP Session | **1401**
- Configuring the Prefixes Advertised into LDP from the Routing Table | **1404**
- Configuring FEC Deaggregation | **1405**
- Configuring Policers for LDP FECs | **1406**
- Configuring LDP IPv4 FEC Filtering | **1407**
- Configuring BFD for LDP LSPs | **1408**
- Configuring ECMP-Aware BFD for LDP LSPs | **1411**
- Configuring a Failure Action for the BFD Session on an LDP LSP | **1412**
- Configuring the Holddown Interval for the BFD Session | **1413**
- Configuring LDP Link Protection | **1413**
- Example: Configuring LDP Link Protection | **1415**
- Understanding Multicast-Only Fast Reroute | **1448**
- Configuring Multicast-Only Fast Reroute | **1457**
- Example: Configuring Multicast-Only Fast Reroute in a Multipoint LDP Domain | **1460**
- Example: Configuring LDP Downstream on Demand | **1483**
- Configuring LDP Native IPv6 Support | **1490**
- Example: Configuring LDP Native IPv6 Support | **1491**
- Example: Configuring Multipoint LDP In-Band Signaling for Point-to-Multipoint LSPs | **1510**
- Mapping Client and Server for Segment Routing to LDP Interoperability | **1549**

- Miscellaneous LDP Properties | **1555**
- Configuring LDP LSP Traceroute | **1563**
- Collecting LDP Statistics | **1564**
- Tracing LDP Protocol Traffic | **1567**

Minimum LDP Configuration

To enable LDP with minimal configuration:

1. Enable all relevant interfaces under family MPLS. In the case of directed LDP, the loopback interface needs to be enabled with family MPLS.
2. (Optional) Configure the relevant interfaces under the `[edit protocol mpls]` hierarchy level.
3. Enable LDP on a single interface, include the `ldp` statement and specify the interface using the `interface` statement.

This is the minimum LDP configuration. All other LDP configuration statements are optional.

```
ldp {
    interface interface-name;
}
```

To enable LDP on all interfaces, specify `all` for *`interface-name`*.

For a list of hierarchy levels at which you can include these statements, see the statement summary sections.

Enabling and Disabling LDP

LDP is routing-instance-aware. To enable LDP on a specific interface, include the following statements:

```
ldp {
    interface interface-name;
}
```

For a list of hierarchy levels at which you can include these statements, see the statement summary sections.

To enable LDP on all interfaces, specify `all` for *`interface-name`*.

If you have configured interface properties on a group of interfaces and want to disable LDP on one of the interfaces, include the `interface` statement with the `disable` option:

```
interface interface-name {
    disable;
}
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section.

Configuring the LDP Timer for Hello Messages

IN THIS SECTION

- Configuring the LDP Timer for Link Hello Messages | **1366**
- Configuring the LDP Timer for Targeted Hello Messages | **1366**

LDP hello messages enable LDP nodes to discover one another and to detect the failure of a neighbor or the link to the neighbor. Hello messages are sent periodically on all interfaces where LDP is enabled.

There are two types of LDP hello messages:

- Link hello messages—Sent through the LDP interface as UDP packets addressed to the LDP discovery port. Receipt of an LDP link hello message on an interface identifies an adjacency with the LDP peer router.
- Targeted hello messages—Sent as UDP packets addressed to the LDP discovery port at a specific address. Targeted hello messages are used to support LDP sessions between routers that are not directly connected. A targeted router determines whether to respond or ignore a targeted hello message. A targeted router that chooses to respond does so by periodically sending targeted hello messages back to the initiating router.

By default, LDP sends hello messages every 5 seconds for link hello messages and every 15 seconds for targeted hello messages. You can configure the LDP timer to alter how often both types of hello messages are sent. However, you cannot configure a time for the LDP timer that is greater than the LDP hold time. For more information, see "Configuring the Delay Before LDP Neighbors Are Considered Down" on page 1366.

Configuring the LDP Timer for Link Hello Messages

To modify how often LDP sends link hello messages, specify a new link hello message interval for the LDP timer using the `hello-interval` statement:

```
hello-interval seconds;
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

Configuring the LDP Timer for Targeted Hello Messages

To modify how often LDP sends targeted hello messages, specify a new targeted hello message interval for the LDP timer by configuring the `hello-interval` statement as an option for the `targeted-hello` statement:

```
targeted-hello {
    hello-interval seconds;
}
```

For a list of hierarchy levels at which you can include these statements, see the statement summary sections for these statements.

Configuring the Delay Before LDP Neighbors Are Considered Down

IN THIS SECTION

- Configuring the LDP Hold Time for Link Hello Messages | **1367**
- Configuring the LDP Hold Time for Targeted Hello Messages | **1368**

The hold time determines how long an LDP node should wait for a hello message before declaring a neighbor to be down. This value is sent as part of a hello message so that each LDP node tells its neighbors how long to wait. The values sent by each neighbor do not have to match.

The hold time should normally be at least three times the hello interval. The default is 15 seconds for link hello messages and 45 seconds for targeted hello messages. However, it is possible to configure an LDP hold time that is close to the value for the hello interval.

NOTE: By configuring an LDP hold time close to the hello interval (less than three times the hello interval), LDP neighbor failures might be detected more quickly. However, this also increases the possibility that the router might declare an LDP neighbor down that is still functioning normally. For more information, see "Configuring the LDP Timer for Hello Messages" on page 1365.

The LDP hold time is also negotiated automatically between LDP peers. When two LDP peers advertise different LDP hold times to one another, the smaller value is used. If an LDP peer router advertises a shorter hold time than the value you have configured, the peer router's advertised hold time is used. This negotiation can affect the LDP keepalive interval as well.

If the local LDP hold time is not shortened during LDP peer negotiation, the user-configured keepalive interval is left unchanged. However, if the local hold time is reduced during peer negotiation, the keepalive interval is recalculated. If the LDP hold time has been reduced during peer negotiation, the keepalive interval is reduced to one-third of the new hold time value. For example, if the new hold-time value is 45 seconds, the keepalive interval is set to 15 seconds.

This automated keepalive interval calculation can cause different keepalive intervals to be configured on each peer router. This enables the routers to be flexible in how often they send keepalive messages, because the LDP peer negotiation ensures they are sent more frequently than the LDP hold time.

When you reconfigure the hold-time interval, changes do not take effect until after the session is reset. The hold time is negotiated when the LDP peering session is initiated and cannot be renegotiated as long as the session is up (required by RFC 5036, *LDP Specification*). To manually force the LDP session to reset, issue the `clear ldp session` command.

Configuring the LDP Hold Time for Link Hello Messages

To modify how long an LDP node should wait for a link hello message before declaring the neighbor down, specify a new time in seconds using the `hold-time` statement:

```
hold-time seconds;
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

Configuring the LDP Hold Time for Targeted Hello Messages

To modify how long an LDP node should wait for a targeted hello message before declaring the neighbor down, specify a new time in seconds using the `hold-time` statement as an option for the `targeted-hello` statement:

```
targeted-hello {
    hold-time seconds;
}
```

For a list of hierarchy levels at which you can include these statements, see the statement summary sections for these statements.

Enabling Strict Targeted Hello Messages for LDP

Use strict targeted hello messages to prevent LDP sessions from being established with remote neighbors that have not been specifically configured. If you configure the `strict-targeted-hellos` statement, an LDP peer does not respond to targeted hello messages coming from a source that is not one of its configured remote neighbors. Configured remote neighbors can include:

- Endpoints of RSVP tunnels for which LDP tunneling is configured
- Layer 2 circuit neighbors

If an unconfigured neighbor sends a hello message, the LDP peer ignores the message and logs an error (with the `error` trace flag) indicating the source. For example, if the LDP peer received a targeted hello from the Internet address 10.0.0.1 and no neighbor with this address is specifically configured, the following message is printed to the LDP log file:

```
LDP: Ignoring targeted hello from 10.0.0.1
```

To enable strict targeted hello messages, include the `strict-targeted-hellos` statement:

```
strict-targeted-hellos;
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

Configuring the Interval for LDP Keepalive Messages

The keepalive interval determines how often a message is sent over the session to ensure that the keepalive timeout is not exceeded. If no other LDP traffic is sent over the session in this much time, a keepalive message is sent. The default is 10 seconds. The minimum value is 1 second.

The value configured for the keepalive interval can be altered during LDP session negotiation if the value configured for the LDP hold time on the peer router is lower than the value configured locally. For more information, see "Configuring the Delay Before LDP Neighbors Are Considered Down" on page 1366.

To modify the keepalive interval, include the `keepalive-interval` statement:

```
keepalive-interval seconds;
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

Configuring the LDP Keepalive Timeout

After an LDP session is established, messages must be exchanged periodically to ensure that the session is still working. The keepalive timeout defines the amount of time that the neighbor LDP node waits before deciding that the session has failed. This value is usually set to at least three times the keepalive interval. The default is 30 seconds.

To modify the keepalive interval, include the `keepalive-timeout` statement:

```
keepalive-timeout seconds;
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

The value configured for the `keepalive-timeout` statement is displayed as the hold time when you issue the `show ldp session detail` command.

Configuring Longest Match for LDP

In order to allow LDP to learn the routes aggregated or summarized across OSPF areas or ISIS levels in inter -domain, Junos OS allows you to configure longest match for LDP based on *RFC5283*.

Before you configure longest match for LDP, you must do the following:

1. Configure the device interfaces.
2. Configure the MPLS protocol.

3. Configure the OSPF protocol.

To configure longest match for LDP, you must do the following:

1. Configure longest match for the LDP protocol.

```
[edit protocols ldp]
user@host# set longest-match
```

2. Configure the LDP protocol on the interface.

```
[edit protocols ldp]
user@host# set interface interface-name
```

For example, to configure the interfaces:

```
[edit protocols ldp]
user@host# set interface ge-0/0/2.0
user@host# set interface lo0.0
```

Example: Configuring Longest Match for LDP

IN THIS SECTION

- Requirements | 1370
- Overview | 1371
- Configuration | 1372
- Verification | 1379

This example shows how to configure longest match for LDP based on *RFC5283*. This allows LDP to learn the routes aggregated or summarized across OSPF areas or ISIS levels in inter-domain.. The longest match policy provides per prefix granularity.

Requirements

This example uses the following hardware and software components:

- Six MX Series routers with OSPF protocol, and LDP enabled on the connected interfaces.
- Junos OS Release 16.1 or later running on all devices.

Before you begin:

- Configure the device interfaces.
- Configure OSPF.

Overview

IN THIS SECTION

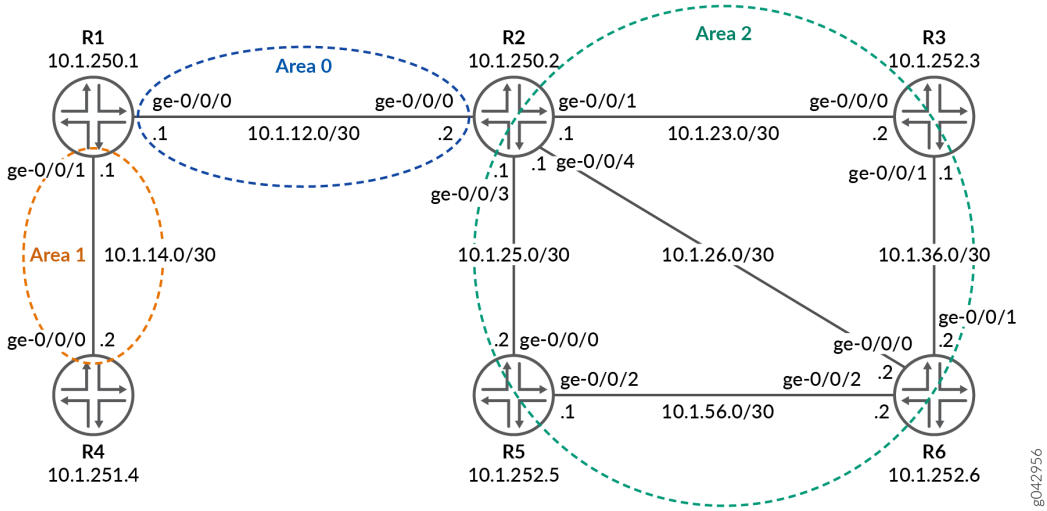
- Topology | **1371**

LDP is often used to establish MPLS label-switched paths (LSPs) throughout a complete network domain using an IGP such as OSPF or IS-IS. In such a network, all links in the domain have IGP adjacencies as well as LDP adjacencies. LDP establishes the LSPs on the shortest path to a destination as determined by IP forwarding. In Junos OS, the LDP implementation does an exact match lookup on the IP address of the FEC in the RIB or IGP routes for label mapping. This exact mapping requires MPLS end-to-end LDP endpoint IP addresses to be configured in all the LERs. This defeats the purpose of IP hierarchical design or default routing in access devices. Configuring `longest-match` helps to overcome this by suppressing the exact match behaviour and setup LSP based on the longest matching route on per-prefix basis.

Topology

In the topology, Figure 81 on page 1372shows the longest match for LDP is configured on Device R0 .

Figure 81: Example Longest Match for LDP



Configuration

IN THIS SECTION

- CLI Quick Configuration | **1372**
- Configuring Device R0 | **1376**
- Results | **1377**

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter `commit` from configuration mode.

R0

```
set interfaces ge-0/0/0 unit 0 family inet address 22.22.22.1/24
set interfaces ge-0/0/1 unit 0 family inet address 15.15.15.1/24
set interfaces ge-0/0/2 unit 0 family inet address 11.11.11.1/24
set interfaces ge-0/0/2 unit 0 family iso
```

```
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.112.1/32 primary
set interfaces lo0 unit 0 family inet address 10.255.112.1/32 preferred
set interfaces lo0 unit 0 family iso address 49.0002.0192.0168.0001.00
set routing-options router-id 10.255.112.1
set protocols mpls interface ge-0/0/2.0
set protocols ospf area 0.0.0.1 interface ge-0/0/2.0
set protocols ospf area 0.0.0.1 interface lo0.0 passive
set protocols ldp longest-match
set protocols ldp interface ge-0/0/2.0
set protocols ldp interface lo0.0
```

R1

```
set interfaces ge-0/0/0 unit 0 family inet address 11.11.11.2/24
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 12.12.12.1/24
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.112.2/32 primary
set interfaces lo0 unit 0 family inet address 10.255.112.2/32 preferred
set interfaces lo0 unit 0 family iso address 49.0002.0192.0168.0002.00
set routing-options router-id 10.255.112.2
set protocols mpls interface ge-0/0/0.0
set protocols mpls interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.1 interface ge-0/0/0.0
set protocols ldp longest-match
set protocols ldp interface ge-0/0/0.0
set protocols ldp interface ge-0/0/1.0
set protocols ldp interface lo0.0
```

R2

```
set interfaces ge-0/0/0 unit 0 family inet address 24.24.24.1/24
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 12.12.12.2/24
set interfaces ge-0/0/1 unit 0 family iso
```

```
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 23.23.23.1/24
set interfaces ge-0/0/2 unit 0 family iso
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 family inet address 22.22.22.2/24
set interfaces ge-0/0/4 unit 0 family inet address 25.25.25.1/24
set interfaces ge-0/0/4 unit 0 family iso
set interfaces ge-0/0/4 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.111.4/32 primary
set interfaces lo0 unit 0 family inet address 10.255.111.4/32 preferred
set interfaces lo0 unit 0 family iso address 49.0003.0192.0168.0003.00
set routing-options router-id 10.255.111.4
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols mpls interface ge-0/0/0.0
set protocols mpls interface ge-0/0/4.0
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.2 area-range 10.255.111.0/24
set protocols ospf area 0.0.0.2 interface ge-0/0/2.0
set protocols ospf area 0.0.0.2 interface ge-0/0/0.0
set protocols ospf area 0.0.0.2 interface ge-0/0/4.0
set protocols ldp interface ge-0/0/0.0
set protocols ldp interface ge-0/0/1.0
set protocols ldp interface ge-0/0/2.0
set protocols ldp interface ge-0/0/4.0
set protocols ldp interface lo0.0
```

R3

```
set interfaces ge-0/0/0 unit 0 family inet address 35.35.35.1/24
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 23.23.23.2/24
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 34.34.34.1/24
set interfaces ge-0/0/2 unit 0 family iso
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.111.1/32 primary
set interfaces lo0 unit 0 family inet address 10.255.111.1/32 preferred
set interfaces lo0 unit 0 family iso address 49.0003.0192.0168.0004.00
```

```
set routing-options router-id 10.255.111.1
set protocols mpls interface ge-0/0/1.0
set protocols ospf area 0.0.0.2 interface ge-0/0/1.0
set protocols ospf area 0.0.0.2 interface fxp0.0 disable
set protocols ospf area 0.0.0.2 interface lo0.0 passive
set protocols ldp interface ge-0/0/1.0
set protocols ldp interface lo0.0
```

R4

```
set interfaces ge-0/0/0 unit 0 family inet address 45.45.45.1/24
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 24.24.24.2/24
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 34.34.34.2/24
set interfaces ge-0/0/2 unit 0 family iso
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.111.2/32 primary
set interfaces lo0 unit 0 family inet address 10.255.111.2/32 preferred
set interfaces lo0 unit 0 family iso address 49.0003.0192.0168.0005.00
set routing-options router-id 10.255.111.2
set protocols mpls interface ge-0/0/1.0
set protocols ospf area 0.0.0.2 interface ge-0/0/1.0
set protocols ospf area 0.0.0.2 interface fxp0.0 disable
set protocols ospf area 0.0.0.2 interface lo0.0 passive
set protocols ldp interface ge-0/0/1.0
set protocols ldp interface lo0.0
```

R5

```
set interfaces ge-0/0/0 unit 0 family inet address 25.25.25.2/24
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 15.15.15.2/24
set interfaces ge-0/0/2 unit 0 family inet address 35.35.35.2/24
set interfaces ge-0/0/3 unit 0 family inet address 45.45.45.2/24
set interfaces ge-0/0/3 unit 0 family iso
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.111.3/32 primary
```

```
set interfaces lo0 unit 0 family inet address 10.255.111.3/32 preferred
set interfaces lo0 unit 0 family iso address 49.0003.0192.0168.0006.00
set routing-options router-id 10.255.111.3
set protocols mpls interface ge-0/0/0.0
set protocols ospf area 0.0.0.2 interface ge-0/0/0.0
set protocols ospf area 0.0.0.2 interface fxp0.0 disable
set protocols ospf area 0.0.0.2 interface lo0.0 passive
set protocols ldp interface ge-0/0/0.0
set protocols ldp interface lo0.0
```

Configuring Device R0

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure Device R0:

1. Configure the interfaces.

```
[edit interfaces]
set ge-0/0/0 unit 0 family inet address 22.22.22.1/24
set ge-0/0/1 unit 0 family inet address 15.15.15.1/24
set ge-0/0/2 unit 0 family inet address 11.11.11.1/24
set ge-0/0/2 unit 0 family iso
set ge-0/0/2 unit 0 family mpls
```

2. Assign the loopback addresses to the device.

```
[edit interfaces lo0 unit 0 family]
set inet address 10.255.112.1/32 primary
set inet address 10.255.112.1/32 preferred
set iso address 49.0002.0192.0168.0001.00
```

3. Configure the router ID.

```
[edit routing-options]
set router-id 10.255.112.1
```

4. Configure the MPLS protocol on the interface.

```
[edit protocols mpls]
set interface ge-0/0/2.0
```

5. Configure the OSPF protocol on the interface.

```
[edit protocols ospf]
set area 0.0.0.1 interface ge-0/0/2.0
set area 0.0.0.1 interface lo0.0 passive
```

6. Configure longest match for the LDP protocol.

```
[edit protocols ldp]
set longest-match
```

7. Configure the LDP protocol on the interface.

```
[edit protocols ldp]
set interface ge-0/0/2.0
set interface lo0.0
```

Results

From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R0# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
```

```
            address 22.22.22.1/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 15.15.15.1/24;
        }
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 11.11.11.1/24;
        }
        family iso;
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.255.112.1/32 {
                primary;
                preferred;
            }
        }
        family iso {
            address 49.0002.0192.0168.0001.00;
        }
    }
}
```

```
user@R0# show protocols
mpls {
    interface ge-0/0/2.0;
}
ospf {
    area 0.0.0.1 {
        interface ge-0/0/2.0;
```

```
        interface lo0.0 {
            passive;
        }
    }
}
ldp {
    longest-match;
    interface ge-0/0/2.0;
    interface lo0.0;
}
```

```
user@R0# show routing-options
router-id 10.255.112.1;
```

If you are done configuring the device, enter `commit` from the configuration mode.

Verification

IN THIS SECTION

- Verifying the Routes | **1379**
- Verifying LDP Overview Information | **1384**
- Verify the LDP Entries in the Internal Topology Table | **1386**
- Verify Only FEC Information of LDP Route | **1388**
- Verify FEC and Shadow Routes of LDP | **1389**

Confirm that the configuration is working properly.

Verifying the Routes

Purpose

Verify that the expected routes are learned.

Action

On Device R0, from operational mode, run the `show route` command to display the routes in the routing table.

```
user@R0> show route

inet.0: 62 destinations, 62 routes (62 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.4.0.0/16        *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
10.5.0.0/16        *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
10.6.128.0/17      *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
10.9.0.0/16        *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
10.10.0.0/16       *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
10.13.4.0/23       *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
10.13.10.0/23      *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
10.82.0.0/15       *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
10.84.0.0/16       *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
10.85.12.0/22      *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
10.92.0.0/16       *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
10.92.16.0/20      *[Direct/0] 10:08:01
                    > via fxp0.0
10.92.20.175/32    *[Local/0] 10:08:01
                       Local via fxp0.0
10.94.0.0/16       *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
10.99.0.0/16       *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
10.102.0.0/16      *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
```

```
10.150.0.0/16      *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
10.155.0.0/16      *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
10.157.64.0/19     *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
10.160.0.0/16      *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
10.204.0.0/16      *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
10.205.0.0/16      *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
10.206.0.0/16      *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
10.207.0.0/16      *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
10.209.0.0/16      *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
10.212.0.0/16      *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
10.213.0.0/16      *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
10.214.0.0/16      *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
10.215.0.0/16      *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
10.216.0.0/16      *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
10.218.13.0/24     *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
10.218.14.0/24     *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
10.218.16.0/20     *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
10.218.32.0/20     *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
10.227.0.0/16      *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
10.255.111.0/24    *[OSPF/10] 09:52:14, metric 3
                    > to 11.11.11.2 via ge-0/0/2.0
10.255.111.4/32    *[OSPF/10] 09:54:10, metric 2
                    > to 11.11.11.2 via ge-0/0/2.0
10.255.112.1/32    *[Direct/0] 09:55:05
```

```
                    > via lo0.0
10.255.112.2/32     *[OSPF/10] 09:54:18, metric 1
                    > to 11.11.11.2 via ge-0/0/2.0
11.11.11.0/24       *[Direct/0] 09:55:05
                    > via ge-0/0/2.0
11.11.11.1/32       *[Local/0] 09:55:05
                      Local via ge-0/0/2.0
12.12.12.0/24       *[OSPF/10] 09:54:18, metric 2
                    > to 11.11.11.2 via ge-0/0/2.0
15.15.15.0/24       *[Direct/0] 09:55:05
                    > via ge-0/0/1.0
15.15.15.1/32       *[Local/0] 09:55:05
                      Local via ge-0/0/1.0
22.22.22.0/24       *[Direct/0] 09:55:05
                    > via ge-0/0/0.0
22.22.22.1/32       *[Local/0] 09:55:05
                      Local via ge-0/0/0.0
23.23.23.0/24       *[OSPF/10] 09:54:10, metric 3
                    > to 11.11.11.2 via ge-0/0/2.0
24.24.24.0/24       *[OSPF/10] 09:54:10, metric 3
                    > to 11.11.11.2 via ge-0/0/2.0
25.25.25.0/24       *[OSPF/10] 09:54:10, metric 3
                    > to 11.11.11.2 via ge-0/0/2.0
128.92.17.45/32     *[OSPF/10] 09:54:05, metric 3
                    > to 11.11.11.2 via ge-0/0/2.0
128.92.20.175/32    *[Direct/0] 10:08:01
                    > via lo0.0
128.92.21.186/32    *[OSPF/10] 09:54:10, metric 3
                    > to 11.11.11.2 via ge-0/0/2.0
128.92.25.135/32    *[OSPF/10] 09:54:10, metric 3
                    > to 11.11.11.2 via ge-0/0/2.0
128.92.27.91/32     *[OSPF/10] 09:54:18, metric 1
                    > to 11.11.11.2 via ge-0/0/2.0
128.92.28.70/32     *[OSPF/10] 09:54:10, metric 2
                    > to 11.11.11.2 via ge-0/0/2.0
172.16.0.0/12       *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
192.168.0.0/16      *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
192.168.102.0/23    *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
207.17.136.0/24     *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
```

```
207.17.136.192/32  *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
207.17.137.0/24    *[Static/5] 10:08:01
                    > to 10.92.31.254 via fxp0.0
224.0.0.5/32       *[OSPF/10] 09:55:05, metric 1
                      MultiRecv

inet.3: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.255.111.1/32    *[LDP/9] 09:41:03, metric 3
                    > to 11.11.11.2 via ge-0/0/2.0, Push 300128
10.255.111.2/32    *[LDP/9] 09:41:03, metric 3
                    > to 11.11.11.2 via ge-0/0/2.0, Push 300144
10.255.111.3/32    *[LDP/9] 09:41:03, metric 3
                    > to 11.11.11.2 via ge-0/0/2.0, Push 300160
10.255.111.4/32    *[LDP/9] 09:54:10, metric 2, tag 0
                    > to 11.11.11.2 via ge-0/0/2.0, Push 300000
10.255.112.2/32    *[LDP/9] 09:54:48, metric 1, tag 0
                    > to 11.11.11.2 via ge-0/0/2.0

iso.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

47.0005.80ff.f800.0000.0108.0001.1280.9202.0175/152
                   *[Direct/0] 10:08:01
                    > via lo0.0
49.0002.0192.0168.0001/72
                   *[Direct/0] 09:55:05
                    > via lo0.0

mpls.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 09:55:05, metric 1
                      Receive
1                  *[MPLS/0] 09:55:05, metric 1
                      Receive
2                  *[MPLS/0] 09:55:05, metric 1
                      Receive
13                 *[MPLS/0] 09:55:05, metric 1
                      Receive
300064             *[LDP/9] 09:54:48, metric 1
```

```
                    > to 11.11.11.2 via ge-0/0/2.0, Pop
300064(S=0)        *[LDP/9] 09:54:48, metric 1
                    > to 11.11.11.2 via ge-0/0/2.0, Pop
300112             *[LDP/9] 09:54:10, metric 2, tag 0
                    > to 11.11.11.2 via ge-0/0/2.0, Swap 300000
300192             *[LDP/9] 09:41:03, metric 3
                    > to 11.11.11.2 via ge-0/0/2.0, Swap 300128
300208             *[LDP/9] 09:41:03, metric 3
                    > to 11.11.11.2 via ge-0/0/2.0, Swap 300144
300224             *[LDP/9] 09:41:03, metric 3
                    > to 11.11.11.2 via ge-0/0/2.0, Swap 300160

inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

abcd::128:92:20:175/128
                   *[Direct/0] 10:08:01
                    > via lo0.0
fe80::5668:a50f:fcc1:1f9c/128
                   *[Direct/0] 10:08:01
                    > via lo0.0
```

Meaning

The output shows all the routes in the routing table of Device R0.

Verifying LDP Overview Information

Purpose

Display LDP overview information.

Action

On Device R0, from operational mode, run the `show ldp overview` command to display the overview of the LDP.

```
user@R0> show ldp overview
Instance: master
  Reference count: 2
  Router ID: 10.255.112.1
```

```
Message id: 8
Configuration sequence: 6
Deaggregate: disabled
Explicit null: disabled
IPv6 tunneling: disabled
Strict targeted hellos: disabled
Loopback if added: yes
Route preference: 9
Unicast transit LSP chaining: disabled
P2MP transit LSP chaining: disabled
Transit LSP statistics based on route statistics: disabled
LDP route acknowledgement: enabled
LDP mtu discovery: disabled
Longest Match: enabled
Capabilities enabled: none
Egress FEC capabilities enabled: entropy-label-capability
Downstream unsolicited Sessions:
  Operational: 1
  Retention: liberal
  Control: ordered
Auto targeted sessions:
  Auto targeted: disabled
Timers:
  Keepalive interval: 10, Keepalive timeout: 30
  Link hello interval: 5, Link hello hold time: 15
  Targeted hello interval: 15, Targeted hello hold time: 45
  Label withdraw delay: 60, Make before break timeout: 30
  Make before break switchover delay: 3
  Link protection timeout: 120
Graceful restart:
  Restart: disabled, Helper: enabled, Restart in process: false
  Reconnect time: 60000, Max neighbor reconnect time: 120000
  Recovery time: 160000, Max neighbor recovery time: 240000
Traffic Engineering:
  Bgp igp: disabled
  Both ribs: disabled
  Mpls forwarding: disabled
IGP:
  Tracking igp metric: disabled
  Sync session up delay: 10
Session protection:
  Session protection: disabled
  Session protection timeout: 0
```

```
  Interface addresses advertising:
    11.11.11.1
    10.255.112.1
    128.92.20.175
  Label allocation:
    Current number of LDP labels allocated: 5
    Total number of LDP labels allocated: 11
    Total number of LDP labels freed: 6
    Total number of LDP label allocation failure: 0
    Current number of labels allocated by all protocols: 5
```

Meaning

The output displays the LDP overview information of Device R0

Verify the LDP Entries in the Internal Topology Table

Purpose

Display the route entries in the Label Distribution Protocol (LDP) internal topology table.

Action

On Device R0, from operational mode, run the `show ldp route` command to display the internal topology table of LDP.

```
user@R0> show ldp route

Destination         Next-hop intf/lsp/table           Next-hop address
 10.4.0.0/16        fxp0.0                            10.92.31.254
 10.5.0.0/16        fxp0.0                            10.92.31.254
 10.6.128.0/17      fxp0.0                            10.92.31.254
 10.9.0.0/16        fxp0.0                            10.92.31.254
 10.10.0.0/16       fxp0.0                            10.92.31.254
 10.13.4.0/23       fxp0.0                            10.92.31.254
 10.13.10.0/23      fxp0.0                            10.92.31.254
 10.82.0.0/15       fxp0.0                            10.92.31.254
 10.84.0.0/16       fxp0.0                            10.92.31.254
 10.85.12.0/22      fxp0.0                            10.92.31.254
 10.92.0.0/16       fxp0.0                            10.92.31.254
 10.92.16.0/20      fxp0.0
```

```
10.92.20.175/32
10.94.0.0/16       fxp0.0                    10.92.31.254
10.99.0.0/16       fxp0.0                    10.92.31.254
10.102.0.0/16      fxp0.0                    10.92.31.254
10.150.0.0/16      fxp0.0                    10.92.31.254
10.155.0.0/16      fxp0.0                    10.92.31.254
10.157.64.0/19     fxp0.0                    10.92.31.254
10.160.0.0/16      fxp0.0                    10.92.31.254
10.204.0.0/16      fxp0.0                    10.92.31.254
10.205.0.0/16      fxp0.0                    10.92.31.254
10.206.0.0/16      fxp0.0                    10.92.31.254
10.207.0.0/16      fxp0.0                    10.92.31.254
10.209.0.0/16      fxp0.0                    10.92.31.254
10.212.0.0/16      fxp0.0                    10.92.31.254
10.213.0.0/16      fxp0.0                    10.92.31.254
10.214.0.0/16      fxp0.0                    10.92.31.254
10.215.0.0/16      fxp0.0                    10.92.31.254
10.216.0.0/16      fxp0.0                    10.92.31.254
10.218.13.0/24     fxp0.0                    10.92.31.254
10.218.14.0/24     fxp0.0                    10.92.31.254
10.218.16.0/20     fxp0.0                    10.92.31.254
10.218.32.0/20     fxp0.0                    10.92.31.254
10.227.0.0/16      fxp0.0                    10.92.31.254
10.255.111.0/24    ge-0/0/2.0                11.11.11.2
10.255.111.4/32    ge-0/0/2.0                11.11.11.2
10.255.112.1/32    lo0.0
10.255.112.2/32    ge-0/0/2.0                11.11.11.2
11.11.11.0/24      ge-0/0/2.0
11.11.11.1/32
12.12.12.0/24      ge-0/0/2.0                11.11.11.2
15.15.15.0/24      ge-0/0/1.0
15.15.15.1/32
22.22.22.0/24      ge-0/0/0.0
22.22.22.1/32
23.23.23.0/24      ge-0/0/2.0                11.11.11.2
24.24.24.0/24      ge-0/0/2.0                11.11.11.2
25.25.25.0/24      ge-0/0/2.0                11.11.11.2
128.92.17.45/32    ge-0/0/2.0                11.11.11.2
128.92.20.175/32   lo0.0
128.92.21.186/32   ge-0/0/2.0                11.11.11.2
128.92.25.135/32   ge-0/0/2.0                11.11.11.2
128.92.27.91/32    ge-0/0/2.0                11.11.11.2
128.92.28.70/32    ge-0/0/2.0                11.11.11.2
```

```
 172.16.0.0/12      fxp0.0                          10.92.31.254
 192.168.0.0/16     fxp0.0                          10.92.31.254
 192.168.102.0/23   fxp0.0                          10.92.31.254
 207.17.136.0/24    fxp0.0                          10.92.31.254
 207.17.136.192/32  fxp0.0                          10.92.31.254
 207.17.137.0/24    fxp0.0                          10.92.31.254
 224.0.0.5/32
```

Meaning

The output displays the route entries in the Label Distribution Protocol (LDP) internal topology table of Device R0.

Verify Only FEC Information of LDP Route

Purpose

Display only the FEC information of LDP route.

Action

On Device R0, from operational mode, run the `show ldp route fec-only` command to display the routes in the routing table.

```
user@R0> show ldp route fec-only

Destination         Next-hop intf/lsp/table           Next-hop address
 10.255.111.1/32    ge-0/0/2.0                        11.11.11.2
 10.255.111.2/32    ge-0/0/2.0                        11.11.11.2
 10.255.111.3/32    ge-0/0/2.0                        11.11.11.2
 10.255.111.4/32    ge-0/0/2.0                        11.11.11.2
 10.255.112.1/32    lo0.0
 10.255.112.2/32    ge-0/0/2.0                        11.11.11.2
```

Meaning

The output displays only the FEC routes of LDP protocol available for Device R0.

Verify FEC and Shadow Routes of LDP

Purpose

Display the FEC and the shadow routes in the routing table.

Action

On Device R0, from operational mode, run the `show ldp route fec-and-route` command to display the FEC and shadow routes in the routing table.

```
user@R0> show ldp route fec-and-route

Destination         Next-hop intf/lsp/table           Next-hop address
 10.4.0.0/16        fxp0.0                            10.92.31.254
 10.5.0.0/16        fxp0.0                            10.92.31.254
 10.6.128.0/17      fxp0.0                            10.92.31.254
 10.9.0.0/16        fxp0.0                            10.92.31.254
 10.10.0.0/16       fxp0.0                            10.92.31.254
 10.13.4.0/23       fxp0.0                            10.92.31.254
 10.13.10.0/23      fxp0.0                            10.92.31.254
 10.82.0.0/15       fxp0.0                            10.92.31.254
 10.84.0.0/16       fxp0.0                            10.92.31.254
 10.85.12.0/22      fxp0.0                            10.92.31.254
 10.92.0.0/16       fxp0.0                            10.92.31.254
 10.92.16.0/20      fxp0.0
 10.92.20.175/32
 10.94.0.0/16       fxp0.0                            10.92.31.254
 10.99.0.0/16       fxp0.0                            10.92.31.254
 10.102.0.0/16      fxp0.0                            10.92.31.254
 10.150.0.0/16      fxp0.0                            10.92.31.254
 10.155.0.0/16      fxp0.0                            10.92.31.254
 10.157.64.0/19     fxp0.0                            10.92.31.254
 10.160.0.0/16      fxp0.0                            10.92.31.254
 10.204.0.0/16      fxp0.0                            10.92.31.254
 10.205.0.0/16      fxp0.0                            10.92.31.254
 10.206.0.0/16      fxp0.0                            10.92.31.254
 10.207.0.0/16      fxp0.0                            10.92.31.254
 10.209.0.0/16      fxp0.0                            10.92.31.254
 10.212.0.0/16      fxp0.0                            10.92.31.254
 10.213.0.0/16      fxp0.0                            10.92.31.254
 10.214.0.0/16      fxp0.0                            10.92.31.254
```

```
10.215.0.0/16      fxp0.0                    10.92.31.254
10.216.0.0/16      fxp0.0                    10.92.31.254
10.218.13.0/24     fxp0.0                    10.92.31.254
10.218.14.0/24     fxp0.0                    10.92.31.254
10.218.16.0/20     fxp0.0                    10.92.31.254
10.218.32.0/20     fxp0.0                    10.92.31.254
10.227.0.0/16      fxp0.0                    10.92.31.254
10.255.111.0/24    ge-0/0/2.0                11.11.11.2
10.255.111.1/32    ge-0/0/2.0                11.11.11.2
10.255.111.2/32    ge-0/0/2.0                11.11.11.2
10.255.111.3/32    ge-0/0/2.0                11.11.11.2
10.255.111.4/32    ge-0/0/2.0                11.11.11.2
10.255.111.4/32    ge-0/0/2.0                11.11.11.2
10.255.112.1/32    lo0.0
10.255.112.1/32    lo0.0
10.255.112.2/32    ge-0/0/2.0                11.11.11.2
10.255.112.2/32    ge-0/0/2.0                11.11.11.2
11.11.11.0/24      ge-0/0/2.0
11.11.11.1/32
12.12.12.0/24      ge-0/0/2.0                11.11.11.2
15.15.15.0/24      ge-0/0/1.0
15.15.15.1/32
22.22.22.0/24      ge-0/0/0.0
22.22.22.1/32
23.23.23.0/24      ge-0/0/2.0                11.11.11.2
24.24.24.0/24      ge-0/0/2.0                11.11.11.2
25.25.25.0/24      ge-0/0/2.0                11.11.11.2
128.92.17.45/32    ge-0/0/2.0                11.11.11.2
128.92.20.175/32   lo0.0
128.92.21.186/32   ge-0/0/2.0                11.11.11.2
128.92.25.135/32   ge-0/0/2.0                11.11.11.2
128.92.27.91/32    ge-0/0/2.0                11.11.11.2
128.92.28.70/32    ge-0/0/2.0                11.11.11.2
172.16.0.0/12      fxp0.0                    10.92.31.254
192.168.0.0/16     fxp0.0                    10.92.31.254
192.168.102.0/23   fxp0.0                    10.92.31.254
207.17.136.0/24    fxp0.0                    10.92.31.254
207.17.136.192/32  fxp0.0                    10.92.31.254
207.17.137.0/24    fxp0.0                    10.92.31.254
224.0.0.5/32
```

Meaning

The output displays the FEC and the shadow routes of Device R0

Configuring LDP Route Preferences

When several protocols calculate routes to the same destination, route preferences are used to select which route is installed in the forwarding table. The route with the lowest preference value is selected. The preference value can be a number in the range 0 through 255. By default, LDP routes have a preference value of 9.

To modify the route preferences, include the `preference` statement:

```
preference preference;
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

LDP Graceful Restart

LDP graceful restart enables a router whose LDP control plane is undergoing a restart to continue to forward traffic while recovering its state from neighboring routers. It also enables a router on which helper mode is enabled to assist a neighboring router that is attempting to restart LDP.

During session initialization, a router advertises its ability to perform LDP graceful restart or to take advantage of a neighbor performing LDP graceful restart by sending the graceful restart TLV. This TLV contains two fields relevant to LDP graceful restart: the reconnect time and the recovery time. The values of the reconnect and recovery times indicate the graceful restart capabilities supported by the router.

When a router discovers that a neighboring router is restarting, it waits until the end of the recovery time before attempting to reconnect. The recovery time is the length of time a router waits for LDP to restart gracefully. The recovery time period begins when an initialization message is sent or received. This time period is also typically the length of time that a neighboring router maintains its information about the restarting router, allowing it to continue to forward traffic.

You can configure LDP graceful restart in both the master instance for the LDP protocol and for a specific routing instance. You can disable graceful restart at the global level for all protocols, at the protocol level for LDP only, and on a specific routing instance. LDP graceful restart is disabled by default, because at the global level, graceful restart is disabled by default. However, helper mode (the ability to assist a neighboring router attempting a graceful restart) is enabled by default.

The following are some of the behaviors associated with LDP graceful restart:

- Outgoing labels are not maintained in restarts. New outgoing labels are allocated.

- When a router is restarting, no label-map messages are sent to neighbors that support graceful restart until the restarting router has stabilized (label-map messages are immediately sent to neighbors that do not support graceful restart). However, all other messages (keepalive, address-message, notification, and release) are sent as usual. Distributing these other messages prevents the router from distributing incomplete information.
- Helper mode and graceful restart are independent. You can disable graceful restart in the configuration, but still allow the router to cooperate with a neighbor attempting to restart gracefully.

Configuring LDP Graceful Restart

IN THIS SECTION

- Enabling Graceful Restart | **1392**
- Disabling LDP Graceful Restart or Helper Mode | **1393**
- Configuring Reconnect Time | **1394**
- Configuring Recovery Time and Maximum Recovery Time | **1394**

When you alter the graceful restart configuration at either the `[edit routing-options graceful-restart]` or `[edit protocols ldp graceful-restart]` hierarchy levels, any running LDP session is automatically restarted to apply the graceful restart configuration. This behavior mirrors the behavior of BGP when you alter its graceful restart configuration.

By default, graceful restart helper mode is enabled, but graceful restart is disabled. Thus, the default behavior of a router is to assist neighboring routers attempting a graceful restart, but not to attempt a graceful restart itself.

To configure LDP graceful restart, see the following sections:

Enabling Graceful Restart

To enable LDP graceful restart, you also need to enable graceful restart on the router. To enable graceful restart, include the `graceful-restart` statement:

```
graceful-restart;
```

You can include this statement at the following hierarchy levels:

- `[edit routing-options]`

- [edit logical-systems *logical-system-name* routing-options]

NOTE: ACX Series routers do not support [edit logical-systems logical-system-name routing-options] hierarchy level.

The graceful-restart statement enables graceful restart for all protocols supporting this feature on the router. For more information about graceful restart, see the Junos OS Routing Protocols Library for Routing Devices.

By default, LDP graceful restart is enabled when you enable graceful restart at both the LDP protocol level and on all the routing instances. However, you can disable both LDP graceful restart and LDP graceful restart helper mode.

Disabling LDP Graceful Restart or Helper Mode

To disable LDP graceful restart and recovery, include the disable statement:

```
ldp {
    graceful-restart {
        disable;
    }
}
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

You can disable helper mode at the LDP protocols level only. You cannot disable helper mode for a specific routing instance. To disable LDP helper mode, include the helper-disable statement:

```
ldp {
    graceful-restart {
        helper-disable;
    }
}
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

The following LDP graceful restart configurations are possible:

- LDP graceful restart and helper mode are both enabled.

- LDP graceful restart is disabled but helper mode is enabled. A router configured in this way cannot restart gracefully but can help a restarting neighbor.
- LDP graceful restart and helper mode are both disabled. The router does not use LDP graceful restart or the graceful restart type, length, and value (TLV) sent in the initialization message. The router behaves as a router that cannot support LDP graceful restart.

A configuration error is issued if you attempt to enable graceful restart and disable helper mode.

Configuring Reconnect Time

After the LDP connection between neighbors fails, neighbors wait a certain amount of time for the gracefully restarting router to resume sending LDP messages. After the wait period, the LDP session can be reestablished. You can configure the wait period in seconds. This value is included in the fault tolerant session TLV sent in LDP initialization messages when LDP graceful restart is enabled.

Suppose that Router A and Router B are LDP neighbors. Router A is the restarting Router. The reconnect time is the time that Router A tells Router B to wait after Router B detects that Router A restarted.

To configure the reconnect time, include the `reconnect-time` statement:

```
graceful-restart {
    reconnect-time seconds;
}
```

You can set the reconnect time to a value in the range from 30 through 300 seconds. By default, it is 60 seconds.

For a list of hierarchy levels at which you can configure these statements, see the statement summary sections for these statements.

Configuring Recovery Time and Maximum Recovery Time

The recovery time is the amount of time a router waits for LDP to restart gracefully. The recovery time period begins when an initialization message is sent or received. This period is also typically the amount of time that a neighboring router maintains its information about the restarting router, allowing it to continue to forward traffic.

To prevent a neighboring router from being adversely affected if it receives a false value for the recovery time from the restarting router, you can configure the maximum recovery time on the neighboring router. A neighboring router maintains its state for the shorter of the two times. For example, Router A is performing an LDP graceful restart. It has sent a recovery time of 900 seconds to neighboring Router

B. However, Router B has its maximum recovery time configured at 400 seconds. Router B will only wait for 400 seconds before it purges its LDP information from Router A.

To configure recovery time, include the `recovery-time` statement and the `maximum-neighbor-recovery-time` statement:

```
graceful-restart {
    maximum-neighbor-recovery-time seconds;
    recovery-time seconds;
}
```

For a list of hierarchy levels at which you can configure these statements, see the statement summary sections for these statements.

Filtering Inbound LDP Label Bindings

IN THIS SECTION

- Examples: Filtering Inbound LDP Label Bindings | **1397**

You can filter received LDP label bindings, applying policies to accept or deny bindings advertised by neighboring routers. To configure received-label filtering, include the `import` statement:

```
import [ policy-names ];
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

The named policy (configured at the `[edit policy-options]` hierarchy level) is applied to all label bindings received from all LDP neighbors. All filtering is done with `from` statements. Table 31 on page 1395 lists the only `from` operators that apply to LDP received-label filtering.

Table 31: from Operators That Apply to LDP Received-Label Filtering

from Operator	Description
`interface`	Matches on bindings received from a neighbor that is adjacent over the specified interface

Table 31: from Operators That Apply to LDP Received-Label Filtering *(Continued)*

from Operator	Description
`neighbor`	Matches on bindings received from the specified LDP router ID
`next-hop`	Matches on bindings received from a neighbor advertising the specified interface address
`route-filter`	Matches on bindings with the specified prefix

If a binding is filtered, it still appears in the LDP database, but is not considered for installation as part of a label-switched path (LSP).

Generally, applying policies in LDP can be used only to block the establishment of LSPs, not to control their routing. This is because the path that an LSP follows is determined by unicast routing, and not by LDP. However, when there are multiple equal-cost paths to the destination through different neighbors, you can use LDP filtering to exclude some of the possible next hops from consideration. (Otherwise, LDP chooses one of the possible next hops at random.)

LDP sessions are not bound to interfaces or interface addresses. LDP advertises only per-router (not per-interface) labels; so if multiple parallel links exist between two routers, only one LDP session is established, and it is not bound to a single interface. When a router has multiple adjacencies to the same neighbor, take care to ensure that the filter does what is expected. (Generally, using `next-hop` and `interface` is not appropriate in this case.)

If a label has been filtered (meaning that it has been rejected by the policy and is not used to construct an LSP), it is marked as filtered in the database:

```
user@host> show ldp database
Input label database, 10.10.255.1:0-10.10.255.6:0
Label Prefix
3 10.10.255.6/32 (Filtered)
Output label database, 10.10.255.1:0-10.10.255.6:0
Label Prefix
3 10.10.255.1/32 (Filtered)
```

For more information about how to configure policies for LDP, see the Routing Policies, Firewall Filters, and Traffic Policers User Guide.

Examples: Filtering Inbound LDP Label Bindings

Accept only /32 prefixes from all neighbors:

```
[edit]
protocols {
    ldp {
        import only-32;
        ...
    }
}
policy-options {
    policy-statement only-32 {
        term first {
            from {
                route-filter 0.0.0.0/0 upto /31;
            }
            then reject;
        }
        then accept;
    }
}
```

Accept `131.108/16` or longer from router ID `10.10.255.2` and accept all prefixes from all other neighbors:

```
[edit]
protocols {
    ldp {
        import nosy-neighbor;
        ...
    }
}
policy-options {
    policy-statement nosy-neighbor {
        term first {
            from {
                neighbor 10.10.255.2;
                route-filter 131.108.0.0/16 orlonger accept;
                route-filter 0.0.0.0/0 orlonger reject;
            }
        }
```

```
        then accept;
    }
}
```

Filtering Outbound LDP Label Bindings

IN THIS SECTION

- Examples: Filtering Outbound LDP Label Bindings | **1399**

You can configure export policies to filter LDP outbound labels. You can filter outbound label bindings by applying routing policies to block bindings from being advertised to neighboring routers. To configure outbound label filtering, include the `export` statement:

```
export [policy-name];
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

The named export policy (configured at the `[edit policy-options]` hierarchy level) is applied to all label bindings transmitted to all LDP neighbors. The only `from` operator that applies to LDP outbound label filtering is `route-filter`, which matches bindings with the specified prefix. The only `to` operators that apply to outbound label filtering are the operators in Table 32 on page 1398.

Table 32: to Operators for LDP Outbound-Label Filtering

to Operator	Description
`interface`	Matches on bindings sent to a neighbor that is adjacent over the specified interface
`neighbor`	Matches on bindings sent to the specified LDP router ID
`next-hop`	Matches on bindings sent to a neighbor advertising the specified interface address

If a binding is filtered, the binding is not advertised to the neighboring router, but it can be installed as part of an LSP on the local router. You can apply policies in LDP to block the establishment of LSPs, but not to control their routing. The path an LSP follows is determined by unicast routing, not by LDP.

LDP sessions are not bound to interfaces or interface addresses. LDP advertises only per-router (not per-interface) labels. If multiple parallel links exist between two routers, only one LDP session is established, and it is not bound to a single interface.

Do not use the `next-hop` and `interface` operators when a router has multiple adjacencies to the same neighbor.

Filtered labels are marked in the database:

```
user@host> show ldp database
Input label database, 10.10.255.1:0-10.10.255.3:0
Label Prefix
100007 10.10.255.2/32
3 10.10.255.3/32
Output label database, 10.10.255.1:0-10.10.255.3:0
Label Prefix
3 10.10.255.1/32
100001 10.10.255.6/32 (Filtered)
```

For more information about how to configure policies for LDP, see the Routing Policies, Firewall Filters, and Traffic Policers User Guide.

Examples: Filtering Outbound LDP Label Bindings

Block transmission of the route for `10.10.255.6/32` to any neighbors:

```
[edit protocols]
ldp {
    export block-one;
}
policy-options {
    policy-statement block-one {
        term first {
            from {
                route-filter 10.10.255.6/32 exact;
            }
            then reject;
        }
        then accept;
    }
}
```

Send only 131.108/16 or longer to router ID 10.10.255.2, and send all prefixes to all other routers:

```
[edit protocols]
ldp {
    export limit-lsps;
}
policy-options {
    policy-statement limit-lsps {
        term allow-one {
            from {
                route-filter 131.108.0.0/16 orlonger;
            }
            to {
                neighbor 10.10.255.2;
            }
            then accept;
        }
        term block-the-rest {
            to {
                neighbor 10.10.255.2;
            }
            then reject;
        }
        then accept;
    }
}
```

Specifying the Transport Address Used by LDP

Routers must first establish a TCP session between each other before they can establish an LDP session. The TCP session enables the routers to exchange the label advertisements needed for the LDP session. To establish the TCP session, each router must learn the other router's transport address. The transport address is an IP address used to identify the TCP session over which the LDP session will run.

To configure the LDP transport address, include the transport-address statement:

```
transport-address (router-id | interface);
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

If you specify the `router-id` option, the address of the router identifier is used as the transport address (unless otherwise configured, the router identifier is typically the same as the loopback address). If you specify the `interface` option, the interface address is used as the transport address for any LDP sessions to neighbors that can be reached over that interface. Note that the router identifier is used as the transport address by default.

NOTE: For proper operation the LDP transport address must be reachable. The router-ID is an identifier, not a routable IP address. For this reason its recommended that the router-ID be set to match the loopback address, and that the loopback address is advertised by the IGP.

You cannot specify the `interface` option when there are multiple parallel links to the same LDP neighbor, because the LDP specification requires that the same transport address be advertised on all interfaces to the same neighbor. If LDP detects multiple parallel links to the same neighbor, it disables interfaces to that neighbor one by one until the condition is cleared, either by disconnecting the neighbor on an interface or by specifying the `router-id` option.

Control Transport Address Used for Targeted-LDP Session

IN THIS SECTION

- Benefits of Controlling Transport Address Used for Targeted-LDP Session | **1401**
- Targeted-LDP Transport Address Overview | **1402**
- Transport Address Preference | **1402**
- Troubleshooting Transport Address Configuration | **1403**

To establish a TCP session between two devices, each device must learn the other device's transport address. The transport address is an IP address used to identify the TCP session over which the LDP session operates. Earlier, this transport address could only be the router-ID or an interface address. With the LDP transport-address feature, you can explicitly configure any IP address as the transport address for targeted LDP neighbors for Layer 2 circuit, MPLS, and VPLS adjacencies. This enables you to control the targeted-LDP sessions using transport-address configuration.

Benefits of Controlling Transport Address Used for Targeted-LDP Session

Configuring transport address for establishing targeted-LDP sessions has the following benefits:

- **Flexible interface configurations**—Provides the flexibility of configuring multiple IP addresses for one loopback interface without interrupting the creation of LDP session between the targeted-LDP neighbors.
- **Ease of operation**—Transport address configured at the interface-level, allows you to use more than one protocol in the IGP backbone for LDP. This enables smooth and easy operations.

Targeted-LDP Transport Address Overview

Prior to Junos OS Release 19.1R1, LDP provided support only for router-ID or the interface address as the transport address on any LDP interface. The adjacencies formed on that interface used one of the IP addresses assigned to the interface or the router-ID. In case of targeted adjacency, the interface is the loopback interface. When multiple loopback addresses were configured on the device, the transport address could not be derived for the interface, and as a result, the LDP session could not be established.

Starting in Junos OS Release 19.1R1, in addition to the default IP addresses used for transport address of targeted-LDP sessions, you can configure any other IP address as the transport address under the `session`, `session-group`, and `interface` configuration statements. The transport address configuration is applicable for configured neighbors only including Layer 2 circuits, MPLS, and VPLS adjacencies. This configuration does not apply to discovered adjacencies (targeted or not).

Transport Address Preference

You can configure transport address for targeted-LDP sessions at the session, session-group, and interface level.

After the transport address is configured, the targeted-LDP session is established based on the transport address preference of LDP.

The order of preference of transport address for targeted neighbor (configured through Layer 2 circuit, MPLS, VPLS, and LDP configuration) is as follows:

1. Under `[edit protocols ldp session]` hierarchy.
2. Under `[edit protocols ldp session-group]` hierarchy.
3. Under `[edit protocols ldp interfcae lo0]` hierarchy.
4. Under `[edit protocols ldp]` hierarchy.
5. Default address.

The order of preference of transport address for the discovered neighbors is as follows:

1. Under `[edit protocols ldp interfcae]` hierarchy.
2. Under `[edit protocols ldp]` hierarchy.

3. Default address.

The order of preference of transport address for auto-targeted neighbors where LDP is configured to accept hello packets is as follows:

1. Under `[edit protocols ldp interfcae lo0]` hierarchy.
2. Under `[edit protocols ldp]` hierarchy.
3. Default address.

Troubleshooting Transport Address Configuration

You can use the following show command outputs to troubleshoot targeted-LDP sessions:

- `show ldp session`
- `show ldp neighbor`

  The `detail` level of output of the `show ldp neighbor` command displays the transport address sent in the hello messages to the targeted neighbor. If this address is not reachable from the neighbor, the LDP session does not come up.

- `show configuration protocols ldp`

You can also enable LDP traceoptions for further troubleshooting.

- If the configuration is changed from using a transport address that is invalid (non reachable) to transport address that is valid, the following traces can be observed:

```
May 29 10:47:11.569722 Incoming connect from 10.55.1.4
May 29 10:47:11.570064 Connection 10.55.1.4 state Closed -> Open
May 29 10:47:11.570727 Session 10.55.1.4 state Nonexistent -> Initialized
May 29 10:47:11.570768 Session 10.55.1.4 state Initialized -> OpenRec
May 29 10:47:11.570799 LDP: Session param Max PDU length 4096 from 10.55.1.4, negotiated 4096
May 29 10:47:11.570823 Session 10.55.1.4 GR state Nonexistent -> Operational
May 29 10:47:11.669295 Session 10.55.1.4 state OpenRec -> Operational
May 29 10:47:11.669387 RPD_LDP_SESSIONUP: LDP session 10.55.1.4 is up
```

- If the configuration is changed from using a transport address that is valid to transport address that is invalid (non reachable),the following traces can be observed:

```
May 29 10:42:36.317942 Session 10.55.1.4 GR state Operational -> Nonexistent
May 29 10:42:36.318171 Session 10.55.1.4 state Operational -> Closing
```

```
May 29 10:42:36.318208 LDP session 10.55.1.4 is down, reason: received notification from peer
May 29 10:42:36.318236 RPD_LDP_SESSIONDOWN: LDP session 10.55.1.4 is down, reason: received
notification from peer
May 29 10:42:36.320081 Connection 10.55.1.4 state Open -> Closed
May 29 10:42:36.322411 Session 10.55.1.4 state Closing -> Nonexistent
```

In case of faulty configuration, perform the following troubleshooting tasks:

- Check the `address family`. The transport address that is configured under the `session` statement must belong to the same address family as the neighbor or session.
- The address that is configured as the transport address under a `neighbor` or `session` statement must be local to the router for the targeted hello messages to start. You can check if the address is configured. If the address is not configured under any interface, the configuration is rejected.

Configuring the Prefixes Advertised into LDP from the Routing Table

IN THIS SECTION

- Example: Configuring the Prefixes Advertised into LDP | **1405**

You can control the set of prefixes that are advertised into LDP and cause the router to be the egress router for those prefixes. By default, only the loopback address is advertised into LDP. To configure the set of prefixes from the routing table to be advertised into LDP, include the `egress-policy` statement:

```
egress-policy policy-name;
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

NOTE: If you configure an egress policy for LDP that does not include the loopback address, it is no longer advertised in LDP. To continue to advertise the loopback address, you need to explicitly configure it as a part of the LDP egress policy.

The named policy (configured at the `[edit policy-options]` or `[edit logical-systems` *`logical-system-name`* `policy-options]` hierarchy level) is applied to all routes in the routing table. Those routes that match the policy are advertised into LDP. You can control the set of neighbors to which those prefixes are advertised by using the `export` statement. Only **from** operators are considered; you can use any valid **from** operator. For more information, see the Junos OS Routing Protocols Library for Routing Devices.

NOTE: ACX Series routers do not support [`edit logical-systems`] hierarchy level.

Example: Configuring the Prefixes Advertised into LDP

Advertise all connected routes into LDP:

```
[edit protocols]
ldp {
    egress-policy connected-only;
}
policy-options {
    policy-statement connected-only {
        from {
            protocol direct;
        }
        then accept;
    }
}
```

Configuring FEC Deaggregation

When an LDP egress router advertises multiple prefixes, the prefixes are bound to a single label and aggregated into a single forwarding equivalence class (FEC). By default, LDP maintains this aggregation as the advertisement traverses the network.

Normally, because an LSP is not split across multiple next hops and the prefixes are bound into a single LSP, load-balancing across equal-cost paths does not occur. You can, however, load-balance across equal-cost paths if you configure a load-balancing policy and deaggregate the FECs.

Deaggregating the FECs causes each prefix to be bound to a separate label and become a separate LSP.

To configure deaggregated FECs, include the `deaggregate` statement:

```
deaggregate;
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

For all LDP sessions, you can configure deaggregated FECs only globally.

Deaggregating a FEC allows the resulting multiple LSPs to be distributed across multiple equal-cost paths and distributes LSPs across the multiple next hops on the egress segments but installs only one next hop per LSP.

To aggregate FECs, include the `no-deaggregate` statement:

```
no-deaggregate;
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

For all LDP sessions, you can configure aggregated FECs only globally.

Configuring Policers for LDP FECs

You can configure the Junos OS to track and police traffic for LDP FECs. LDP FEC policers can be used to do any of the following:

- Track or police the ingress traffic for an LDP FEC.
- Track or police the transit traffic for an LDP FEC.
- Track or police LDP FEC traffic originating from a specific forwarding class.
- Track or police LDP FEC traffic originating from a specific virtual routing and forwarding (VRF) site.
- Discard false traffic bound for a specific LDP FEC.

To police traffic for an LDP FEC, you must first configure a filter. Specifically, you need to configure either the `interface` statement or the `interface-set` statement at the `[edit firewall family` *`protocol-family`* `filter` *`filter-name`* `term` *`term-name`* `from]` hierarchy level. The `interface` statement allows you to match the filter to a single interface. The `interface-set` statement allows you to match the filter to multiple interfaces.

For more information on how to configure the `interface` statement, the `interface-set` statement, and policers for LDP FECs, see the Routing Policies, Firewall Filters, and Traffic Policers User Guide.

Once you have configured the filters, you need to include them in the `policing` statement configuration for LDP. To configure policers for LDP FECs, include the `policing` statement:

```
policing {
    fec fec-address {
        ingress-traffic filter-name;
        transit-traffic filter-name;
```

```
    }
}
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

The `policing` statement includes the following options:

- `fec`—Specify the FEC address for the LDP FEC you want to police.
- `ingress-filter`—Specify the name of the ingress traffic filter.
- `transit-traffic`—Specify the name of the transit traffic filter.

Configuring LDP IPv4 FEC Filtering

By default, when a targeted LDP session is established, the Junos OS always exchanges both the IPv4 forwarding equivalence classes (FECs) and the Layer 2 circuit FECs over the targeted LDP session. For an LDP session to an indirectly connected neighbor, you might only want to export Layer 2 circuit FECs to the neighbor if the session was specifically configured to support Layer 2 circuits or VPLS.

In a mixed vendor network where all non-BGP prefixes are advertised into LDP, the LDP database can become large. For this type of environment, it can be useful to prevent the advertisement of IPv4 FECs over LDP sessions formed because of Layer 2 circuit or LDP VPLS configuration. Similarly, it can be useful to filter any IPv4 FECs received in this sort of environment.

If all the LDP neighbors associated with an LDP session are Layer 2 only, you can configure the Junos OS to advertise only Layer 2 circuit FECs by configuring the `l2-smart-policy` statement. This feature also automatically filters out the IPv4 FECs received on this session. Configuring an explicit export or import policy that is activated for `l2-smart-policy` disables this feature in the corresponding direction.

If one of the LDP session's neighbors is formed because of a discovered adjacency or if the adjacency is formed because of an LDP tunneling configuration on one or more RSVP LSPs, the IPv4 FECs are advertised and received using the default behavior.

To prevent LDP from exporting IPv4 FECs over LDP sessions with Layer 2 neighbors only and to filter out IPv4 FECs received over such sessions, include the `l2-smart-policy` statement:

```
l2-smart-policy;
```

For a list of hierarchy levels at which you can configure this statement, see the statement summary for this statement.

Configuring BFD for LDP LSPs

You can configure Bidirectional Forwarding Detection (BFD) for LDP LSPs. The BFD protocol is a simple hello mechanism that detects failures in a network. Hello packets are sent at a specified, regular interval. A neighbor failure is detected when the router stops receiving a reply after a specified interval. BFD works with a wide variety of network environments and topologies. The failure detection timers for BFD have shorter time limits than the failure detection mechanisms of static routes, providing faster detection.

An error is logged whenever a BFD session for a path fails. The following shows how BFD for LDP LSP log messages might appear:

```
RPD_LDP_BFD_UP: LDP BFD session for FEC  10.255.16.14/32 is up
RPD_LDP_BFD_DOWN: LDP BFD session for FEC  10.255.16.14/32 is down
```

You can also configure BFD for RSVP LSPs, as described in "Configuring BFD for RSVP-Signaled LSPs" on page 213.

The BFD failure detection timers are adaptive and can be adjusted to be more or less aggressive. For example, the timers can adapt to a higher value if the adjacency fails, or a neighbor can negotiate a higher value for a timer than the configured value. The timers adapt to a higher value when a BFD session flap occurs more than three times in a span of 15 seconds. A back-off algorithm increases the receive (Rx) interval by two if the local BFD instance is the reason for the session flap. The transmission (Tx) interval is increased by two if the remote BFD instance is the reason for the session flap. You can use the `clear bfd adaptation` command to return BFD interval timers to their configured values. The `clear bfd adaptation` command is hitless, meaning that the command does not affect traffic flow on the routing device.

To enable BFD for LDP LSPs, include the `oam` and `bfd-liveness-detection` statements:

```
oam {
    bfd-liveness-detection {
        detection-time threshold milliseconds;
        ecmp;
        failure-action {
            remove-nexthop;
            remove-route;
        }
        holddown-interval seconds;
        ingress-policy ingress-policy-name;
        minimum-interval milliseconds;
        minimum-receive-interval milliseconds;
        minimum-transmit-interval milliseconds;
```

```
        multiplier detection-time-multiplier;
        no-adaptation;
        transmit-interval {
            minimum-interval milliseconds;
            threshold milliseconds;
        }
        version (0 | 1 | automatic);
    }
    fec fec-address {
        bfd-liveness-detection {
            detection-time threshold milliseconds;
            ecmp;
            failure-action {
                remove-nexthop;
                remove-route;
            }
            holddown-interval milliseconds;
            ingress-policy ingress-policy-name;
            minimum-interval milliseconds;
            minimum-receive-interval milliseconds;
            minimum-transmit-interval milliseconds;
            multiplier detection-time-multiplier;
            no-adaptation;
            transmit-interval {
                minimum-interval milliseconds;
                threshold milliseconds;
            }
            version (0 | 1 | automatic);
        }
        no-bfd-liveness-detection;
        periodic-traceroute {
            disable;
            exp exp-value;
            fanout fanout-value;
            frequency minutes;
            paths number-of-paths;
            retries retry-attempts;
            source address;
            ttl ttl-value;
            wait seconds;
        }
    }
    lsp-ping-interval seconds;
```

```
    periodic-traceroute {
        disable;
        exp exp-value;
        fanout fanout-value;
        frequency minutes;
        paths number-of-paths;
        retries retry-attempts;
        source address;
        ttl ttl-value;
        wait seconds;
    }
}
```

You can enable BFD for the LDP LSPs associated with a specific forwarding equivalence class (FEC) by configuring the FEC address using the `fec` option at the `[edit protocols ldp]` hierarchy level. Alternatively, you can configure an Operation Administration and Management (OAM) ingress policy to enable BFD on a range of FEC addresses. For more information, see "Configuring OAM Ingress Policies for LDP" on page 1860.

You cannot enable BFD LDP LSPs unless their equivalent FEC addresses are explicitly configured or OAM is enabled on the FECs using an OAM ingress policy. If BFD is not enabled for any FEC addresses, the BFD session will not come up.

You can configure the `oam` statement at the following hierarchy levels:

- `[edit protocols ldp]`
- `[edit logical-systems logical-system-name protocols ldp]`

NOTE: ACX Series routers do not support `[edit logical-systems]` hierarchy level.

The `oam` statement includes the following options:

- `fec`—Specify the FEC address. You must either specify a FEC address or configure an OAM ingress policy to ensure that the BFD session comes up.
- `lsp-ping-interval`—Specify the duration of the LSP ping interval in seconds. To issue a ping on an LDP-signaled LSP, use the `ping mpls ldp` command. For more information, see the CLI Explorer.

The `bfd-liveness-detection` statement includes the following options:

- `ecmp`—Cause LDP to establish BFD sessions for all ECMP paths configured for the specified FEC. If you configure the `ecmp` option, you must also configure the `periodic-traceroute` statement for the specified FEC. If you do not do so, the commit operation fails. You can configure the `periodic-`

`traceroute` statement at the global hierarchy level (`[edit protocols ldp oam]`) while only configuring the `ecmp` option for a specific FEC (`[edit protocols ldp oam fec` *address* `bfd-liveness-detection]`).

- *holddown-interval*—Specify the duration the BFD session should remain up before adding the route or next hop. Specifying a time of 0 seconds causes the route or next hop to be added as soon as the BFD session comes back up.
- `minimum-interval`—Specify the minimum transmit and receive interval. If you configure the `minimum-interval` option, you do not need to configure the `minimum-receive-interval` option or the `minimum-transmit-interval` option.
- `minimum-receive-interval`—Specify the minimum receive interval. The range is from 1 through 255,000 milliseconds.
- `minimum-transmit-interval`—Specify the minimum transmit interval. The range is from 1 through 255,000 milliseconds.
- `multiplier`—Specify the detection time multiplier. The range is from 1 through 255.
- *version*—Specify the BFD version. The options are BFD version 0 or BFD version 1. By default, the Junos OS software attempts to automatically determine the BFD version.

Configuring ECMP-Aware BFD for LDP LSPs

When you configure BFD for a FEC, a BFD session is established for only one active local next-hop for the router. However, you can configure multiple BFD sessions, one for each FEC associated with a specific equal-cost multipath (ECMP) path. For this to function properly, you also need to configure LDP LSP periodic traceroute. (See "Configuring LDP LSP Traceroute" on page 1563.) LDP LSP traceroute is used to discover ECMP paths. A BFD session is initiated for each ECMP path discovered. Whenever a BFD session for one of the ECMP paths fails, an error is logged.

LDP LSP traceroute is run periodically to check the integrity of the ECMP paths. The following might occur when a problem is discovered:

- If the latest LDP LSP traceroute for a FEC differs from the previous traceroute, the BFD sessions associated with that FEC (the BFD sessions for address ranges that have changed from previous run) are brought down and new BFD sessions are initiated for the destination addresses in the altered ranges.
- If the LDP LSP traceroute returns an error (for example, a timeout), all the BFD sessions associated with that FEC are torn down.

To configure LDP to establish BFD sessions for all ECMP paths configured for the specified FEC, include the ecmp statement.

```
ecmp;
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

Along with the ecmp statement, you must also include the periodic-traceroute statement, either in the global LDP OAM configuration (at the [edit protocols ldp oam] or [edit logical-systems *logical-system-name* protocols ldp oam] hierarchy level) or in the configuration for the specified FEC (at the [edit protocols ldp oam fec *address*] or [edit logical-systems *logical-system-name* protocols ldp oam fec *address*] hierarchy level). Otherwise, the commit operation fails.

NOTE: ACX Series routers do not support [edit logical-systems] hierarchy level.

Configuring a Failure Action for the BFD Session on an LDP LSP

You can configure route and next-hop properties in the event of a BFD session failure event on an LDP LSP. The failure event could be an existing BFD session that has gone down or could be a BFD session that never came up. LDP adds back the route or next hop when the relevant BFD session comes back up.

You can configure one of the following failure action options for the failure-action statement in the event of a BFD session failure on the LDP LSP:

- remove-nexthop—Removes the route corresponding to the next hop of the LSP's route at the ingress node when a BFD session failure event is detected.
- remove-route—Removes the route corresponding to the LSP from the appropriate routing tables when a BFD session failure event is detected. If the LSP is configured with ECMP and a BFD session corresponding to any path goes down, the route is removed.

To configure a failure action in the event of a BFD session failure on an LDP LSP, include either the remove-nexthop option or the remove-route option for the failure-action statement:

```
failure-action {
    remove-nexthop;
    remove-route;
}
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

Configuring the Holddown Interval for the BFD Session

You can specify the duration the BFD session should be up before adding a route or next hop by configuring the `holddown-interval` statement at either the `[edit protocols ldp oam bfd-livenesss-detection]` hierarchy level or at the `[edit protocols ldp oam fec address bfd-livenesss-detection]` hierarchy level. Specifying a time of 0 seconds causes the route or next hop to be added as soon as the BFD session comes back up.

```
holddown-interval seconds;
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

Configuring LDP Link Protection

You can configure Label Distribution Protocol (LDP) link protection for both unicast and multicast LDP label-switched paths (LSPs) to provide resiliency during link or node failure.

Before you begin:

1. Configure the device interfaces.
2. Configure the router ID and autonomous system number for the device.
3. Configure the following protocols:
   a. RSVP
   b. MPLS with traffic engineering capability.
   c. OSPF with traffic engineering capability.

NOTE: For multicast LDP link protection with loop-free alternative (LFA), enable link protection.

```
[edit protocols]
user@R0# set ospf area 0 interface all link-protection
```

To configure LDP link protection:

1. Enable point-to-multipoint LDP LSPs.

```
[edit protocols]
user@R0# set ldp p2mp
```

2. Enable LDP on all the interfaces of Router R0 (excluding the management interface) and configure link protection with dynamic RSVP bypass LSP.

```
[edit protocols]
user@R0# set ldp interface all link-protection dynamic-rsvp-lsp
user@R0# set ldp interface fxp0.0 disable
```

3. Verify and commit the configuration.

   For example:

```
[edit protocols]
user@R0# show protocols
rsvp {
    interface all;
    interface fxp0.0 {
        disable;
    }
}
mpls {
    traffic-engineering;
    interface all;
    interface fxp0.0 {
        disable;
    }
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface all {
            metric 1;
        }
        interface fxp0.0 {
            disable;
        }
    }
}
```

```
ldp {
    interface all {
        link-protection {
            dynamic-rsvp-lsp;
        }
    }
    interface fxp0.0 {
        disable;
    }
    p2mp;
}
```

```
[edit]
user@R0# commit
commit complete
```

Example: Configuring LDP Link Protection

IN THIS SECTION

- LDP Link Protection Overview | **1415**
- Example: Configuring LDP Link Protection | **1436**

LDP Link Protection Overview

IN THIS SECTION

- Introduction to LDP | **1416**
- Junos OS LDP Protocol Implementation | **1416**
- Understanding Multipoint Extensions to LDP | **1416**
- Using Multipoint Extensions to LDP on Targeted LDP Sessions | **1417**
- Current Limitations of LDP Link Protection | **1418**
- Using RSVP LSP as a Solution | **1419**
- Understanding Multicast LDP Link Protection | **1422**

- Different Modes for Providing LDP Link Protection | **1423**
- Label Operation for LDP Link Protection | **1425**
- Sample Multicast LDP Link Protection Configuration | **1433**
- Make-Before-Break | **1434**
- Caveats and Limitations | **1436**

Introduction to LDP

The Label Distribution Protocol (LDP) is a protocol for distributing labels in non-traffic-engineered applications. LDP allows routers to establish label-switched paths (LSPs) through a network by mapping network-layer routing information directly to the data link LSPs.

These LSPs might have an endpoint at a directly attached neighbor (comparable to IP hop-by-hop forwarding) or at a network egress node, enabling switching through all intermediary nodes. LSPs established by LDP can also traverse traffic-engineered LSPs created by RSVP.

LDP associates a forwarding equivalence class (FEC) with each LSP it creates. The FEC associated with an LSP specifies which packets are mapped to that LSP. LSPs are extended through a network as each router chooses the label advertised by the next hop for the FEC and splices it to the label it advertises to all other routers. This process forms a tree of LSPs that converge on the egress router.

Junos OS LDP Protocol Implementation

The Junos OS implementation of LDP supports LDP version 1. Junos OS supports a simple mechanism for tunneling between routers in an interior gateway protocol (IGP), to eliminate the required distribution of external routes within the core. Junos OS allows an MPLS tunnel next hop to all egress routers in the network, with only an IGP running in the core to distribute routes to egress routers. Edge routers run BGP but do not distribute external routes to the core. Instead, the recursive route lookup at the edge resolves to an LSP switched to the egress router. No external routes are necessary on the transit LDP routers.

Understanding Multipoint Extensions to LDP

An LDP defines mechanisms for setting up point-to-point, multipoint-to-point, point-to-multipoint, and multipoint-to-multipoint LSPs in the network. The point-to-multipoint and multipoint-to-multipoint LSPs are collectively referred to as multipoint LSPs, where traffic flows from a single source to multiple destinations, and from multiple sources to multiple destinations, respectively. The destination or egress routers are called leaf nodes, and traffic from the source traverses one or more transit nodes before reaching the leaf nodes.

NOTE: Junos OS does not provide support for multipoint-to-multipoint LSPs.

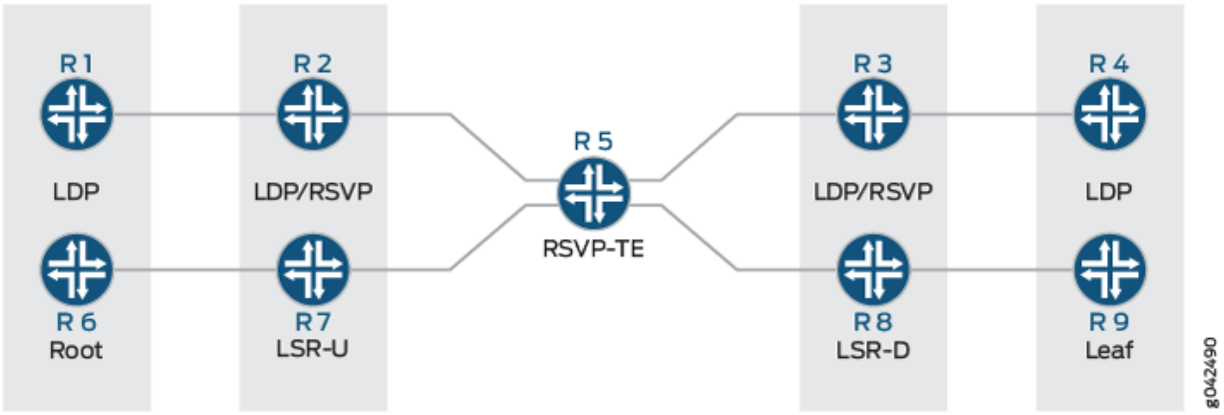
By taking advantage of the MPLS packet replication capability of the network, multipoint LSPs avoid unnecessary packet replication at the ingress router. Packet replication takes place only when packets are forwarded to two or more different destinations requiring different network paths.

Using Multipoint Extensions to LDP on Targeted LDP Sessions

The specification for the multipoint extensions to LDP requires that the two endpoints of an LDP session are directly connected by a Layer 2 medium, or are considered to be neighbors by the network's IGP. This is referred to as an LDP link session. When the two endpoints of an LDP session are not directly connected, the session is referred to as a targeted LDP session.

Past Junos OS implementations support multicast LDP for link sessions only. With the introduction of the LDP link protection feature, the multicast LDP capabilities are extended to targeted LDP sessions. Figure 2 shows a sample topology.

Figure 82: Multicast LDP Support for Targeted LDP Session



Routers R7 and R8 are the upstream (LSR-U) and downstream (LSR-D) label-switched routers (LSRs), respectively, and deploy multicast LDP. The core router, Router R5, has RSVP-TE enabled.

When LSR-D is setting up the point-to-multipoint LSP with root and LSP ID attributes, it determines the upstream LSR-U as a next-hop on the best path to the root (currently, this next-hop is assumed to be an IGP next hop).

With the multicast LDP support on targeted LDP sessions, you can determine if there is an LSP next hop to LSR-U which is on LSR-D's path to root, where LSR-D and LSR-U are not directly connected

neighbors, but targeted LDP peers. The point-to-multipoint label advertised on the targeted LDP session between LSR-D and LSR-U is not used unless there is an LSP between LSR-D and LSR-U. Therefore, a corresponding LSP in the reverse direction from LSR-U to LSR-D is required.

Data is transmitted on the point-to-multipoint LSP using unicast replication of packets, where LSR-U sends one copy to each downstream LSR of the point-to-multipoint LSP.

The data transmission is implemented in the following ways:

1. The point-to-multipoint capabilities on the targeted LDP session are negotiated.
2. The algorithm to select the upstream LSR is changed, where if IGP next hops are unavailable, or in other words, there is no LDP link session between LSR-D and LSR-U, an RSVP LSP is used as the next hop to reach LSR-U.
3. The incoming labels received over the targeted LDP sessions are installed as a branch next hop for this point-to-multipoint FEC route with the LDP label as the inner label and the RSVP label as the outer label.

Current Limitations of LDP Link Protection

When there is a link or node failure in an LDP network deployment, fast traffic recovery should be provided to recover impacted traffic flows for mission-critical services. In the case of multipoint LSPs, when one of the links of the point-to-multipoint tree fails, the subtrees might get detached until the IGP reconverges and the multipoint LSP is established using the best path from the downstream router to the new upstream router.

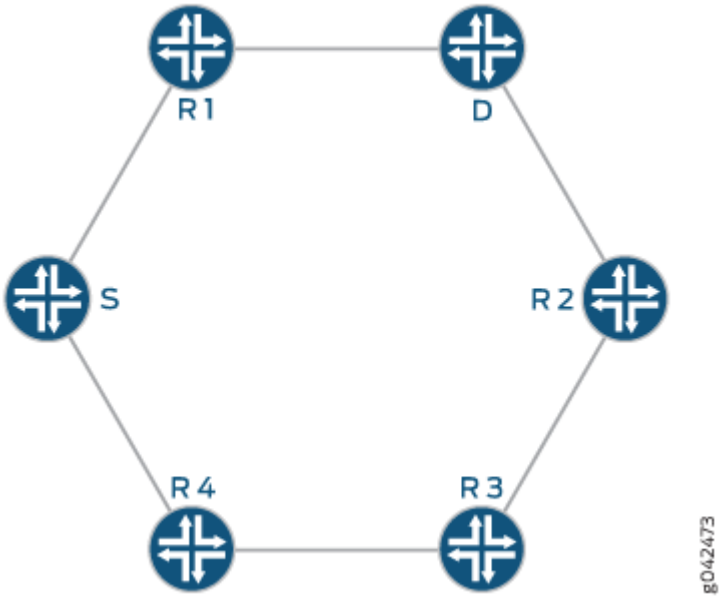
In fast reroute using local repair for LDP traffic, a backup path (repair path) is pre-installed in the Packet Forwarding Engine. When the primary path fails, traffic is rapidly moved to the backup path without having to wait for the routing protocols to converge. Loop-free alternate (LFA) is one of the methods used to provide IP fast reroute capability in the core and service provider networks.

Without LFA, when a link or a router fails or is returned to service, the distributed routing algorithms compute the new routes based on the changes in the network. The time during which the new routes are computed is referred to as routing transition. Until the routing transition is completed, the network connectivity is interrupted because the routers adjacent to a failure continue to forward the data packets through the failed component until an alternative path is identified.

However, LFA does not provide full coverage in all network deployments because of the IGP metrics. As a result, this is a limitation to the current LDP link protection schemes.

Figure 3 illustrates a sample network with incomplete LFA coverage, where traffic flows from the source router (S) to the destination router (D) through Router R1. Assuming that each link in the network has the same metric, if the link between the Router S and Router R1 fails, Router R4 is not an LFA that protects the S-R1 link, so traffic resiliency is lost. Thus, full coverage is not achieved by using plain LFA. In typical networks, there is always some percentage of LFA coverage gap with plain LFA.

Figure 83: Incomplete Coverage Problem with LFA



Using RSVP LSP as a Solution

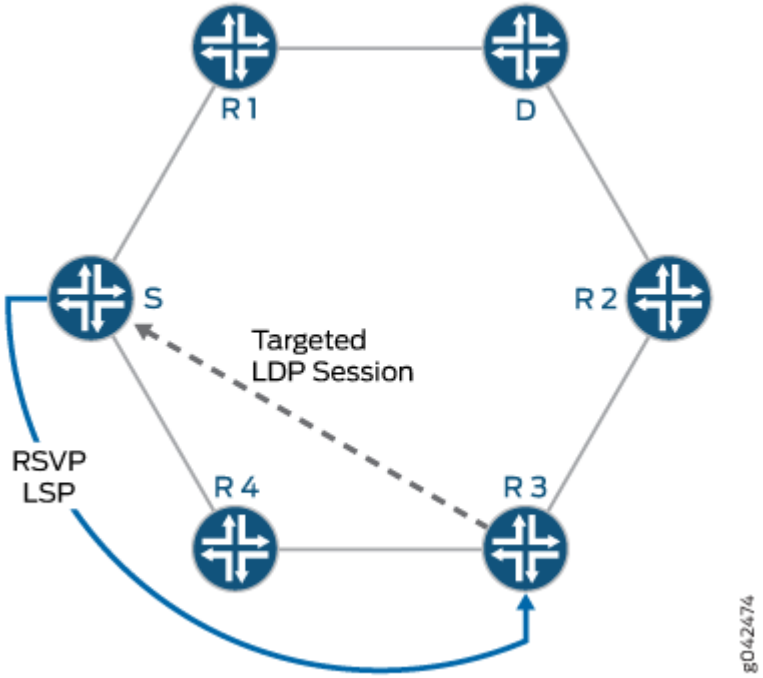
The key to protect the traffic flowing through LDP LSPs is to have an explicit tunnel to re-route the traffic in the event of a link or node failure. The explicit path has to terminate on the next downstream router, and the traffic needs to be accepted on the explicit path, where the RPF check should pass.

RSVP LSPs help overcome the current limitations of loop-free alternate (LFA) for both point-to-point and point-to-multipoint LDP LSPs by extending the LFA coverage in the following methods:

Manually Configured RSVP LSPs

Considering the example used in Figure 3, when the S-R1 link fails, and Router R4 is not an LFA for that particular link, a manually created RSVP LSP is used as a patch to provide complete LFA coverage. The RSVP LSP is pre-signaled and pre-installed in the Packet Forwarding Engine of Router S, so that it can be used as soon as Router S detects that the link has failed.

Figure 84: Manually Configured RSVP LSP Coverage



In this case, an RSVP LSP is created between Routers S, R4, and R3 as illustrated in Figure 4. A targeted LDP session is created between Router S and Router R3, as a result of which, when the S-R1 link fails, traffic reaches Router R3. Router R3 forwards the traffic to Router R2, as it is the shortest path to reach the destination, Router D.

Dynamically Configured RSVP LSPs

In this method, the RSVP LSPs are created automatically and pre-installed in the system so that they can be used immediately when there is a link failure. Here, the egress is the node on the other side of the link being protected, thereby improving the LFA coverage.

Benefits of Enabling Dynamic RSVP LSPs

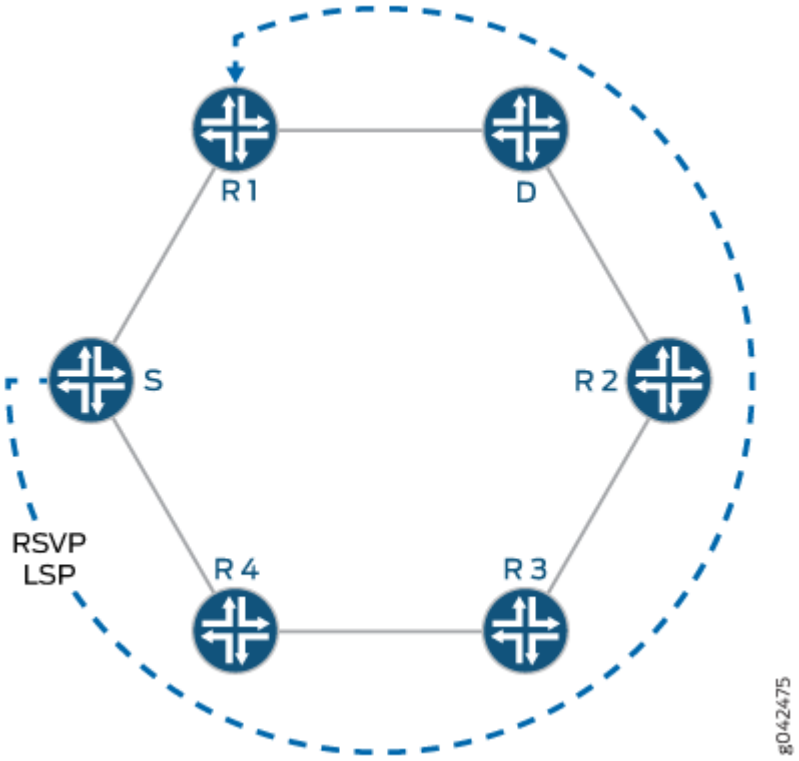
- Ease of configuration.
- 100 percent coverage against link failure as long as there is an alternate path to the far end of the link being protected.
- Setting up and tearing down of the RSVP bypass LSP is automatic.
- RSVP LSP only used for link protection and not for forwarding traffic while the link being protected is up.

- Reduces the total number of RSVP LSPs required on the system.

Considering the example used in Figure 3, in order to protect traffic against the potential failure of the S-R1 link, because Router R4 is not an LFA for that particular link, an RSVP bypass LSP is automatically created to Router R1, which is the node on the far side of the protected link as illustrated in Figure 5. From Router R1, traffic is forwarded to its original destination, Router D.

The RSVP LSP is pre-signaled and pre-installed in the Packet Forwarding Engine of Router S so that it can be used as soon as Router S detects that the link has failed.

Figure 85: Dynamically Configured RSVP LSP Coverage



An alternative mode of operation is not to use LFA at all, and to always have the RSVP LSP created to cover all link failures.

To enable dynamic RSVP LSPs, include the `dynamic-rsvp-lsp` statement at the `[edit protocols ldp interface` *`interface-name`* `link-protection]` hierarchy level, in addition to enabling the RSVP protocol on the appropriate interfaces.

Understanding Multicast LDP Link Protection

A point-to-multipoint LDP label-switched path (LSP) is an LDP-signaled LSP that is point-to-multipoint, and is referred to as multicast LDP.

A multicast LDP LSP can be used to send traffic from a single root or ingress node to a number of leaf or egress nodes traversing one or more transit nodes. Multicast LDP link protection enables fast reroute of traffic carried over point-to-multipoint LDP LSPs in case of a link failure. When one of the links of the point-to-multipoint tree fails, the subtrees might get detached until the IGP reconverges and the multipoint LSP is established using the best path from the downstream router to the new upstream router.

To protect the traffic flowing through the multicast LDP LSP, you can configure an explicit tunnel to re-route the traffic in the event of link failure. The explicit path has to terminate on the next downstream router. The reverse path forwarding for the traffic should be successful.

Multicast LDP link protection introduces the following features and functionality:

- Use of dynamic RSVP LSP as bypass tunnels

  The RSVP LSP's Explicit Route Object (ERO) is calculated using Constrained Shortest Path First (CSPF) with the constraint as the link to avoid. The LSP is signaled and torn down dynamically whenever link protection is necessary.

- Make-before-break

  The make-before-break feature ensures that there is minimum packet loss when attempting to signal a new LSP path before tearing down the old LSP path for the multicast LDP LSP.

- Targeted LDP session

  A targeted adjacency to the downstream label-switching router (LSR) is created for two reasons:

  - To keep the session up after link failure.
  - To use the point-to-multipoint label received from the session to send traffic to the downstream LSR on the RSVP LSP bypass tunnel.

    When the downstream LSR sets up the multicast LDP LSP with the root node and LSP ID, it uses that upstream LSR, which is on the best path toward the root.

NOTE: Multicast LDP link protection is not required when there are multiple link adjacencies (parallel links) to the downstream LSR.

Different Modes for Providing LDP Link Protection

Following are three different modes of operation available for unicast and multicast LDP link protection:

- **Case A: LFA only**

  Under this mode of operation, multicast LDP link protection is provided using an existing viable loop-free alternate (LFA). In the absence of a viable LFA, link protection is not provided for the multicast LDP LSP.

- **Case B: LFA and Dynamic RSVP LSP**

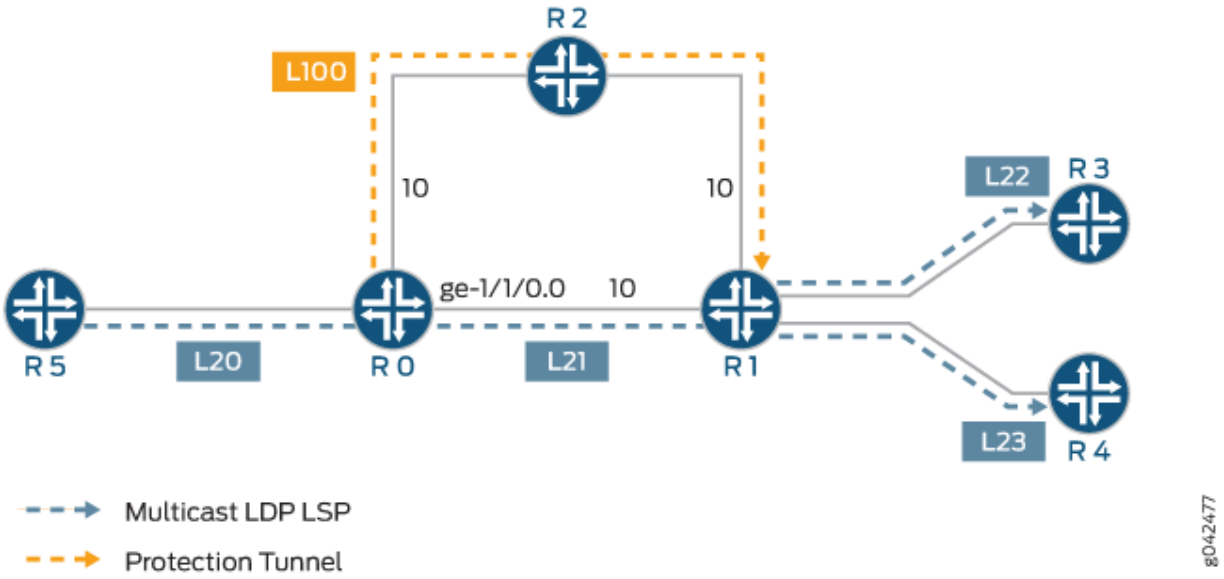
  Under this mode of operation, multicast LDP link protection is provided using an existing viable LFA. In the absence of a viable LFA, an RSVP bypass LSP is created automatically to provide link protection for the multicast LDP LSP.

- **Case C: Dynamic RSVP LSP only**

  Under this mode of operation, LFA is not used for link protection. Multicast LDP link protection is provided by using automatically created RSVP bypass LSP.

Figure 6 is a sample topology illustrating the different modes of operation for multicast LDP link protection. Router R5 is the root connecting to two leaf nodes, Routers R3 and R4. Router R0 and Router R1 are the upstream and downstream label-switched routers (LSRs), respectively. A multicast LDP LSP runs among the root and leaf nodes.

Figure 86: Multicast LDP Link Protection Sample Topology



Considering that Router R0 needs to protect the multicast LDP LSP in the case that the R0-R1 link fails, the different modes of link protection operate in the following manner:

- **Case A: LFA only**

  Router R0 checks if a viable LFA path exists that can avoid the R0-R1 link to reach Router R1. Based on the metrics, Router R2 is a valid LFA path for the R0-R1 link and is used to forward unicast LDP traffic. If multiple multicast LDP LSPs use the R0-R1 link, the same LFA (Router R2) is used for multicast LDP link protection.

  When the R0-R1 link fails, the multicast LDP LSP traffic is moved onto the LFA path by Router R0, and the unicast LDP label to reach Router R1 (L100) is pushed on top of the multicast LDP label (L21).

- **Case B: LFA and Dynamic RSVP LSP**

  Router R0 checks if a viable LFA path exists that can avoid the R0-R1 link to reach Router R1. Based on the metrics, Router R2 is a valid LFA path for the R0-R1 link and is used to forward unicast LDP traffic. If multiple multicast LDP LSPs use the R0-R1 link, the same LFA (Router R2) is used for multicast LDP link protection. When the R0-R1 link fails, the multicast LDP LSP traffic is moved onto the LFA path by Router R0.

  However, if the metric on the R2-R1 link was 50 instead of 10, Router 2 is a not a valid LFA for the R0-R1 link. In this case, an RSVP LSP is automatically created to protect the multicast LDP traffic traveling between Routers R0 and R1.

- **Case C: Dynamic RSVP LSP only**

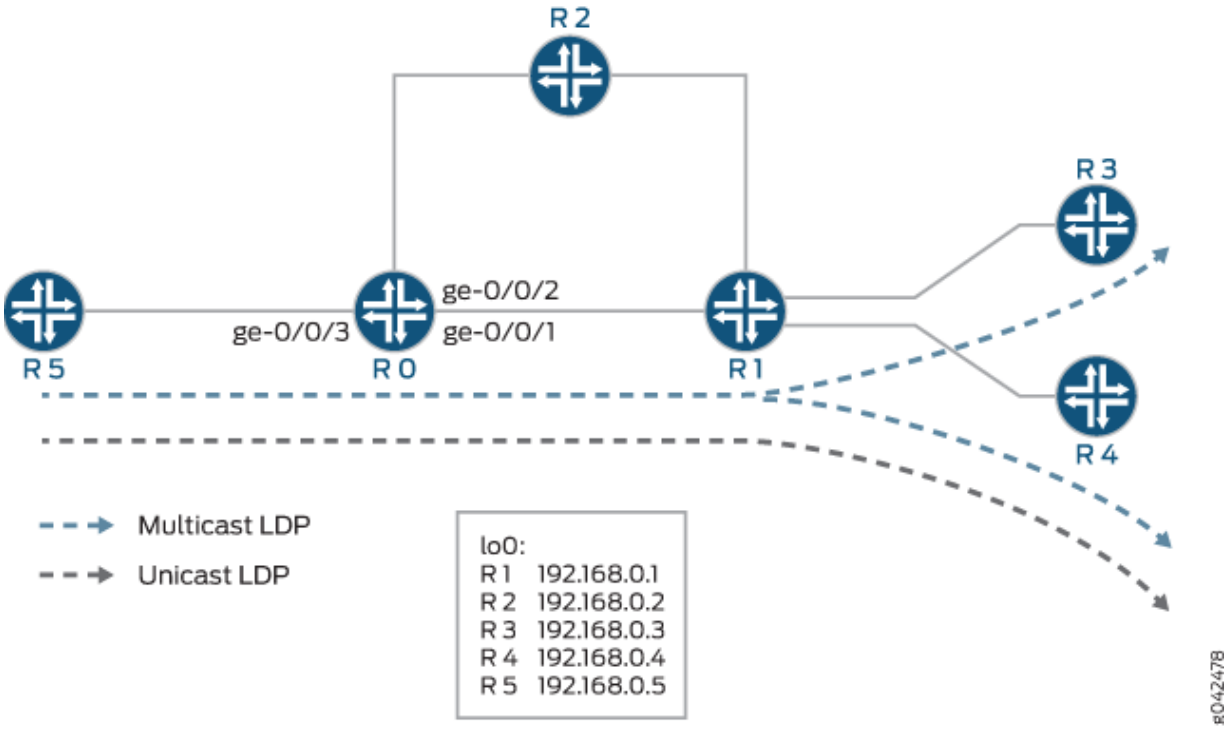
  An RSVP LSP is signaled automatically from Router R0 to Router R1 through Router R2, avoiding interface ge-1/1/0. If multiple multicast LDP LSPs use the R0-R1 link, the same RSVP LSP is used for multicast LDP link protection.

  When the R0-R1 link fails, the multicast LDP LSP traffic is moved onto the RSVP LSP by Router R0, and the RSVP label to reach Router R1 (L100) is pushed on top of the multicast LDP label (L21).

Label Operation for LDP Link Protection

Using the same network topology as in Figure 5, Figure 7 illustrates the label operation for unicast and multicast LDP link protection.

Figure 87: LDP Label Operation Sample Topology



Router R5 is the root connecting to two leaf nodes, Routers R3 and R4. Router R0 and Router R1 are the upstream and downstream label-switched routers (LSRs), respectively. A multicast LDP LSP runs among the root and leaf nodes. An unicast LDP path connects Router R1 to Router R5.

The label operation is performed differently under the following modes of LDP link protection:

Case A: LFA Only

Using the `show route detail` command output on Router R0, the unicast LDP traffic and multicast LDP traffic can be derived.

```
user@R0> show route detail
299840 (1 entry, 1 announced)
        *LDP    Preference: 9
                Next hop type: Router
                Address: 0x93bc22c
                Next-hop reference count: 1
                Next hop: 11.0.0.6 via ge-0/0/1.0 weight 0x1, selected
                Label operation: Swap 299824
                Session Id: 0x1
                Next hop: 11.0.0.10 via ge-0/0/2.0 weight 0xf000
                Label operation: Swap 299808
                Session Id: 0x3
                State: <Active Int>
                Age: 3:16       Metric: 1
                Validation State: unverified
                Task: LDP
                Announcement bits (1): 0-KRT
                AS path: I
                Prefixes bound to route: 192.168.0.4/32

299856 (1 entry, 1 announced)
        *LDP    Preference: 9
                Next hop type: Flood
                Address: 0x9340e04
                Next-hop reference count: 3
                Next hop type: Router, Next hop index: 262143
                Address: 0x93bc3dc
                Next-hop reference count: 2
                Next hop: 11.0.0.6 via ge-0/0/1.0 weight 0x1
                Label operation: Swap 299888
                Next hop: 11.0.0.10 via ge-0/0/2.0 weight 0xf000
                Label operation: Swap 299888, Push 299776(top)
                Label TTL action: prop-ttl, prop-ttl(top)
                State: <Active Int AckRequest>
                Age: 3:16       Metric: 1
                Validation State: unverified
                Task: LDP
```

```
                Announcement bits (1): 0-KRT
                AS path: I
                FECs bound to route: P2MP root-addr 192.168.0.5, lsp-id 99
```

Label 299840 is traffic arriving at Router R0 that corresponds to unicast LDP traffic to Router R1. Label 299856 is traffic arriving at Router 0 that corresponds to multicast LDP traffic from the root node R5 to the leaf egress nodes, R3 and R4.

The main path for both unicast and multicast LDP LSPs is through interface ge-0/0/1 (the link to Router R1), and the LFA path is through interface ge-0/0/2 (the link to Router R2). The LFA path is not used unless the ge-0/0/1 interface goes down.

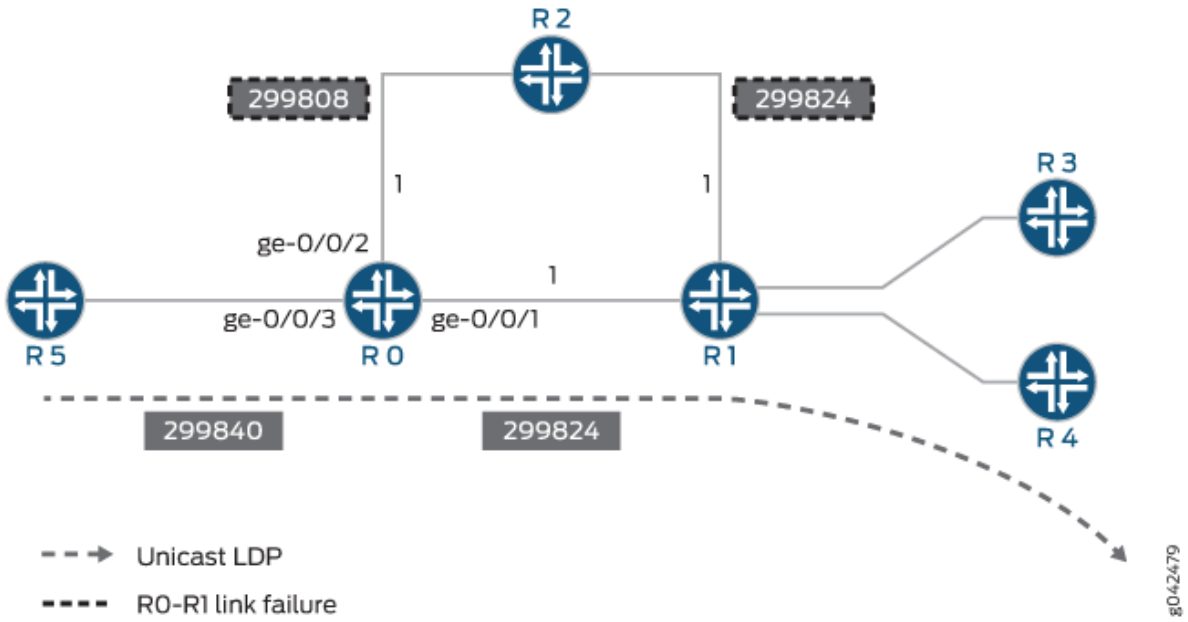
In the label operation for Case A, the LFA-only mode of operation is different for unicast and multicast LDP traffic:

- Unicast label operation

  For unicast LDP traffic, the FECs and associated labels are advertised on all the links in the network on which LDP is enabled. This means that in order to provide LFA action for the unicast LDP traffic to Router R4, instead of swapping the incoming label for label 299824 advertised by Router R1 for FEC R4, Router R0 simply swaps the incoming label for label 299808 advertised by Router R2 for FEC R4. This is the standard Junos OS LFA operation for unicast LDP traffic.

  Figure 8 illustrates the label operation for unicast traffic when the R0-R1 link fails. The grey boxes show the label operation for unicast LDP traffic under normal condition, and the dotted boxes show the label operation for unicast LDP traffic when the R0-R1 link fails.

Figure 88: Unicast LDP Label Operation



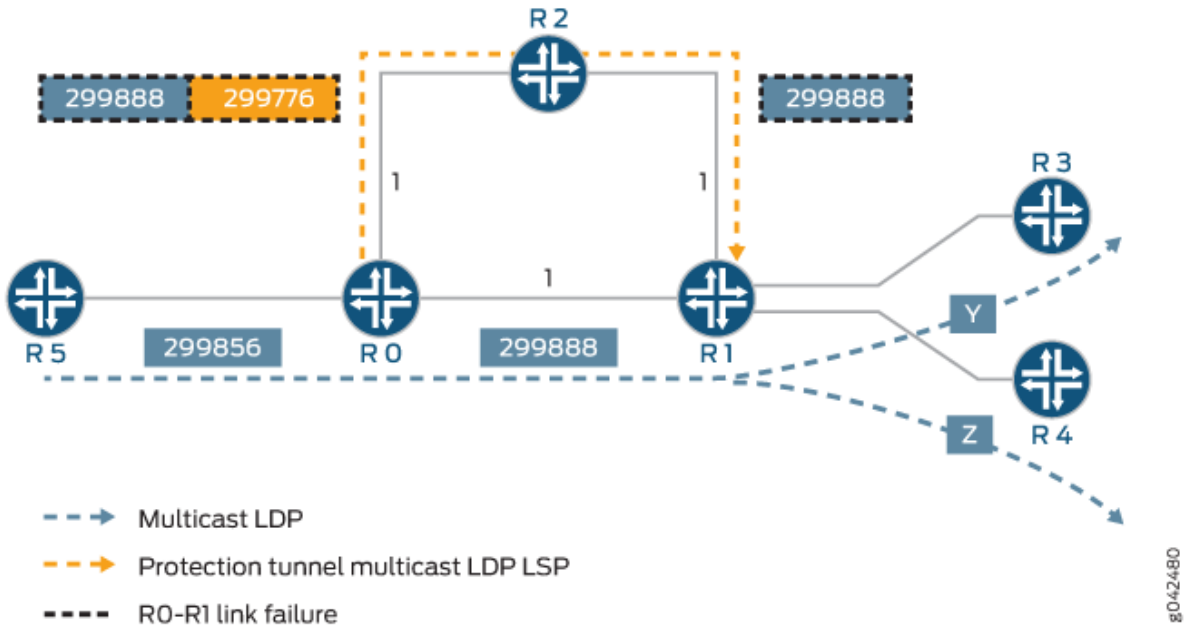
- Multicast label operation

  The label operation for multicast LDP traffic differs from the unicast LDP label operation, because multipoint LSP labels are only advertised along the best path from the leaf node to the ingress node. As a result, Router R2 has no knowledge of the multicast LDP. To overcome this, the multicast LDP LSP traffic is simply tunneled inside the unicast LDP LSP path through Router R2 that terminates at Router R1.

  In order to achieve this, Router R0 first swaps the incoming multicast LDP LSP label 299856 to label 299888 advertised by Router R1. Label 299776 is then pushed on top, which is the LDP label advertised by Router R2 for FEC R1. When the packet arrives at Router R2, the top label is popped out due to penultimate hop-popping. This means that the packet arrives at Router R1 with the multicast LDP label 299888 that Router R1 had originally advertised to Router R0.

  Figure 9 illustrates the label operation for multicast LDP traffic when the R0-R1 link fails. The blue boxes show the label operation for multicast LDP traffic under normal condition, and the dotted boxes show the label operation for multicast LDP traffic when the R0-R1 link fails.

Figure 89: Multicast LDP Label Operation



When the metric on the R2-R1 link is set to 1000 instead of 1, Router R2 is not a valid LFA for Router R0. In this case, if Router R2 receives a packet destined for Router R1, R3, or R4 before its IGP has converged, the packet is sent back to Router R0, resulting in looping packets.

Because Router R0 has no viable LFA, no backup paths are installed in the Packet Forwarding Engine. If the R0-R1 link fails, traffic flow is interrupted until the IGP and LDP converge and new entries are installed on the affected routers.

The `show route detail` command displays the state when no LFA is available for link protection.

```
user@host> show route detail
299840 (1 entry, 1 announced)
        *LDP    Preference: 9
                Next hop type: Router, Next hop index: 578
                Address: 0x9340d20
                Next-hop reference count: 2
                Next hop: 11.0.0.6 via ge-0/0/1.0, selected
                Label operation: Swap 299824
                Session Id: 0x1
                State: <Active Int>
                Age: 5:38       Metric: 1
                Validation State: unverified
```

```
                Task: LDP
                Announcement bits (1): 0-KRT
                AS path: I
                Prefixes bound to route: 192.168.0.4/32

299856 (1 entry, 1 announced)
        *LDP    Preference: 9
                Next hop type: Flood
                Address: 0x9340e04
                Next-hop reference count: 3
                Next hop type: Router, Next hop index: 579
                Address: 0x93407c8
                Next-hop reference count: 2
                Next hop: 11.0.0.6 via ge-0/0/1.0
                Label operation: Swap 299888
                State: <Active Int AckRequest>
                Age: 5:38       Metric: 1
                Validation State: unverified
                Task: LDP
                Announcement bits (1): 0-KRT
                AS path: I
                FECs bound to route: P2MP root-addr 192.168.0.5, lsp-id 99
```

Case B: LFA and Dynamic RSVP LSP

In this mode of operation, if there is a viable LFA neighbor, the label operation behavior is similar to that of Case A, LFA only mode. However, if there is no viable LFA neighbor, an RSVP bypass tunnel is automatically created.

If the metric on the link R2-R1 is set to 1000 instead of 1, Router R2 is not an LFA for Router R0. On learning that there are no LFA paths to protect the R0-R1 link failure, an RSVP bypass tunnel is automatically created with Router R1 as the egress node and follows a path that avoids the R0-R1 link (for instance, R0-R2-R1).

If the R0-R1 link fails, the unicast LDP and multicast LDP traffic is tunneled through the RSVP bypass tunnel. The RSVP bypass tunnel is not used for normal forwarding and is used only to provide link protection to LDP traffic in the case of R0-R1 link failure.

Using the `show route detail` command, the unicast and multicast LDP traffic can be derived.

```
user@host> show route detail
299840 (1 entry, 1 announced)
```

```
        *LDP    Preference: 9
                Next hop type: Router
                Address: 0x940c3dc
                Next-hop reference count: 1
                Next hop: 11.0.0.6 via ge-0/0/1.0 weight 0x1, selected
                Label operation: Swap 299824
                Session Id: 0x1
                Next hop: 11.0.0.10 via ge-0/0/2.0 weight 0x8001
                Label-switched-path ge-0/0/1.0:BypassLSP->192.168.0.1
                Label operation: Swap 299824, Push 299872(top)
                Label TTL action: prop-ttl, prop-ttl(top)
                Session Id: 0x3
                State: <Active Int NhAckRequest>
                Age: 19         Metric: 1
                Validation State: unverified
                Task: LDP
                Announcement bits (1): 0-KRT
                AS path: I
                Prefixes bound to route: 192.168.0.4/32

299856 (1 entry, 1 announced)
        *LDP    Preference: 9
                Next hop type: Flood
                Address: 0x9340e04
                Next-hop reference count: 3
                Next hop type: Router, Next hop index: 262143
                Address: 0x940c154
                Next-hop reference count: 2
                Next hop: 11.0.0.6 via ge-0/0/1.0 weight 0x1
                Label operation: Swap 299888
                Next hop: 11.0.0.10 via ge-0/0/2.0 weight 0x8001
                Label-switched-path ge-0/0/1.0:BypassLSP->192.168.0.1
                Label operation: Swap 299888, Push 299872(top)
                Label TTL action: prop-ttl, prop-ttl(top)
                State: < Active Int AckRequest>
                Age: 20         Metric: 1
                Validation State: unverified
                Task: LDP
                Announcement bits (1): 0-KRT
                AS path: I
                FECs bound to route: P2MP root-addr 192.168.0.5, lsp-id 99
```

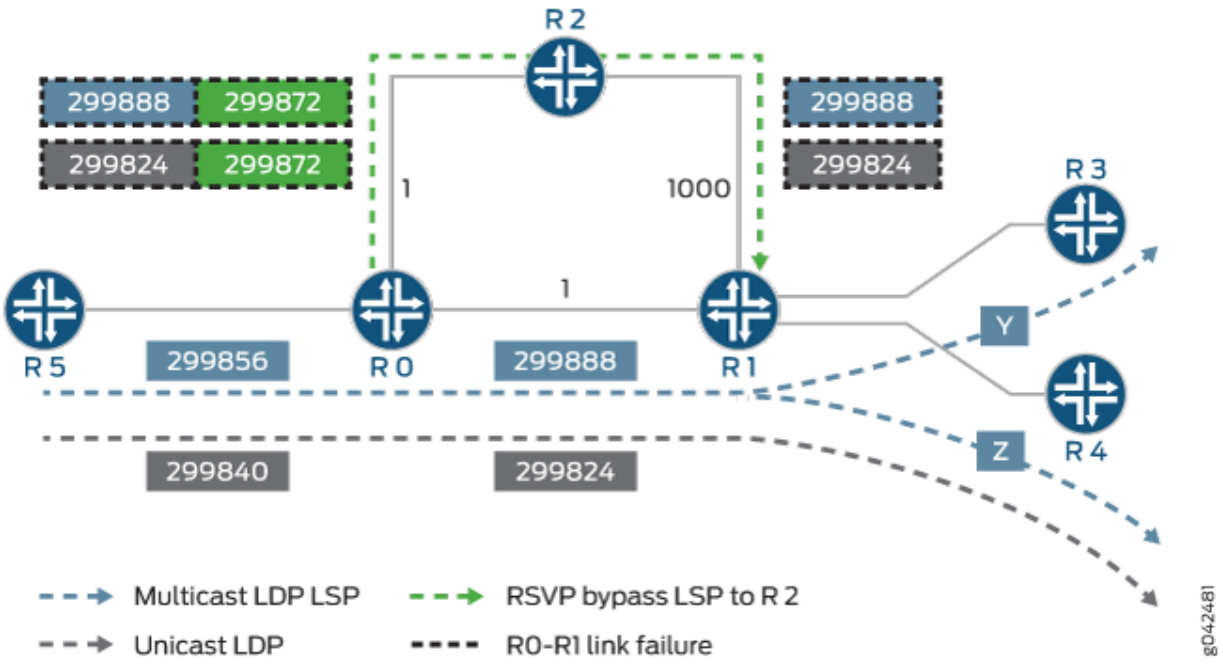
The main path for both unicast and multicast LDP LSP is through interface ge-0/0/1 (the link to Router R1), and the LFA path is through interface ge-0/0/2 (the link to Router R2). The LFA path is not used unless the ge-0/0/1 interface goes down.

Label 299840 is traffic arriving at Router R0 that corresponds to unicast LDP traffic to Router R4. Label 299856 is traffic arriving at Router 0 that corresponds to multicast LDP traffic from the root node R5 to the leaf egress nodes, R3 and R4.

As seen in the `show route detail` command output, the label operations for the protection path are the same for unicast LDP and multicast LDP traffic. The incoming LDP label at Router R0 is swapped to the LDP label advertised by Router R1 to Router R0. The RSVP label 299872 for the bypass tunnel is then pushed onto the packet. Penultimate hop-popping is used on the bypass tunnel, causing Router R2 to pop that label. Thus the packet arrives at Router R1 with the LDP label that it had originally advertised to Router R0.

Figure 10 illustrates the label operation for unicast LDP and multicast LDP traffic protected by the RSVP bypass tunnel. The grey and blue boxes represent label values used under normal conditions for unicast and multicast LDP traffic, respectively. The dotted boxes represent label values used when the R0-R1 link fails.

Figure 90: LDP Link Protection Label Operation



Case C: Dynamic RSVP LSP Only

In this mode of operation, LFA is not used at all. A dynamic RSVP bypass LSP is automatically created in order to provide link protection. The output from the `show route detail` command and the label operations are similar to Case B, LFA and dynamic RSVP LSP mode.

Sample Multicast LDP Link Protection Configuration

To enable multicast LDP link protection, the following configuration is required on Router R0:

NOTE: In this sample, multicast LDP link protection is enabled on the ge-1/0/0 interface of Router R0 that connects to Router R1, although typically all the interfaces need to be configured for link protection.

Router R0

```
protocols {
    rsvp {
        interface all;
        interface ge-0/0/0.0 {
            disable;
        }
    }
    mpls {
        interface all;
        interface ge-0/0/0.0 {
            disable;
        }
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface lo0.0;
            interface ge-0/0/1.0 {
                link-protection;
            }
            interface ge-0/0/2.0;
            interface ge-0/0/3.0;
        }
    }
    ldp {
```

```
        make-before-break {
            timeout seconds;
            switchover-delay seconds;
        }
        interface ge-1/1/0.0 {
            link-protection {
                disable;
                dynamic-rsvp-lsp;
            }
        }
    }
}
```

The following configuration statements apply to the different modes of multicast LDP protection as follows:

- `link-protection` statement at `[edit protocols ospf interface ge-0/0/1.0]`

  This configuration is applied only for Case A (LFA only) and Case B (LFA and dynamic RSVP LSP) modes of multicast LDP link protection. Configuring link protection under an IGP is not required for Case C (dynamic RSVP LSP only).

- `link-protection` statement at `[edit protocols ldp interface ge-0/0/1.0]`

  This configuration is required for all modes of multicast LDP protection. However, if the only LDP traffic present is unicast, and dynamic RSVP bypasses are not required, then this configuration is not required, as the `link-protection` statement at the `[edit protocols ospf interface ge-0/0/1.0]` hierarchy level results in LFA action for the LDP unicast traffic.

- `dynamic-rsvp-lsp` statement at `[edit protocols ldp interface ge-0/0/1.0 link-protection]`

  This configuration is applied only for Case B (LFA and dynamic RSVP LSP) and Case C (dynamic RSVP LSP only) modes of LDP link protection. Dynamic RSVP LSP configuration does not apply to Case A (LFA only).

Make-Before-Break

The make-before-break feature is enabled by default on Junos OS and provides some benefits for point-to-multipoint LSPs.

For a point-to-multipoint LSP, a label-switched router (LSR) selects the LSR that is its next hop to the root of the LSP as its upstream LSR. When the best path to reach the root changes, the LSR chooses a new upstream LSR. During this period, the LSP might be temporarily broken, resulting in packet loss until the LSP reconverges to a new upstream LSR. The goal of make-before-break in this case is to minimize the packet loss. In cases where the best path from the LSR to the root changes but the LSP

continues to forward traffic to the previous next hop to the root, a new LSP should be established before the old LSP is withdrawn to minimize the duration of packet loss.

Taking for example, after a link failure, a downstream LSR (for instance, LSR-D) still receives and/or forwards packets to the other downstream LSRs, as it continues to receive packets from the one hop RSVP LSP. Once routing converges, LSR-D selects a new upstream LSR (LSR-U) for this point-to-multipoint LSP's FEC (FEC-A). The new LSR might already be forwarding packets for FEC-A to the downstream LSRs other than LSR-D. After LSR-U receives a label for FEC-A from LSR-D, it notifies LSR-D when it has learnt that LSP for FEC-A has been established from the root to itself. When LSR-D receives such a notification, it changes its next hop for the LSP root to LSR-U. This is a route delete and add operation on LSR-D. At this point, LSR-D does an LSP switchover, and traffic tunneled through RSVP LSP or LFA is dropped, and traffic from LSR-U is accepted. The new transit route for LSR-U is added. The RPF check is changed to accept traffic from LSR-U and to drop traffic from the old upstream LSR, or the old route is deleted and the new route is added.

The assumption is that LSR-U has received a make-before-break notification from its upstream router for the FEC-A point-to-multipoint LSP and has installed a forwarding state for the LSP. At that point it should signal LSR-D by means of make-before-break notification that it has become part of the tree identified by FEC-A and that LSR-D should initiate its switchover to the LSP. Otherwise, LSR-U should remember that it needs to send notification to LSR-D when it receives a make-before-break notification from the upstream LSR for FEC-A and installs a forwarding state for this LSP. LSR-D continues to receive traffic from the old next hop to the root node using one hop RSVP LSP or LFA path until it switches over to the new point-to-multipoint LSP to LSR-U.

The make-before-break functionality with multicast LDP link protection includes the following features:

- Make-before-break capability

  An LSR advertises that it is capable of handling make-before-break LSPs using the capability advertisement. If the peer is not make-before-break capable, the make-before-break parameters are not sent to this peer. If an LSR receives a make-before-break parameter from a downstream LSR (LSR-D) but the upstream LSR (LSR-U) is not make-before-break capable, the LSR immediately sends a make-before-break notification to LSR-D, and the make-before-break capable LSP is not established. Instead, the normal LSP is established.

- Make-before-break status code

  The make-before-break status code includes:

  - 1—make-before-break request
  - 2—make-before-break acknowledgment

  When a downstream LSR sends a label-mapping message for point-to-multipoint LSP, it includes the make-before-break status code as 1 (request). When the upstream LSR updates the forwarding state for the point-to-multipoint LSP, it informs the downstream LSR with a notification message

containing the make-before-break status code as 2 (acknowledgment). At that point, the downstream LSR does an LSP switchover.

Caveats and Limitations

The Junos OS implementation of the LDP link protection feature has the following caveats and limitations:

- Make-before-break is not supported for the following point-to-multipoint LSPs on an egress LSR:
  - Next-generation multicast virtual private network (MVPN) with virtual routing and forwarding (VRF) label
  - Static LSP
- The following features are not supported:
  - Nonstop active routing for point-to-multipoint LSP in Junos OS Releases 12.3, 13.1 and 13.2
  - Graceful restart switchover point-to-multipoint LSP
  - Link protection for routing instance

Example: Configuring LDP Link Protection

IN THIS SECTION

- Requirements | **1436**
- Overview | **1437**
- Configuration | **1439**
- Verification | **1446**

This example shows how to configure Label Distribution Protocol (LDP) link protection for both unicast and multicast LDP label-switched paths (LSPs).

Requirements

This example uses the following hardware and software components:

- Six routers that can be a combination of M Series, MX Series, or T Series routers with one root node and two leaf nodes running a point-to-multipoint LDP LSP.
- Junos OS Release 12.3 or later running on all the routers.

Before you begin:

1. Configure the device interfaces.
2. Configure the following protocols:
   a. RSVP
   b. MPLS
   c. OSPF or any other IGP
   d. LDP

Overview

IN THIS SECTION

- Topology | **1438**

LDP link protection enables fast reroute of traffic carried over LDP LSPs in case of a link failure. LDP point-to-multipoint LSPs can be used to send traffic from a single root or ingress node to a number of leaf nodes or egress nodes traversing one or more transit nodes. When one of the links of the point-to-multipoint tree fails, the subtrees can get detached until the IGP reconverges and multicast LDP initiates label mapping using the best path from the downstream router to the new upstream router. To protect the traffic in the event of a link failure, you can configure an explicit tunnel so that traffic can be rerouted using the tunnel. Junos OS supports make-before-break capabilities to ensure minimum packet loss when attempting to signal a new LSP path before tearing down the old LSP path. This feature also adds targeted LDP support for multicast LDP link protection.

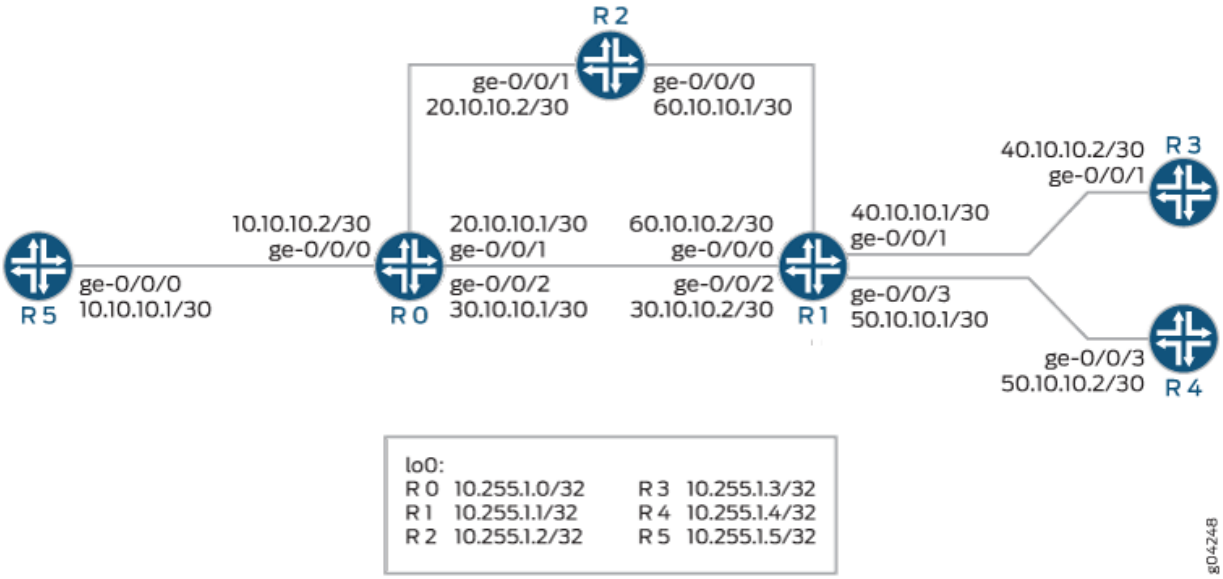
When configuring LDP link protection, be aware of the following considerations:

- Configure traffic engineering under IGP (if it is not supported by default), and include the interfaces configured for MPLS and RSVP so that constrained-based link protected dynamic RSVP LSP is signaled by RSVP using Constrained Shortest Path First (CSPF). When this condition is not satisfied, RSVP LSP might not come up and LDP cannot use it as a protected next hop.
- Configure a path between two label-switched routers (LSRs) to provide IP connectivity between the routers when there is a link failure. This enables CSPF to calculate an alternate path for link protection. When the connectivity between the routers is lost, the LDP targeted adjacency does not come up and dynamic RSVP LSP cannot be signaled, resulting in no protection for the LDP forwarding equivalence class (FEC) for which the peer is the downstream LSR.

- If link protection is active only on one LSR, then the other LSR should not be configured with the `strict-targeted-hellos` statement. This enables the LSR without link protection to allow asymmetric remote neighbor discovery and send periodic targeted hellos to the LSR that initiated the remote neighbor. When this condition is not satisfied, LDP targeted adjacency is not formed.
- LDP must be enabled on the loopback interface of the LSR to create remote neighbors based on LDP tunneling, LDP-based virtual private LAN service (VPLS), Layer 2 circuits, or LDP session protection. When this condition is not satisfied, LDP targeted adjacency is not formed.
- For unicast LDP LSP, loop-free alternate (LFA) should be configured in IGP.
- The ingress route to merge point should have at least one next hop avoiding the primary link between the merge point and the point of local repair for unicast LDP LSP.
- Point of local repair should have a unicast LDP label for the backup next hop to reach the merge point.

Topology

Figure 91: LDP Link Protection



In this example, Router R5 is the root connecting to two leaf nodes, Routers R3 and R4. Router R0 is the point of local repair.

Configuration

IN THIS SECTION

- CLI Quick Configuration | 1439
- Procedure | 1442

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

R5

```
set interfaces ge-0/0/0 unit 0 family inet address 10.10.10.1/30
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.1.5/32
set routing-options router-id 10.255.1.5
set routing-options autonomous-system 100
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols mpls traffic-engineering
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all metric 1
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ldp interface all link-protection dynamic-rsvp-lsp
set protocols ldp interface fxp0.0 disable
set protocols ldp p2mp
```

R0

```
set interfaces ge-0/0/0 unit 0 family inet address 10.10.10.2/30
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 20.10.10.1/30
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 30.10.10.1/30
```

```
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.1.0/32
set routing-options router-id 10.255.1.0
set routing-options autonomous-system 100
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols mpls traffic-engineering
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all metric 1
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ldp interface all link-protection dynamic-rsvp-lsp
set protocols ldp interface fxp0.0 disable
set protocols ldp p2mp
```

R1

```
set interfaces ge-0/0/0 unit 0 family inet address 60.10.10.2/30
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 40.10.10.1/30
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 30.10.10.2/30
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 family inet address 50.10.10.1/30
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.1.1/32
set routing-options router-id 10.255.1.1
set routing-options autonomous-system 100
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols mpls traffic-engineering
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all metric 1
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ldp interface all link-protection dynamic-rsvp-lsp
set protocols ldp interface fxp0.0 disable
set protocols ldp p2mp
```

R2

```
set interfaces ge-0/0/0 unit 0 family inet address 60.10.10.1/30
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 20.10.10.2/30
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.1.2/32
set routing-options router-id 10.255.1.2
set routing-options autonomous-system 100
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols mpls traffic-engineering
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ldp interface all link-protection dynamic-rsvp-lsp
set protocols ldp interface fxp0.0 disable
set protocols ldp p2mp
```

R3

```
set interfaces ge-0/0/1 unit 0 family inet address 40.10.10.2/30
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.1.3/32
set routing-options router-id 10.255.1.3
set routing-options autonomous-system 100
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols mpls traffic-engineering
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all metric 1
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ldp interface all link-protection dynamic-rsvp-lsp
set protocols ldp interface fxp0.0 disable
set protocols ldp p2mp root-address 10.255.1.5 lsp-id 1
```

R4

```
set interfaces ge-0/0/3 unit 0 family inet address 50.10.10.2/30
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.1.4/32
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols mpls traffic-engineering
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all metric 1
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ldp interface all link-protection dynamic-rsvp-lsp
set protocols ldp interface fxp0.0 disable
set protocols ldp p2mp root-address 10.255.1.5 lsp-id 1
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode*.

To configure Router R0:

1. Configure the Router R0 interfaces.

```
[edit interfaces]
user@R0# set ge-0/0/0 unit 0 family inet address 10.10.10.2/30
user@R0# set ge-0/0/0 unit 0 family mpls
user@R0# set ge-0/0/1 unit 0 family inet address 20.10.10.1/30
user@R0# set ge-0/0/1 unit 0 family mpls
user@R0# set ge-0/0/2 unit 0 family inet address 30.10.10.1/30
user@R0# set ge-0/0/2 unit 0 family mpls
user@R0# set lo0 unit 0 family inet address 10.255.1.0/32
```

2. Configure the router ID and autonomous system of Router R0.

```
[edit routing-options]
user@R0# set router-id 10.255.1.0
user@R0# set autonomous-system 100
```

3. Enable RSVP on all the interfaces of Router R0 (excluding the management interface).

```
[edit protocols]
user@R0# set rsvp interface all
user@R0# set rsvp interface fxp0.0 disable
```

4. Enable MPLS on all the interfaces of Router R0 (excluding the management interface) along with traffic engineering capabilities.

```
[edit protocols]
user@R0# set mpls traffic-engineering
user@R0# set mpls interface all
user@R0# set mpls interface fxp0.0 disable
```

5. Enable OSPF on all the interfaces of Router R0 (excluding the management interface), assign equal cost metric for the links, and enable traffic engineering capabilities.

```
[edit protocols]
user@R0# set ospf traffic-engineering
user@R0# set ospf area 0.0.0.0 interface all metric 1
user@R0# set ospf area 0.0.0.0 interface fxp0.0 disable
```

NOTE: For multicast LDP link protection with loop-free alternative (LFA), enable the following configuration under the `[edit protocols]` hierarchy level:

```
set ospf area 0 interface all link-protection
```

6. Enable LDP on all the interfaces of Router R0 (excluding the management interface) and configure link protection with dynamic RSVP bypass LSP.

```
[edit protocols]
user@R0# set ldp interface all link-protection dynamic-rsvp-lsp
user@R0# set ldp interface fxp0.0 disable
user@R0# set ldp p2mp
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R0# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 10.10.10.2/30;
        }
        family mpls;
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 20.10.10.1/30;
        }
        family mpls;
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 30.10.10.1/30;
        }
        family mpls;
    }
}
lo0 {
```

```
    unit 0 {
        family inet {
            address 10.255.1.0/32;
        }
    }
}
```

```
user@R0# show routing-options
router-id 10.255.1.0;
autonomous-system 100;
```

```
user@R0# show protocols
rsvp {
    interface all;
    interface fxp0.0 {
        disable;
    }
}
mpls {
    traffic-engineering;
    interface all;
    interface fxp0.0 {
        disable;
    }
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface all {
            metric 1;
        }
        interface fxp0.0 {
            disable;
        }
    }
}
ldp {
    interface all {
        link-protection {
            dynamic-rsvp-lsp;
```

```
        }
    }
    interface fxp0.0 {
        disable;
    }
    p2mp;
}
```

Verification

IN THIS SECTION

- Verifying the Bypass RSVP LSP Path | **1446**
- Verifying Label Operation | **1447**

Verify that the configuration is working properly.

Verifying the Bypass RSVP LSP Path

Purpose

Verify that the bypass RSVP LSP path has been created on the point of local repair (PLR).

Action

From operational mode, run the `show route tale mpls.0` command.

```
user@R0> show route tale mpls.0
mpls.0: 17 destinations, 17 routes (17 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 05:28:13, metric 1
                      Receive
1                  *[MPLS/0] 05:28:13, metric 1
                      Receive
2                  *[MPLS/0] 05:28:13, metric 1
                      Receive
13                 *[MPLS/0] 05:28:13, metric 1
                      Receive
```

```
299792             *[LDP/9] 00:41:41, metric 1
                    > to 30.10.10.2 via ge-0/0/2.0, Pop
299792(S=0)        *[LDP/9] 00:41:41, metric 1
                    > to 30.10.10.2 via ge-0/0/2.0, Pop
299808             *[LDP/9] 00:41:41, metric 1
                    > to 20.10.10.2 via ge-0/0/1.0, Pop
299808(S=0)        *[LDP/9] 00:41:41, metric 1
                    > to 20.10.10.2 via ge-0/0/1.0, Pop
299920             *[RSVP/7/1] 01:51:43, metric 1
                    > to 30.10.10.2 via ge-0/0/2.0, label-switched-path ge-0/0/0.0:BypassLSP-
>10.255.1.1
299920(S=0)        *[RSVP/7/1] 01:51:43, metric 1
                    > to 30.10.10.2 via ge-0/0/2.0, label-switched-path ge-0/0/0.0:BypassLSP-
>10.255.1.1
299936             *[RSVP/7/1] 01:51:25, metric 1
                    > to 20.10.10.2 via ge-0/0/1.0, label-switched-path ge-0/0/0.0:BypassLSP-
>10.255.1.2
299936(S=0)        *[RSVP/7/1] 01:51:25, metric 1
                    > to 20.10.10.2 via ge-0/0/1.0, label-switched-path ge-0/0/0.0:BypassLSP-
>10.255.1.2
299952             *[LDP/9] 00:06:11, metric 1
                    > to 10.10.10.1 via ge-0/0/0.0, Pop
299952(S=0)        *[LDP/9] 00:06:11, metric 1
                    > to 10.10.10.1 via ge-0/0/0.0, Pop
299968             *[LDP/9] 00:05:39, metric 1
                    > to 30.10.10.2 via ge-0/0/2.0, Swap 299984
299984             *[LDP/9] 00:05:38, metric 1
                    > to 30.10.10.2 via ge-0/0/2.0, Swap 300000
300000             *[LDP/9] 00:05:15, metric 1
                    > to 30.10.10.2 via ge-0/0/2.0, Swap 300016
```

Meaning

When the R0-R1 link goes down, the RSVP bypass LSP is used to route traffic.

Verifying Label Operation

Purpose

Verify the label swapping at the PLR.

Action

From operational mode, run the `show route table mpls.0 label label extensive` command.

```
user@R0> show route table mpls.0 label 300000 extensive
mpls.0: 17 destinations, 17 routes (17 active, 0 holddown, 0 hidden)
300000 (1 entry, 1 announced)
TSI:
KRT in-kernel 300000 /52 -> {Swap 300016}
        *LDP    Preference: 9
                Next hop type: Router, Next hop index: 589
                Address: 0x9981610
                Next-hop reference count: 2
                Next hop: 30.10.10.2 via ge-0/0/2.0, selected
                Label operation: Swap 300016
                Load balance label: Label 300016: None;
                Session Id: 0x2
                State: <Active Int>
                Local AS:   100
                Age: 12:50      Metric: 1
                Validation State: unverified
                Task: LDP
                Announcement bits (1): 1-KRT
                AS path: I
                Prefixes bound to route: 10.255.1.4/32
```

Meaning

The label is bound to reach Router R4, which is a leaf node.

Understanding Multicast-Only Fast Reroute

IN THIS SECTION

- MoFRR Overview | **1449**
- PIM Functionality | **1451**
- Multipoint LDP Functionality | **1452**
- Packet Forwarding | **1453**

- Limitations and Caveats | 1455

Multicast-only fast reroute (MoFRR) minimizes packet loss for traffic in a multicast distribution tree when link failures occur, enhancing multicast routing protocols like Protocol Independent Multicast (PIM) and multipoint Label Distribution Protocol (multipoint LDP) on devices that support these features.

NOTE: On switches, MoFRR with MPLS label-switched paths and multipoint LDP is not supported.

For MX Series routers, MoFRR is supported only on MX Series routers with MPC line cards. As a prerequisite, you must configure the router into `network-services enhanced-ip` mode, and all the line cards in the router must be MPCs.

With MoFRR enabled, devices send join messages on primary and backup upstream paths toward a multicast source. Devices receive data packets from both the primary and backup paths, and discard the redundant packets based on priority (weights that are assigned to the primary and backup paths). When a device detects a failure on the primary path, it immediately starts accepting packets from the secondary interface (the backup path). The fast switchover greatly improves convergence times upon primary path link failures.

One application for MoFRR is streaming IPTV. IPTV streams are multicast as UDP streams, so any lost packets are not retransmitted, leading to a less-than-satisfactory user experience. MoFRR can improve the situation.

MoFRR Overview

With fast reroute on unicast streams, an upstream routing device preestablishes MPLS label-switched paths (LSPs) or precomputes an IP loop-free alternate (LFA) fast reroute backup path to handle failure of a segment in the downstream path.

In multicast routing, the receiving side usually originates the traffic distribution graphs. This is unlike unicast routing, which generally establishes the path from the source to the receiver. PIM (for IP), multipoint LDP (for MPLS), and RSVP-TE (for MPLS) are protocols that are capable of establishing multicast distribution graphs. Of these, PIM and multipoint LDP receivers initiate the distribution graph setup, so MoFRR can work with these two multicast protocols where they are supported.

In a multicast tree, if the device detects a network component failure, it takes some time to perform a reactive repair, leading to significant traffic loss while setting up an alternate path. MoFRR reduces traffic loss in a multicast distribution tree when a network component fails. With MoFRR, one of the downstream routing devices sets up an alternative path toward the source to receive a backup live

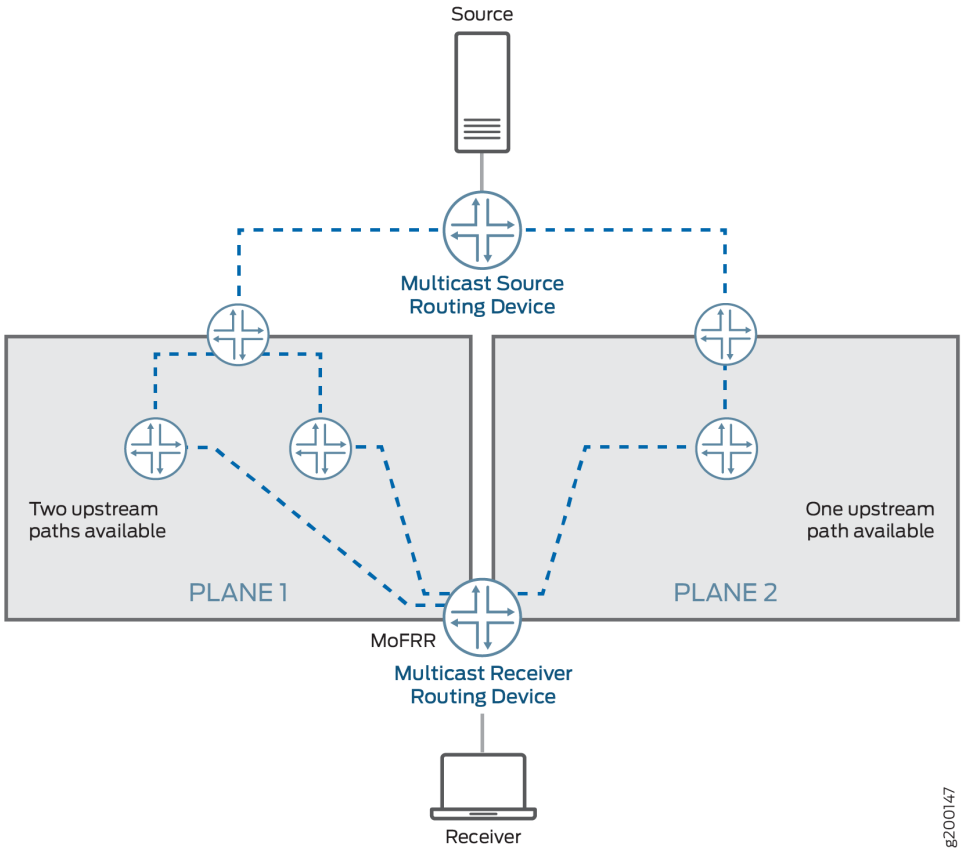
stream of the same multicast traffic. When a failure happens along the primary stream, the MoFRR routing device can quickly switch to the backup stream.

With MoFRR enabled, for each (S,G) entry, the device uses two of the available upstream interfaces to send a join message and to receive multicast traffic. The protocol attempts to select two disjoint paths if two such paths are available. If disjoint paths are not available, the protocol selects two non-disjoint paths. If two non-disjoint paths are not available, only a primary path is selected with no backup. MoFRR prioritizes the disjoint backup in favor of load balancing the available paths.

MoFRR is supported for both IPv4 and IPv6 protocol families.

Figure 92 on page 1450 shows two paths from the multicast receiver routing device (also referred to as the egress provider edge (PE) device) to the multicast source routing device (also referred to as the ingress PE device).

Figure 92: MoFRR Sample Topology



With MoFRR enabled, the egress (receiver side) routing device sets up two multicast trees, a primary path and a backup path, toward the multicast source for each (S,G). In other words, the egress routing

device propagates the same (S,G) join messages toward two different upstream neighbors, thus creating two multicast trees.

One of the multicast trees goes through plane 1 and the other through plane 2, as shown in Figure 92 on page 1450. For each (S,G), the egress routing device forwards traffic received on the primary path and drops traffic received on the backup path.

MoFRR is supported on both equal-cost multipath (ECMP) paths and non-ECMP paths. The device needs to enable unicast loop-free alternate (LFA) routes to support MoFRR on non-ECMP paths. You enable LFA routes using the `link-protection` statement in the interior gateway protocol (IGP) configuration. When you enable link protection on an OSPF or IS-IS interface, the device creates a backup LFA path to the primary next hop for all destination routes that traverse the protected interface.

Junos OS implements MoFRR in the IP network for IP MoFRR and at the MPLS label-edge routing device (LER) for multipoint LDP MoFRR.

Multipoint LDP MoFRR is used at the egress device of an MPLS network, where the packets are forwarded to an IP network. With multipoint LDP MoFRR, the device establishes two paths toward the upstream PE routing device for receiving two streams of MPLS packets at the LER. The device accepts one of the streams (the primary), and the other one (the backup) is dropped at the LER. IF the primary path fails, the device accepts the backup stream instead. Inband signaling support is a prerequisite for MoFRR with multipoint LDP (see Understanding Multipoint LDP Inband Signaling for Point-to-Multipoint LSPs).

PIM Functionality

Junos OS supports MoFRR for shortest-path tree (SPT) joins in PIM source-specific multicast (SSM) and any-source multicast (ASM). MoFRR is supported for both SSM and ASM ranges. To enable MoFRR for (*,G) joins, include the `mofrr-asm-starg` configuration statement at the `[edit routing-options multicast stream-protection]` hierarchy. For each group G, MoFRR will operate for either (S,G) or (*,G), but not both. (S,G) always takes precedence over (*,G).

With MoFRR enabled, a PIM routing device propagates join messages on two upstream reverse-path forwarding (RPF) interfaces to receive multicast traffic on both links for the same join request. MoFRR gives preference to two paths that do not converge to the same immediate upstream routing device. PIM installs appropriate multicast routes with upstream RPF next hops with two interfaces (for the primary and backup paths).

When the primary path fails, the backup path is upgraded to primary status, and the device forwards traffic accordingly. If there are alternate paths available, MoFRR calculates a new backup path and updates or installs the appropriate multicast route.

You can enable MoFRR with PIM join load balancing (see the `join-load-balance automatic` statement). However, in that case the distribution of join messages among the links might not be even. When a new

ECMP link is added, join messages on the primary path are redistributed and load-balanced. The join messages on the backup path might still follow the same path and might not be evenly redistributed.

You enable MoFRR using the `stream-protection` configuration statement at the `[edit routing-options multicast]` hierarchy. MoFRR is managed by a set of filter policies.

When an egress PIM routing device receives a join message or an IGMP report, it checks for an MoFRR configuration and proceeds as follows:

- If the MoFRR configuration is not present, PIM sends a join message upstream toward one upstream neighbor (for example, plane 2 in Figure 92 on page 1450).
- If the MoFRR configuration is present, the device checks for a policy configuration.
- If a policy is not present, the device checks for primary and backup paths (upstream interfaces), and proceeds as follows:
  - If primary and backup paths are not available—PIM sends a join message upstream toward one upstream neighbor (for example, plane 2 in Figure 92 on page 1450).
  - If primary and backup paths are available—PIM sends the join message upstream toward two of the available upstream neighbors. Junos OS sets up primary and secondary multicast paths to receive multicast traffic (for example, plane 1 in Figure 92 on page 1450).
- If a policy is present, the device checks whether the policy allows MoFRR for this (S,G), and proceeds as follows:
  - If this policy check fails—PIM sends a join message upstream toward one upstream neighbor (for example, plane 2 in Figure 92 on page 1450).
  - If this policy check passes—The device checks for primary and backup paths (upstream interfaces).
    - If the primary and backup paths are not available, PIM sends a join message upstream toward one upstream neighbor (for example, plane 2 in Figure 92 on page 1450).
    - If the primary and backup paths are available, PIM sends the join message upstream toward two of the available upstream neighbors. The device sets up primary and secondary multicast paths to receive multicast traffic (for example, plane 1 in Figure 92 on page 1450).

Multipoint LDP Functionality

To avoid MPLS traffic duplication, multipoint LDP usually selects only one upstream path. (See section 2.4.1.1. Determining One's 'upstream LSR' in RFC 6388, *Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths*.)

For multipoint LDP with MoFRR, the multipoint LDP device selects two separate upstream peers and sends two separate labels, one to each upstream peer. The device uses the same algorithm described in

RFC 6388 to select the primary upstream path. The device uses the same algorithm to select the backup upstream path but excludes the primary upstream LSR as a candidate. The two different upstream peers send two streams of MPLS traffic to the egress routing device. The device selects only one of the upstream neighbor paths as the primary path from which to accept the MPLS traffic. The other path becomes the backup path, and the device drops that traffic. When the primary upstream path fails, the device starts accepting traffic from the backup path. The multipoint LDP device selects the two upstream paths based on the interior gateway protocol (IGP) root device next hop.

A *forwarding equivalency class (FEC)* is a group of IP packets that are forwarded in the same manner, over the same path, and with the same forwarding treatment. Normally, the label that is put on a particular packet represents the FEC to which that packet is assigned. In MoFRR, two routes are placed into the mpls.0 table for each FEC—one route for the primary label and the other route for the backup label.

If there are parallel links toward the same immediate upstream device, the device considers both parallel links to be the primary. At any point in time, the upstream device sends traffic on only one of the multiple parallel links.

A *bud node* is an LSR that is an egress LSR, but also has one or more directly connected downstream LSRs. For a bud node, the traffic from the primary upstream path is forwarded to a downstream LSR. If the primary upstream path fails, the MPLS traffic from the backup upstream path is forwarded to the downstream LSR. This means that the downstream LSR next hop is added to both MPLS routes along with the egress next hop.

As with PIM, you enable MoFRR with multipoint LDP using the `stream-protection` configuration statement at the `[edit routing-options multicast]` hierarchy, and it's managed by a set of filter policies.

If you have enabled the multipoint LDP point-to-multipoint FEC for MoFRR, the device factors the following considerations into selecting the upstream path:

- The targeted LDP sessions are skipped if there is a nontargeted LDP session. If there is a single targeted LDP session, the targeted LDP session is selected, but the corresponding point-to-multipoint FEC loses the MoFRR capability because there is no interface associated with the targeted LDP session.
- All interfaces that belong to the same upstream LSR are considered to be the primary path.
- For any root-node route updates, the upstream path is changed based on the latest next hops from the IGP. If a better path is available, multipoint LDP attempts to switch to the better path.

Packet Forwarding

For either PIM or multipoint LDP, the device performs multicast source stream selection at the ingress interface. This preserves fabric bandwidth and maximizes forwarding performance because it:

- Avoids sending out duplicate streams across the fabric

- Prevents multiple route lookups (that result in packet drops).

For PIM, each IP multicast stream contains the same destination address. Regardless of the interface on which the packets arrive, the packets have the same route. The device checks the interface upon which each packet arrives and forwards only those that are from the primary interface. If the interface matches a backup stream interface, the device drops the packets. If the interface doesn't match either the primary or backup stream interface, the device handles the packets as exceptions in the control plane.

Figure 93 on page 1454 shows this process with sample primary and backup interfaces for routers with PIM. Figure 94 on page 1454 shows this similarly for switches with PIM.

Figure 93: MoFRR IP Route Lookup in the Packet Forwarding Engine on Routers

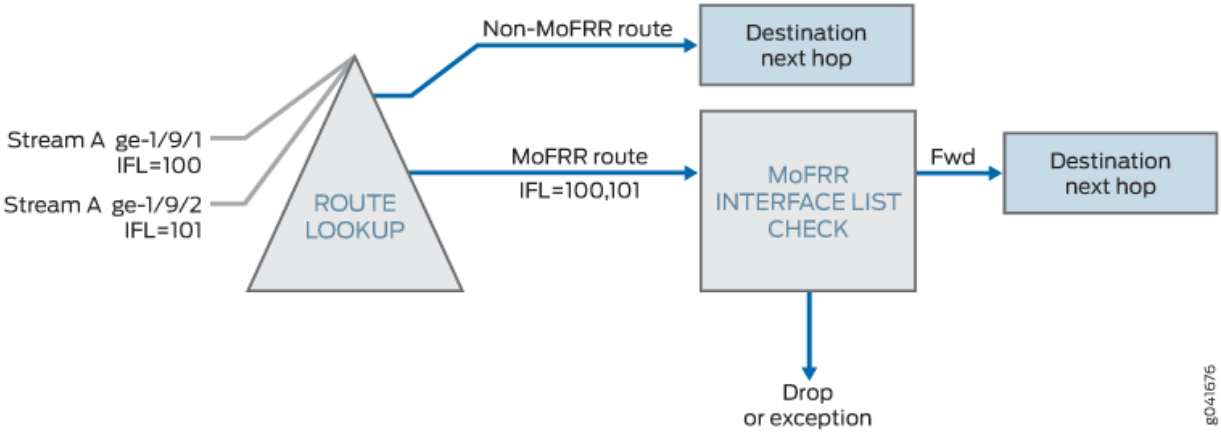
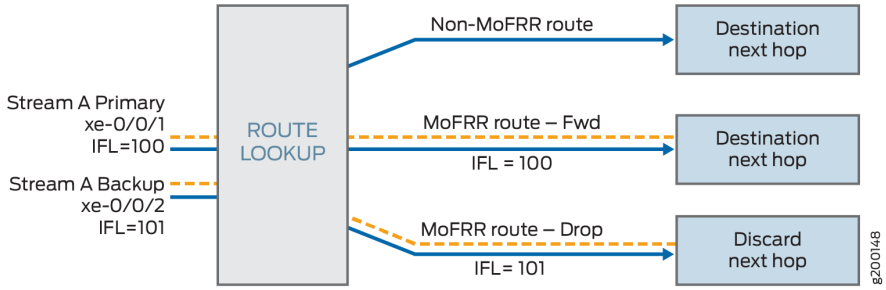


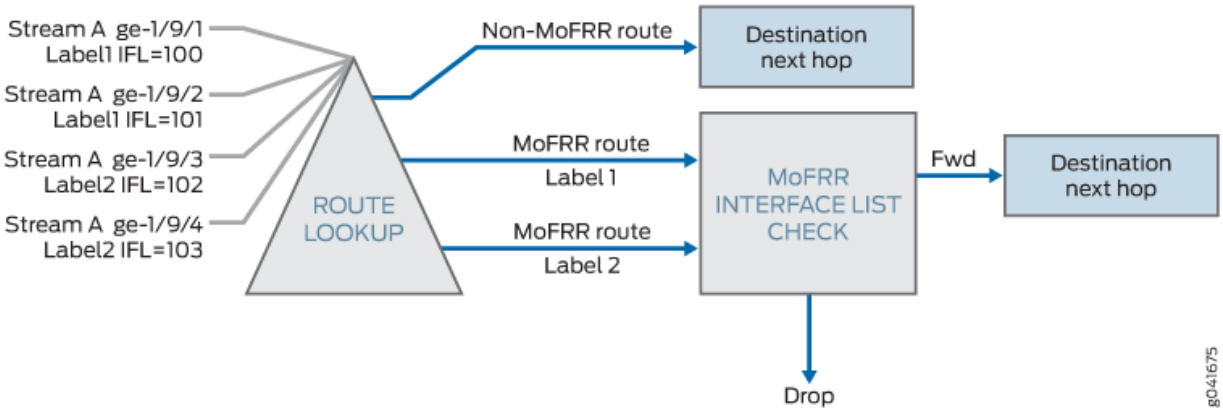
Figure 94: MoFRR IP Route Handling in the Packet Forwarding Engine on Switches



For MoFRR with multipoint LDP on routers, the device uses multiple MPLS labels to control MoFRR stream selection. Each label represents a separate route, but each references the same interface list check. The device only forwards the primary label, and drops all others. Multiple interfaces can receive packets using the same label.

Figure 95 on page 1455 shows this process for routers with multipoint LDP.

Figure 95: MoFRR MPLS Route Lookup in the Packet Forwarding Engine



Limitations and Caveats

MoFRR Limitations and Caveats on Switching and Routing Devices

MoFRR has the following limitations and caveats on routing and switching devices:

- MoFRR failure detection is supported for immediate link protection of the routing device on which MoFRR is enabled and not on all the links (end-to-end) in the multicast traffic path.
- MoFRR supports fast reroute on two selected disjoint paths toward the source. Two of the selected upstream neighbors cannot be on the same interface—in other words, two upstream neighbors on a LAN segment. The same is true if the upstream interface happens to be a multicast tunnel interface.
- Detection of the maximum end-to-end disjoint upstream paths is not supported. The receiver side (egress) routing device only makes sure that there is a disjoint upstream device (the immediate previous hop). PIM and multipoint LDP do not support the equivalent of explicit route objects (EROs). Hence, disjoint upstream path detection is limited to control over the immediately previous hop device. Because of this limitation, the path to the upstream device of the previous hop selected as primary and backup might be shared.
- You might see some traffic loss in the following scenarios:
  - A better upstream path becomes available on an egress device.
  - MoFRR is enabled or disabled on the egress device while there is an active traffic stream flowing.
- PIM join load balancing for join messages for backup paths are not supported.

- For a multicast group G, MoFRR is not allowed for both (S,G) and (*,G) join messages. (S,G) join messages have precedence over (*,G).
- MoFRR is not supported for multicast traffic streams that use two different multicast groups. Each (S,G) combination is treated as a unique multicast traffic stream.
- The bidirectional PIM range is not supported with MoFRR.
- PIM dense-mode is not supported with MoFRR.
- Multicast statistics for the backup traffic stream are not maintained by PIM and therefore are not available in the operational output of `show` commands.
- Rate monitoring is not supported.

MoFRR Limitations on Switching Devices with PIM

MoFRR with PIM has the following limitations on switching devices:

- MoFRR is not supported when the upstream interface is an integrated routing and bridging (IRB) interface, which impacts other multicast features such as Internet Group Management Protocol version 3 (IGMPv3) snooping.
- Packet replication and multicast lookups while forwarding multicast traffic can cause packets to recirculate through PFEs multiple times. As a result, displayed values for multicast packet counts from the `show pfe statistics traffic` command might show higher numbers than expected in output fields such as `Input packets` and `Output packets`. You might notice this behavior more frequently in MoFRR scenarios because duplicate primary and backup streams increase the traffic flow in general.

MoFRR Limitations and Caveats on Routing Devices with Multipoint LDP

MoFRR has the following limitations and caveats on routers when used with multipoint LDP:

- MoFRR does not apply to multipoint LDP traffic received on an RSVP tunnel because the RSVP tunnel is not associated with any interface.
- Mixed upstream MoFRR is not supported. This refers to PIM multipoint LDP in-band signaling, wherein one upstream path is through multipoint LDP and the second upstream path is through PIM.
- Multipoint LDP labels as inner labels are not supported.
- If the source is reachable through multiple ingress (source-side) provider edge (PE) routing devices, multipoint LDP MoFRR is not supported.
- Targeted LDP upstream sessions are not selected as the upstream device for MoFRR.

- Multipoint LDP link protection on the backup path is not supported because there is no support for MoFRR inner labels.

Configuring Multicast-Only Fast Reroute

You can configure multicast-only fast reroute (MoFRR) to minimize packet loss in a network when there is a link failure.

When fast reroute is applied to unicast streams, an upstream router preestablishes MPLS label-switched paths (LSPs) or precomputes an IP loop-free alternate (LFA) fast reroute backup path to handle failure of a segment in the downstream path.

In multicast routing, the traffic distribution graphs are usually originated by the receiver. This is unlike unicast routing, which usually establishes the path from the source to the receiver. Protocols that are capable of establishing multicast distribution graphs are PIM (for IP), multipoint LDP (for MPLS) and RSVP-TE (for MPLS). Of these, PIM and multipoint LDP receivers initiate the distribution graph setup, and therefore:

- On the QFX series, MoFRR is supported in PIM domains.
- On the MX Series and SRX Series, MoFRR is supported in PIM and multipoint LDP domains.

The configuration steps are the same for enabling MoFRR for PIM on all devices that support this feature, unless otherwise indicated. Configuration steps that are not applicable to multipoint LDP MoFRR are also indicated.

(For MX Series routers only) MoFRR is supported on MX Series routers with MPC line cards. As a prerequisite,all the line cards in the router must be MPCs.

To configure MoFRR on routers or switches:

1. (For MX Series and SRX Series routers only) Set the router to enhanced IP mode.

```
[edit chassis]
user@host# set network-services enhanced-ip
```

2. Enable MoFRR.

```
[edit routing-options multicast]
user@host# set stream-protection
```

3. (Optional) Configure a routing policy that filters for a restricted set of multicast streams to be affected by your MoFRR configuration.

   You can apply filters that are based on source or group addresses.

For example:

```
[edit policy-options]
policy-statement mofrr-select {
    term A {
        from {
            source-address-filter 225.1.1.1/32 exact;
        }
        then {
            accept;
        }
    }
    term B {
        from {
            source-address-filter 226.0.0.0/8 orlonger;
        }
        then {
            accept;
        }
    }
    term C {
        from {
            source-address-filter 227.1.1.0/24 orlonger;
            source-address-filter 227.4.1.0/24 orlonger;
            source-address-filter 227.16.1.0/24 orlonger;
        }
        then {
            accept;
        }
    }
    term D {
        from {
            source-address-filter 227.1.1.1/32 exact
        }
        then {
            reject; #MoFRR disabled
        }
    }
    ...
}
```

4. (Optional) If you configured a routing policy to filter the set of multicast groups to be affected by your MoFRR configuration, apply the policy for MoFRR stream protection.

```
[edit routing-options multicast stream-protection]
user@host# set policy policy-name
```

For example:

```
routing-options {
    multicast {
        stream-protection {
            policy mofrr-select
        }
    }
}
```

5. (Optional) In a PIM domain with MoFRR, allow MoFRR to be applied to any-source multicast (ASM) (*,G) joins.

This is not supported for multipoint LDP MoFRR.

```
[edit routing-options multicast stream-protection]
user@host# set mofrr-asm-starg
```

6. (Optional) In a PIM domain with MoFRR, allow only a disjoint RPF (an RPF on a separate plane) to be selected as the backup RPF path.

This is not supported for multipoint LDP MoFRR. In a multipoint LDP MoFRR domain, the same label is shared between parallel links to the same upstream neighbor. This is not the case in a PIM domain, where each link forms a neighbor. The `mofrr-disjoint-upstream-only` statement does not allow a backup RPF path to be selected if the path goes to the same upstream neighbor as that of the primary RPF path. This ensures that MoFRR is triggered only on a topology that has multiple RPF upstream neighbors.

```
[edit routing-options multicast stream-protection]
user@host# set mofrr-disjoint-upstream-only
```

7. (Optional) In a PIM domain with MoFRR, prevent sending join messages on the backup path, but retain all other MoFRR functionality.

This is not supported for multipoint LDP MoFRR.

```
[edit routing-options multicast stream-protection]
user@host# set mofrr-no-backup-join
```

8. (Optional) In a PIM domain with MoFRR, allow new primary path selection to be based on the unicast gateway selection for the unicast route to the source and to change when there is a change in the unicast selection, rather than having the backup path be promoted as primary. This ensures that the primary RPF hop is always on the best path.

   When you include the `mofrr-primary-selection-by-routing` statement, the backup path is not guaranteed to get promoted to be the new primary path when the primary path goes down.

   This is not supported for multipoint LDP MoFRR.

```
[edit routing-options multicast stream-protection]
user@host# set mofrr-primary-path-selection-by-routing
```

Example: Configuring Multicast-Only Fast Reroute in a Multipoint LDP Domain

IN THIS SECTION

- Requirements | **1461**
- Overview | **1461**
- CLI Quick Configuration | **1462**
- Configuration | **1471**
- Verification | **1478**

This example shows how to configure multicast-only fast reroute (MoFRR) to minimize packet loss in a network when there is a link failure.

Multipoint LDP MoFRR is used at the egress node of an MPLS network, where the packets are forwarded to an IP network. In the case of multipoint LDP MoFRR, the two paths toward the upstream provider edge (PE) router are established for receiving two streams of MPLS packets at the label-edge router (LER). One of the streams (the primary) is accepted, and the other one (the backup) is dropped at the LER. The backup stream is accepted if the primary path fails.

Requirements

No special configuration beyond device initialization is required before configuring this example.

In a multipoint LDP domain, for MoFRR to work, only the egress PE router needs to have MoFRR enabled. The other routers do not need to support MoFRR.

MoFRR is supported on MX Series platforms with MPC line cards. As a prerequisite, the router must be set to `network-services enhanced-ip` mode, and all the line-cards in the platform must be MPCs.

This example requires Junos OS Release 14.1 or later on the egress PE router.

Overview

IN THIS SECTION

- Topology | **1461**

In this example, Device R3 is the egress edge router. MoFRR is enabled on this device only.

OSPF is used for connectivity, though any interior gateway protocol (IGP) or static routes can be used.

For testing purposes, routers are used to simulate the source and the receiver. Device R4 and Device R8 are configured to statically join the desired group by using the `set protocols igmp interface interface-name static group` *group* command. In the case when a real multicast receiver host is not available, as in this example, this static IGMP configuration is useful. On the receivers, to make them listen to the multicast group address, this example uses `set protocols sap listen` *group*.

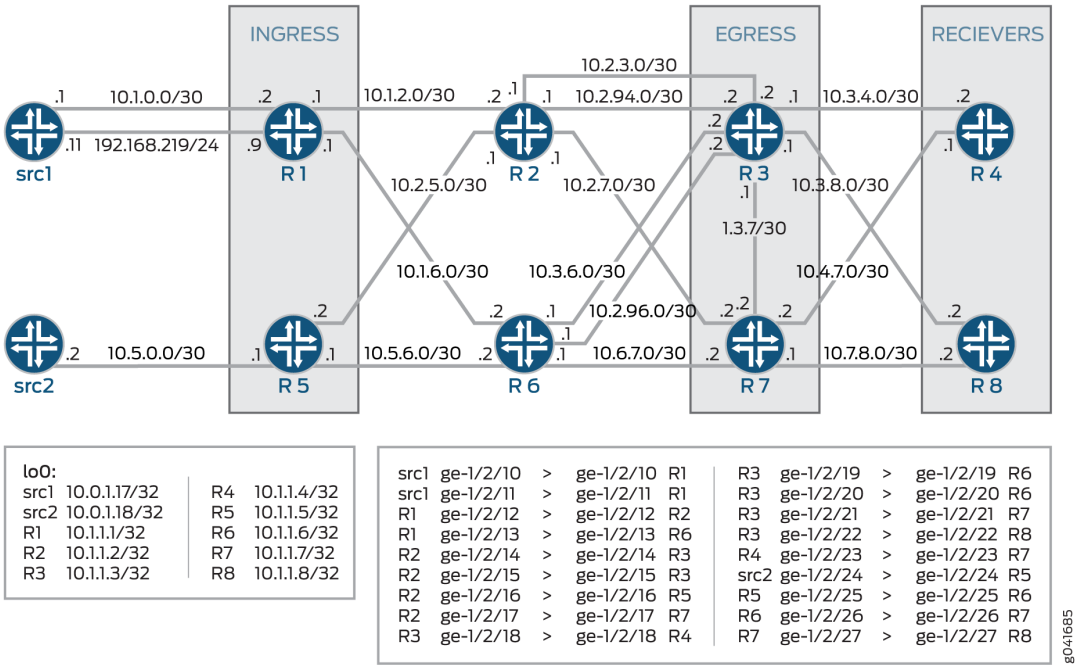
MoFRR configuration includes a policy option that is not shown in this example, but is explained separately. The option is configured as follows:

```
stream-protection {
    policy policy-name;
}
```

Topology

Figure 96 on page 1462 shows the sample network.

Figure 96: MoFRR in a Multipoint LDP Domain



"CLI Quick Configuration" on page 1462 shows the configuration for all of the devices in Figure 96 on page 1462.

The section "Configuration" on page 1471 describes the steps on Device R3.

CLI Quick Configuration

IN THIS SECTION

- CLI Quick Configuration | **1462**

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the `[edit]` hierarchy level.

Device src1

```
set interfaces ge-1/2/10 unit 0 description src1-to-R1
set interfaces ge-1/2/10 unit 0 family inet address 10.5.0.1/30
set interfaces ge-1/2/11 unit 0 description src1-to-R1
set interfaces ge-1/2/11 unit 0 family inet address 192.168.219.11/24
set interfaces lo0 unit 0 family inet address 10.0.1.17/32
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

Device src2

```
set interfaces ge-1/2/24 unit 0 description src2-to-R5
set interfaces ge-1/2/24 unit 0 family inet address 10.5.0.2/30
set interfaces lo0 unit 0 family inet address 10.0.1.18/32
set protocols rsvp interface all
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

Device R1

```
set interfaces ge-1/2/12 unit 0 description R1-to-R2
set interfaces ge-1/2/12 unit 0 family inet address 10.1.2.1/30
set interfaces ge-1/2/12 unit 0 family mpls
set interfaces ge-1/2/13 unit 0 description R1-to-R6
set interfaces ge-1/2/13 unit 0 family inet address 10.1.6.1/30
set interfaces ge-1/2/13 unit 0 family mpls
set interfaces ge-1/2/10 unit 0 description R1-to-src1
set interfaces ge-1/2/10 unit 0 family inet address 10.1.0.2/30
set interfaces ge-1/2/11 unit 0 description R1-to-src1
set interfaces ge-1/2/11 unit 0 family inet address 192.168.219.9/30
set interfaces lo0 unit 0 family inet address 10.1.1.1/32
set protocols rsvp interface all
set protocols mpls interface all
set protocols bgp group ibgp local-address 10.1.1.1
set protocols bgp group ibgp export static-route-tobgp
set protocols bgp group ibgp peer-as 65010
set protocols bgp group ibgp neighbor 10.1.1.3
set protocols bgp group ibgp neighbor 10.1.1.7
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
```

```
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ldp interface ge-1/2/12.0
set protocols ldp interface ge-1/2/13.0
set protocols ldp interface lo0.0
set protocols ldp p2mp
set protocols pim mldp-inband-signalling policy mldppim-ex
set protocols pim rp static address 10.1.1.5
set protocols pim interface lo0.0
set protocols pim interface ge-1/2/10.0
set protocols pim interface ge-1/2/11.0
set policy-options policy-statement mldppim-ex term B from source-address-filter 192.168.0.0/24
orlonger
set policy-options policy-statement mldppim-ex term B from source-address-filter
192.168.219.11/32 orlonger
set policy-options policy-statement mldppim-ex term B then p2mp-lsp-root address 10.1.1.2
set policy-options policy-statement mldppim-ex term B then accept
set policy-options policy-statement mldppim-ex term A from source-address-filter 10.1.1.7/32
orlonger
set policy-options policy-statement mldppim-ex term A from source-address-filter 10.1.0.0/30
orlonger
set policy-options policy-statement mldppim-ex term A then accept
set policy-options policy-statement static-route-tobgp term static from protocol static
set policy-options policy-statement static-route-tobgp term static from protocol direct
set policy-options policy-statement static-route-tobgp term static then accept
set routing-options autonomous-system 65010
```

Device R2

```
set interfaces ge-1/2/12 unit 0 description R2-to-R1
set interfaces ge-1/2/12 unit 0 family inet address 10.1.2.2/30
set interfaces ge-1/2/12 unit 0 family mpls
set interfaces ge-1/2/14 unit 0 description R2-to-R3
set interfaces ge-1/2/14 unit 0 family inet address 10.2.3.1/30
set interfaces ge-1/2/14 unit 0 family mpls
set interfaces ge-1/2/16 unit 0 description R2-to-R5
set interfaces ge-1/2/16 unit 0 family inet address 10.2.5.1/30
set interfaces ge-1/2/16 unit 0 family mpls
set interfaces ge-1/2/17 unit 0 description R2-to-R7
set interfaces ge-1/2/17 unit 0 family inet address 10.2.7.1/30
set interfaces ge-1/2/17 unit 0 family mpls
set interfaces ge-1/2/15 unit 0 description R2-to-R3
set interfaces ge-1/2/15 unit 0 family inet address 10.2.94.1/30
```

```
set interfaces ge-1/2/15 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.1.1.2/32
set interfaces lo0 unit 0 family mpls
set protocols rsvp interface all
set protocols mpls interface all
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ldp interface all
set protocols ldp p2mp
set policy-options policy-statement mldppim-ex term B from source-address-filter 192.168.0.0/24
orlonger
set policy-options policy-statement mldppim-ex term B from source-address-filter
192.168.219.11/32 orlonger
set policy-options policy-statement mldppim-ex term B then p2mp-lsp-root address 10.1.1.2
set policy-options policy-statement mldppim-ex term B then accept
set routing-options autonomous-system 65010
```

Device R3

```
set chassis network-services enhanced-ip
set interfaces ge-1/2/14 unit 0 description R3-to-R2
set interfaces ge-1/2/14 unit 0 family inet address 10.2.3.2/30
set interfaces ge-1/2/14 unit 0 family mpls
set interfaces ge-1/2/18 unit 0 description R3-to-R4
set interfaces ge-1/2/18 unit 0 family inet address 10.3.4.1/30
set interfaces ge-1/2/18 unit 0 family mpls
set interfaces ge-1/2/19 unit 0 description R3-to-R6
set interfaces ge-1/2/19 unit 0 family inet address 10.3.6.2/30
set interfaces ge-1/2/19 unit 0 family mpls
set interfaces ge-1/2/21 unit 0 description R3-to-R7
set interfaces ge-1/2/21 unit 0 family inet address 10.3.7.1/30
set interfaces ge-1/2/21 unit 0 family mpls
set interfaces ge-1/2/22 unit 0 description R3-to-R8
set interfaces ge-1/2/22 unit 0 family inet address 10.3.8.1/30
set interfaces ge-1/2/22 unit 0 family mpls
set interfaces ge-1/2/15 unit 0 description R3-to-R2
set interfaces ge-1/2/15 unit 0 family inet address 10.2.94.2/30
set interfaces ge-1/2/15 unit 0 family mpls
set interfaces ge-1/2/20 unit 0 description R3-to-R6
set interfaces ge-1/2/20 unit 0 family inet address 10.2.96.2/30
set interfaces ge-1/2/20 unit 0 family mpls
```

```
set interfaces lo0 unit 0 family inet address 10.1.1.3/32 primary
set routing-options autonomous-system 65010
set routing-options multicast stream-protection
set protocols rsvp interface all
set protocols mpls interface all
set protocols bgp group ibgp local-address 10.1.1.3
set protocols bgp group ibgp peer-as 10
set protocols bgp group ibgp neighbor 10.1.1.1
set protocols bgp group ibgp neighbor 10.1.1.5
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ldp interface all
set protocols ldp p2mp
set protocols pim mldp-inband-signalling policy mldppim-ex
set protocols pim interface lo0.0
set protocols pim interface ge-1/2/18.0
set protocols pim interface ge-1/2/22.0
set policy-options policy-statement mldppim-ex term B from source-address-filter 192.168.0.0/24
orlonger
set policy-options policy-statement mldppim-ex term B from source-address-filter
192.168.219.11/32 orlonger
set policy-options policy-statement mldppim-ex term B then accept
set policy-options policy-statement mldppim-ex term A from source-address-filter 10.1.0.1/30
orlonger
set policy-options policy-statement mldppim-ex term A then accept
set policy-options policy-statement static-route-tobgp term static from protocol static
set policy-options policy-statement static-route-tobgp term static from protocol direct
set policy-options policy-statement static-route-tobgp term static then accept
```

Device R4

```
set interfaces ge-1/2/18 unit 0 description R4-to-R3
set interfaces ge-1/2/18 unit 0 family inet address 10.3.4.2/30
set interfaces ge-1/2/18 unit 0 family mpls
set interfaces ge-1/2/23 unit 0 description R4-to-R7
set interfaces ge-1/2/23 unit 0 family inet address 10.4.7.1/30
set interfaces lo0 unit 0 family inet address 10.1.1.4/32
set protocols igmp interface ge-1/2/18.0 version 3
set protocols igmp interface ge-1/2/18.0 static group 232.1.1.1 group-count 2
set protocols igmp interface ge-1/2/18.0 static group 232.1.1.1 source 192.168.219.11
```

```
set protocols igmp interface ge-1/2/18.0 static group 232.2.2.2 source 10.2.7.7
set protocols sap listen 232.1.1.1
set protocols sap listen 232.2.2.2
set protocols rsvp interface all
set protocols mpls interface all
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols pim mldp-inband-signalling policy mldppim-ex
set protocols pim interface ge-1/2/23.0
set protocols pim interface ge-1/2/18.0
set protocols pim interface lo0.0
set policy-options policy-statement static-route-tobgp term static from protocol static
set policy-options policy-statement static-route-tobgp term static from protocol direct
set policy-options policy-statement static-route-tobgp term static then accept
set policy-options policy-statement mldppim-ex term B from source-address-filter 192.168.0.0/24
orlonger
set policy-options policy-statement mldppim-ex term B from source-address-filter
192.168.219.11/32 orlonger
set policy-options policy-statement mldppim-ex term B then p2mp-lsp-root address 10.1.1.2
set policy-options policy-statement mldppim-ex term B then accept
set routing-options autonomous-system 65010
```

Device R5

```
set interfaces ge-1/2/24 unit 0 description R5-to-src2
set interfaces ge-1/2/24 unit 0 family inet address 10.5.0.1/30
set interfaces ge-1/2/16 unit 0 description R5-to-R2
set interfaces ge-1/2/16 unit 0 family inet address 10.2.5.2/30
set interfaces ge-1/2/16 unit 0 family mpls
set interfaces ge-1/2/25 unit 0 description R5-to-R6
set interfaces ge-1/2/25 unit 0 family inet address 10.5.6.1/30
set interfaces ge-1/2/25 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.1.1.5/32
set protocols rsvp interface all
set protocols mpls interface all
set protocols bgp group ibgp local-address 10.1.1.5
set protocols bgp group ibgp export static-route-tobgp
set protocols bgp group ibgp peer-as 65010
set protocols bgp group ibgp neighbor 10.1.1.7
set protocols bgp group ibgp neighbor 10.1.1.3
set protocols ospf traffic-engineering
```

```
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ldp interface ge-1/2/16.0
set protocols ldp interface ge-1/2/25.0
set protocols ldp p2mp
set protocols pim interface lo0.0
set protocols pim interface ge-1/2/24.0
set policy-options policy-statement static-route-tobgp term static from protocol static
set policy-options policy-statement static-route-tobgp term static from protocol direct
set policy-options policy-statement static-route-tobgp term static then accept
set routing-options autonomous-system 65010
```

Device R6

```
set interfaces ge-1/2/13 unit 0 description R6-to-R1
set interfaces ge-1/2/13 unit 0 family inet address 10.1.6.2/30
set interfaces ge-1/2/13 unit 0 family mpls
set interfaces ge-1/2/19 unit 0 description R6-to-R3
set interfaces ge-1/2/19 unit 0 family inet address 10.3.6.1/30
set interfaces ge-1/2/19 unit 0 family mpls
set interfaces ge-1/2/25 unit 0 description R6-to-R5
set interfaces ge-1/2/25 unit 0 family inet address 10.5.6.2/30
set interfaces ge-1/2/25 unit 0 family mpls
set interfaces ge-1/2/26 unit 0 description R6-to-R7
set interfaces ge-1/2/26 unit 0 family inet address 10.6.7.1/30
set interfaces ge-1/2/26 unit 0 family mpls
set interfaces ge-1/2/20 unit 0 description R6-to-R3
set interfaces ge-1/2/20 unit 0 family inet address 10.2.96.1/30
set interfaces ge-1/2/20 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.1.1.6/30
set protocols rsvp interface all
set protocols mpls interface all
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ldp interface all
set protocols ldp p2mp
```

Device R7

```
set interfaces ge-1/2/17 unit 0 description R7-to-R2
set interfaces ge-1/2/17 unit 0 family inet address 10.2.7.2/30
set interfaces ge-1/2/17 unit 0 family mpls
set interfaces ge-1/2/21 unit 0 description R7-to-R3
set interfaces ge-1/2/21 unit 0 family inet address 10.3.7.2/30
set interfaces ge-1/2/21 unit 0 family mpls
set interfaces ge-1/2/23 unit 0 description R7-to-R4
set interfaces ge-1/2/23 unit 0 family inet address 10.4.7.2/30
set interfaces ge-1/2/23 unit 0 family mpls
set interfaces ge-1/2/26 unit 0 description R7-to-R6
set interfaces ge-1/2/26 unit 0 family inet address 10.6.7.2/30
set interfaces ge-1/2/26 unit 0 family mpls
set interfaces ge-1/2/27 unit 0 description R7-to-R8
set interfaces ge-1/2/27 unit 0 family inet address 10.7.8.1/30
set interfaces ge-1/2/27 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.1.1.7/32
set protocols rsvp interface all
set protocols mpls interface all
set protocols bgp group ibgp local-address 10.1.1.7
set protocols bgp group ibgp export static-route-tobgp
set protocols bgp group ibgp peer-as 65010
set protocols bgp group ibgp neighbor 10.1.1.5
set protocols bgp group ibgp neighbor 10.1.1.1
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ldp interface ge-1/2/17.0
set protocols ldp interface ge-1/2/21.0
set protocols ldp interface ge-1/2/26.0
set protocols ldp p2mp
set protocols pim mldp-inband-signalling policy mldppim-ex
set protocols pim interface lo0.0
set protocols pim interface ge-1/2/27.0
set policy-options policy-statement mldppim-ex term B from source-address-filter 192.168.0.0/24
orlonger
set policy-options policy-statement mldppim-ex term B from source-address-filter
192.168.219.11/32 orlonger
set policy-options policy-statement mldppim-ex term B then accept
set policy-options policy-statement mldppim-ex term A from source-address-filter 10.1.0.1/30
orlonger
```

```
set policy-options policy-statement mldppim-ex term A then accept
set policy-options policy-statement static-route-tobgp term static from protocol static
set policy-options policy-statement static-route-tobgp term static from protocol direct
set policy-options policy-statement static-route-tobgp term static then accept
set routing-options autonomous-system 65010
set routing-options multicast stream-protection policy mldppim-ex
```

Device R8

```
set interfaces ge-1/2/22 unit 0 description R8-to-R3
set interfaces ge-1/2/22 unit 0 family inet address 10.3.8.2/30
set interfaces ge-1/2/22 unit 0 family mpls
set interfaces ge-1/2/27 unit 0 description R8-to-R7
set interfaces ge-1/2/27 unit 0 family inet address 10.7.8.2/30
set interfaces ge-1/2/27 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.1.1.8/32
set protocols igmp interface ge-1/2/22.0 version 3
set protocols igmp interface ge-1/2/22.0 static group 232.1.1.1 group-count 2
set protocols igmp interface ge-1/2/22.0 static group 232.1.1.1 source 192.168.219.11
set protocols igmp interface ge-1/2/22.0 static group 232.2.2.2 source 10.2.7.7
set protocols sap listen 232.1.1.1
set protocols sap listen 232.2.2.2
set protocols rsvp interface all
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols pim mldp-inband-signalling policy mldppim-ex
set protocols pim interface ge-1/2/27.0
set protocols pim interface ge-1/2/22.0
set protocols pim interface lo0.0
set policy-options policy-statement static-route-tobgp term static from protocol static
set policy-options policy-statement static-route-tobgp term static from protocol direct
set policy-options policy-statement static-route-tobgp term static then accept
set policy-options policy-statement mldppim-ex term B from source-address-filter 192.168.0.0/24
orlonger
set policy-options policy-statement mldppim-ex term B from source-address-filter
192.168.219.11/32 orlonger
set policy-options policy-statement mldppim-ex term B then p2mp-lsp-root address 10.1.1.2
set policy-options policy-statement mldppim-ex term B then accept
set routing-options autonomous-system 65010
```

Configuration

IN THIS SECTION

- Procedure | 1471

Procedure

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the Junos OS CLI User Guide.

To configure Device R3:

1. Enable enhanced IP mode.

```
[edit chassis]
user@R3# set network-services enhanced-ip
```

2. Configure the device interfaces.

```
[edit interfaces]
user@R3# set ge-1/2/14 unit 0 description R3-to-R2
user@R3# set ge-1/2/14 unit 0 family inet address 10.2.3.2/30
user@R3# set ge-1/2/14 unit 0 family mpls
user@R3# set ge-1/2/18 unit 0 description R3-to-R4
user@R3# set ge-1/2/18 unit 0 family inet address 10.3.4.1/30
user@R3# set ge-1/2/18 unit 0 family mpls
user@R3# set ge-1/2/19 unit 0 description R3-to-R6
user@R3# set ge-1/2/19 unit 0 family inet address 10.3.6.2/30
user@R3# set ge-1/2/19 unit 0 family mpls
user@R3# set ge-1/2/21 unit 0 description R3-to-R7
user@R3# set ge-1/2/21 unit 0 family inet address 10.3.7.1/30
user@R3# set ge-1/2/21 unit 0 family mpls
user@R3# set ge-1/2/22 unit 0 description R3-to-R8
user@R3# set ge-1/2/22 unit 0 family inet address 10.3.8.1/30
```

```
user@R3# set ge-1/2/22 unit 0 family mpls
user@R3# set ge-1/2/15 unit 0 description R3-to-R2
user@R3# set ge-1/2/15 unit 0 family inet address 10.2.94.2/30
user@R3# set ge-1/2/15 unit 0 family mpls
user@R3# set ge-1/2/20 unit 0 description R3-to-R6
user@R3# set ge-1/2/20 unit 0 family inet address 10.2.96.2/30
user@R3# set ge-1/2/20 unit 0 family mpls
user@R3# set lo0 unit 0 family inet address 10.1.1.3/32 primary
```

3. Configure the autonomous system (AS) number.

```
user@R3# set routing-options autonomous-system 6510
```

4. Configure the routing policies.

```
[edit policy-options policy-statement mldppim-ex]
user@R3# set term B from source-address-filter 192.168.0.0/24 orlonger
user@R3# set term B from source-address-filter 192.168.219.11/32 orlonger
user@R3# set term B then accept
user@R3# set term A from source-address-filter 10.1.0.1/30 orlonger
user@R3# set term A then accept
[edit policy-options policy-statement static-route-tobgp]
user@R3# set  term static from protocol static
user@R3# set  term static from protocol direct
user@R3# set term static then accept
```

5. Configure PIM.

```
[edit protocols pim]
user@R3# set mldp-inband-signalling policy mldppim-ex
user@R3# set interface lo0.0
user@R3# set interface ge-1/2/18.0
user@R3# set interface ge-1/2/22.0
```

6. Configure LDP.

```
[edit protocols ldp]
user@R3# set interface all
user@R3# set p2mp
```

7. Configure an IGP or static routes.

```
[edit protocols ospf]
user@R3# set traffic-engineering
user@R3# set area 0.0.0.0 interface all
user@R3# set area 0.0.0.0 interface fxp0.0 disable
user@R3# set area 0.0.0.0 interface lo0.0 passive
```

8. Configure internal BGP.

```
[edit protocols bgp group ibgp]
user@R3# set local-address 10.1.1.3
user@R3# set peer-as 65010
user@R3# set neighbor 10.1.1.1
user@R3# set neighbor 10.1.1.5
```

9. Configure MPLS and, optionally, RSVP.

```
[edit protocols mpls]
user@R3# set interface all
[edit protocols rsvp]
user@R3# set interface all
```

10. Enable MoFRR.

```
[edit routing-options multicast]
user@R3# set stream-protection
```

Results

From configuration mode, confirm your configuration by entering the `show chassis`, `show interfaces`, `show protocols`, `show policy-options`, and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R3# show chassis
network-services enhanced-ip;
```

```
user@R3# show interfaces
ge-1/2/14 {
    unit 0 {
        description R3-to-R2;
        family inet {
            address 10.2.3.2/30;
        }
        family mpls;
    }
}
ge-1/2/18 {
    unit 0 {
        description R3-to-R4;
        family inet {
            address 10.3.4.1/30;
        }
        family mpls;
    }
}
ge-1/2/19 {
    unit 0 {
        description R3-to-R6;
        family inet {
            address 10.3.6.2/30;
        }
        family mpls;
    }
}
ge-1/2/21 {
    unit 0 {
        description R3-to-R7;
        family inet {
```

```
            address 10.3.7.1/30;
        }
        family mpls;
    }
}
ge-1/2/22 {
    unit 0 {
        description R3-to-R8;
        family inet {
            address 10.3.8.1/30;
        }
        family mpls;
    }
}
ge-1/2/15 {
    unit 0 {
        description R3-to-R2;
        family inet {
            address 10.2.94.2/30;
        }
        family mpls;
    }
}
ge-1/2/20 {
    unit 0 {
        description R3-to-R6;
        family inet {
            address 10.2.96.2/30;
        }
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 192.168.15.1/32;
            address 10.1.1.3/32 {
                primary;
            }
        }
```

```
    }
}
```

```
user@R3# show protocols
rsvp {
    interface all;
}
mpls {
    interface all;
}
bgp {
    group ibgp {
        local-address 10.1.1.3;
        peer-as 65010;
        neighbor 10.1.1.1;
        neighbor 10.1.1.5;
    }
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface all;
        interface fxp0.0 {
            disable;
        }
        interface lo0.0 {
            passive;
        }
    }
}
ldp {
    interface all;
    p2mp;
}
pim {
    mldp-inband-signalling {
        policy mldppim-ex;
    }
    interface lo0.0;
    interface ge-1/2/18.0;
```

```
    interface ge-1/2/22.0;
}
```

```
user@R3# show policy-options
policy-statement mldppim-ex {
    term B {
        from {
            source-address-filter 192.168.0.0/24 orlonger;
            source-address-filter 192.168.219.11/32 orlonger;
        }
        then accept;
    }
    term A {
        from {
            source-address-filter 10.1.0.1/30 orlonger;
        }
        then accept;
    }
}
policy-statement static-route-tobgp {
    term static {
        from protocol [ static direct ];
        then accept;
    }
}
```

```
user@R3# show routing-options
autonomous-system 65010;
multicast {
    stream-protection;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Checking the LDP Point-to-Multipoint Forwarding Equivalency Classes | **1478**
- Examining the Label Information | **1479**
- Checking the Multicast Routes | **1481**
- Checking the LDP Point-to-Multipoint Traffic Statistics | **1482**

Confirm that the configuration is working properly.

Checking the LDP Point-to-Multipoint Forwarding Equivalency Classes

Purpose

Make sure the MoFRR is enabled, and determine what labels are being used.

Action

```
user@R3> show ldp p2mp fec

LDP P2MP FECs:
 P2MP root-addr 10.1.1.1, grp: 232.1.1.1, src: 192.168.219.11
  MoFRR enabled
  Fec type: Egress (Active)
  Label: 301568
 P2MP root-addr 10.1.1.1, grp: 232.1.1.2, src: 192.168.219.11
  MoFRR enabled
  Fec type: Egress (Active)
  Label: 301600
```

Meaning

The output shows that MoFRR is enabled, and it shows that the labels 301568 and 301600 are being used for the two multipoint LDP point-to-multipoint LSPs.

Examining the Label Information

Purpose

Make sure that the egress device has two upstream interfaces for the multicast group join.

Action

```
user@R3> show route label 301568 detail

mpls.0: 18 destinations, 18 routes (18 active, 0 holddown, 0 hidden)
301568 (1 entry, 1 announced)
        *LDP    Preference: 9
                Next hop type: Flood
                Address: 0x2735208
                Next-hop reference count: 3
                Next hop type: Router, Next hop index: 1397
                Address: 0x2735d2c
                Next-hop reference count: 3
                Next hop: 10.3.8.2 via ge-1/2/22.0
                Label operation: Pop
                Load balance label:  None;
                Next hop type: Router, Next hop index: 1395
                Address: 0x2736290
                Next-hop reference count: 3
                Next hop: 10.3.4.2 via ge-1/2/18.0
                Label operation: Pop
                Load balance label:  None;
                State: <Active Int AckRequest MulticastRPF>
                Local AS:    65010
                Age: 54:05      Metric: 1
                Validation State: unverified
                Task: LDP
                Announcement bits (1): 0-KRT
                AS path: I
                FECs bound to route: P2MP root-addr 10.1.1.1, grp: 232.1.1.1, src: 192.168.219.11
                Primary Upstream : 10.1.1.3:0--10.1.1.2:0
                   RPF Nexthops  :
                        ge-1/2/15.0, 10.2.94.1, Label: 301568, weight: 0x1
                        ge-1/2/14.0, 10.2.3.1, Label: 301568, weight: 0x1
                Backup Upstream : 10.1.1.3:0--10.1.1.6:0
                   RPF Nexthops :
```

```
                        ge-1/2/20.0, 10.2.96.1, Label: 301584, weight: 0xfffe
                        ge-1/2/19.0, 10.3.6.1, Label: 301584, weight: 0xfffe
```

```
user@R3> show route label 301600 detail

mpls.0: 18 destinations, 18 routes (18 active, 0 holddown, 0 hidden)
301600 (1 entry, 1 announced)
        *LDP    Preference: 9
                Next hop type: Flood
                Address: 0x27356b4
                Next-hop reference count: 3
                Next hop type: Router, Next hop index: 1520
                Address: 0x27350f4
                Next-hop reference count: 3
                Next hop: 10.3.8.2 via ge-1/2/22.0
                Label operation: Pop
                Load balance label:  None;
                Next hop type: Router, Next hop index: 1481
                Address: 0x273645c
                Next-hop reference count: 3
                Next hop: 10.3.4.2 via ge-1/2/18.0
                Label operation: Pop
                Load balance label:  None;
                State: <Active Int AckRequest MulticastRPF>
                Local AS:    65010
                Age: 54:25      Metric: 1
                Validation State: unverified
                Task: LDP
                Announcement bits (1): 0-KRT
                AS path: I
                FECs bound to route: P2MP root-addr 10.1.1.1, grp: 232.1.1.2, src: 192.168.219.11
                Primary Upstream : 10.1.1.3:0--10.1.1.6:0
                   RPF Nexthops  :
                        ge-1/2/20.0, 10.2.96.1, Label: 301600, weight: 0x1
                        ge-1/2/19.0, 10.3.6.1, Label: 301600, weight: 0x1
                Backup Upstream : 10.1.1.3:0--1.1.1.2:0
                   RPF Nexthops :
                        ge-1/2/15.0, 10.2.94.1, Label: 301616, weight: 0xfffe
                        ge-1/2/14.0, 10.2.3.1, Label: 301616, weight: 0xfffe
```

Meaning

The output shows the primary upstream paths and the backup upstream paths. It also shows the RPF next hops.

Checking the Multicast Routes

Purpose

Examine the IP multicast forwarding table to make sure that there is an upstream RPF interface list, with a primary and a backup interface.

Action

```
user@R3> show ldp p2mp path
P2MP path type: Transit/Egress
  Output Session (label): 10.1.1.2:0 (301568) (Primary)
  Egress Nexthops: Interface ge-1/2/18.0
                   Interface ge-1/2/22.0
  RPF Nexthops:    Interface ge-1/2/15.0, 10.2.94.1, 301568, 1
                   Interface ge-1/2/20.0, 10.2.96.1, 301584, 65534
                   Interface ge-1/2/14.0, 10.2.3.1, 301568, 1
                   Interface ge-1/2/19.0, 10.3.6.1, 301584, 65534
  Attached FECs:  P2MP root-addr 10.1.1.1, grp: 232.1.1.1, src: 192.168.219.11 (Active)
P2MP path type: Transit/Egress
  Output Session (label): 10.1.1.6:0 (301584) (Backup)
  Egress Nexthops: Interface ge-1/2/18.0
                   Interface ge-1/2/22.0
  RPF Nexthops:    Interface ge-1/2/15.0, 10.2.94.1, 301568, 1
                   Interface ge-1/2/20.0, 10.2.96.1, 301584, 65534
                   Interface ge-1/2/14.0, 10.2.3.1, 301568, 1
                   Interface ge-1/2/19.0, 10.3.6.1, 301584, 65534
  Attached FECs:  P2MP root-addr 10.1.1.1, grp: 232.1.1.1, src: 192.168.219.11 (Active)
P2MP path type: Transit/Egress
  Output Session (label): 10.1.1.6:0 (301600) (Primary)
  Egress Nexthops: Interface ge-1/2/18.0
                   Interface ge-1/2/22.0
  RPF Nexthops:    Interface ge-1/2/15.0, 10.2.94.1, 301616, 65534
                   Interface ge-1/2/20.0, 10.2.96.1, 301600, 1
                   Interface ge-1/2/14.0, 10.2.3.1, 301616, 65534
                   Interface ge-1/2/19.0, 10.3.6.1, 301600, 1
```

```
  Attached FECs:  P2MP root-addr 10.1.1.1, grp: 232.1.1.2, src: 192.168.219.11 (Active)
P2MP path type: Transit/Egress
  Output Session (label): 10.1.1.2:0 (301616) (Backup)
  Egress Nexthops: Interface ge-1/2/18.0
                   Interface ge-1/2/22.0
  RPF Nexthops:    Interface ge-1/2/15.0, 10.2.94.1, 301616, 65534
                   Interface ge-1/2/20.0, 10.2.96.1, 301600, 1
                   Interface ge-1/2/14.0, 10.2.3.1, 301616, 65534
                   Interface ge-1/2/19.0, 10.3.6.1, 301600, 1
  Attached FECs:  P2MP root-addr 10.1.1.1, grp: 232.1.1.2, src: 192.168.219.11 (Active)
```

Meaning

The output shows primary and backup sessions, and RPF next hops.

Checking the LDP Point-to-Multipoint Traffic Statistics

Purpose

Make sure that both primary and backup statistics are listed.

Action

```
user@R3> show ldp traffic-statistics p2mp

P2MP FEC Statistics:

FEC(root_addr:lsp_id/grp,src)     Nexthop              Packets             Bytes Shared
10.1.1.1:232.1.1.1,192.168.219.11, Label: 301568
                                  10.3.8.2                    0                0 No
                                  10.3.4.2                    0                0 No
10.1.1.1:232.1.1.1,192.168.219.11, Label: 301584, Backup route
                                  10.3.4.2                    0                0 No
                                  10.3.8.2                    0                0 No
10.1.1.1:232.1.1.2,192.168.219.11, Label: 301600
                                  10.3.8.2                    0                0 No
                                  10.3.4.2                    0                0 No
10.1.1.1:232.1.1.2,192.168.219.11, Label: 301616, Backup route
                                  10.3.4.2                    0                0 No
                                  10.3.8.2                    0                0 No
```

Meaning

The output shows both primary and backup routes with the labels.

Example: Configuring LDP Downstream on Demand

IN THIS SECTION

- Requirements | **1483**
- Overview | **1483**
- Configuration | **1484**
- Verification | **1488**

This example shows how to configure *LDP* downstream on demand. LDP is commonly configured using downstream unsolicited advertisement mode, meaning label advertisements for all routes are received from all LDP peers. As service providers integrate the access and aggregation networks into a single MPLS domain, LDP downstream on demand is needed to distribute the bindings between the access and aggregation networks and to reduce the processing requirements for the control plane.

Downstream nodes could potentially receive tens of thousands of label bindings from upstream aggregation nodes. Instead of learning and storing all label bindings for all possible loopback addresses within the entire MPLS network, the downstream aggregation node can be configured using LDP downstream on demand to only request the label bindings for the FECs corresponding to the loopback addresses of those egress nodes on which it has services configured.

Requirements

This example uses the following hardware and software components:

- M Series router
- Junos OS 12.2

Overview

You can enable LDP downstream on demand label advertisement for an LDP session by including the *downstream-on-demand* statement at the `[edit protocols ldp session]` hierarchy level. If you have configured downstream on demand, the Juniper Networks router advertises the downstream on demand request to its peer routers. For a downstream on demand session to be established between two routers, both have to advertise downstream on demand mode during LDP session establishment. If

one router advertises downstream unsolicited mode and the other advertises downstream on demand, downstream unsolicited mode is used.

Configuration

IN THIS SECTION

- Configuring LDP Downstream on Demand | **1484**
- Distributing LDP Downstream on Demand Routes into Labeled BGP | **1485**

Configuring LDP Downstream on Demand

Step-by-Step Procedure

To configure a LDP downstream on demand policy and then configure that policy and enable LDP downstream on demand on the LDP session:

1. Configure the downstream on demand policy (*DOD-Request-Loopbacks* in this example).

   This policy causes the router to forward label request messages only to the FECs that are matched by the *DOD-Request-Loopbacks* policy.

   ```
   [edit policy-options]
   user@host# set prefix-list Request-Loopbacks 10.1.1.1/32
   user@host# set prefix-list Request-Loopbacks 10.1.1.2/32
   user@host# set prefix-list Request-Loopbacks 10.1.1.3/32
   user@host# set prefix-list Request-Loopbacks 10.1.1.4/32
   user@host# set policy-statement DOD-Request-Loopbacks term 1 from prefix-list Request-
   Loopbacks
   user@host# set policy-statement DOD-Request-Loopbacks term 1 then accept
   ```

2. Specify the DOD-Request-Loopbacks policy using the `dod-request-policy` statement at the `[edit protocols ldp]` hierarchy level.

   The policy specified with the `dod-request-policy` statement is used to identify the prefixes to send label request messages. This policy is similar to an egress policy or an import policy. When processing routes from the inet.0 routing table, the Junos OS software checks for routes matching the `DOD-Request-Loopbacks` policy (in this example). If the route matches the policy and the LDP session is

negotiated with DOD advertisement mode, label request messages are sent to the corresponding downstream LDP session.

```
[edit protocols ldp]
user@host# set dod-request-policy DOD-Request-Loopbacks
```

3. Include the `downstream-on-demand` statement in the configuration for the LDP session to enable downstream on demand distribution mode.

```
[edit protocols ldp]
user@host# set session 172.16.1.1 downstream-on-demand
```

Distributing LDP Downstream on Demand Routes into Labeled BGP

Step-by-Step Procedure

To distribute LDP downstream on demand routes into labeled BGP, use a BGP export policy.

1. Configure the LDP route policy (`redistribute_ldp` in this example).

```
[edit policy-options]
user@host# set policy-statement redistribute_ldp term 1 from protocol ldp
user@host# set policy-statement redistribute_ldp term 1 from tag 1000
user@host# set policy-statement redistribute_ldp term 1 then accept
```

2. Include the LDP route policy, `redistribute_ldp` in the BGP configuration (as a part of the BGP group configuration `ebgp-to-abr` in this example).

   BGP forwards the LDP routes based on the `redistribute_ldp` policy to the remote PE router

```
[edit protocols bgp]
user@host# set group ebgp-to-abr type external
user@host# set group ebgp-to-abr local-address 192.168.0.1
user@host# set group ebgp-to-abr peer-as 65319
user@host# set group ebgp-to-abr local-as 65320
user@host# set group ebgp-to-abr neighbor 192.168.6.1 family inet unicast
user@host# set group ebgp-to-abr neighbor 192.168.6.1 family inet labeled-unicast rib inet.3
user@host# set group ebgp-to-abr neighbor 192.168.6.1 export redistribute_ldp
```

Step-by-Step Procedure

To restrict label propagation to other routers configured in downstream unsolicited mode (instead of downstream on demand), configure the following policies:

1. Configure the `dod-routes` policy to accept routes from LDP.

```
user@host# set policy-options policy-statement dod-routes term 1 from protocol ldp
user@host# set policy-options policy-statement dod-routes term 1 from tag 1145307136
user@host# set policy-options policy-statement dod-routes term 1 then accept
```

2. Configure the `do-not-propagate-du-sessions` policy to not forward routes to neighbors `10.1.1.1`, `10.2.2.2`, and `10.3.3.3`.

```
user@host# set policy-options policy-statement do-not-propagate-du-sessions term 1 to
neighbor 10.1.1.1
user@host# set policy-options policy-statement do-not-propagate-du-sessions term 1 to
neighbor 10.2.2.2
user@host# set policy-options policy-statement do-not-propagate-du-sessions term 1 to
neighbor 10.3.3.3
user@host# set policy-options policy-statement do-not-propagate-du-sessions term 1 then reject
```

3. Configure the `filter-dod-on-du-sessions` policy to prevent the routes examined by the `dod-routes` policy from being forwarded to the neighboring routers defined in the `do-not-propagate-du-sessions` policy.

```
user@host# set policy-options policy-statement filter-dod-routes-on-du-sessions term 1 from
policy dod-routes
user@host# set policy-options policy-statement filter-dod-routes-on-du-sessions term 1 to
policy do-not-propagate-du-sessions
```

4. Specify the `filter-dod-routes-on-du-sesssion` policy as the export policy for BGP group `ebgp-to-abr`.

```
[edit protocols bgp]
user@host# set group ebgp-to-abr neighbor 192.168.6.2 export filter-dod-routes-on-du-sessions
```

Results

From configuration mode, confirm your configuration by entering the `show policy-options` and `show protocols ldp` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host#
show policy-options
prefix-list Request-Loopbacks {
    10.1.1.1/32;
    10.1.1.2/32;
    10.1.1.3/32;
    10.1.1.4/32;
}
policy-statement DOD-Request-Loopbacks {
    term 1 {
        from {
            prefix-list Request-Loopbacks;
        }
        then accept;
    }
}
policy-statement redistribute_ldp {
    term 1 {
        from {
            protocol ldp;
            tag 1000;
        }
        then accept;
    }
}
```

```
user@host#
show protocols ldp
dod-request-policy DOD-Request-Loopbacks;
session 172.16.1.1 {
```

```
    downstream-on-demand;
}
```

```
user@host#
show protocols bgp
group ebgp-to-abr {
    type external;
    local-address 192.168.0.1;
    peer-as 65319;
    local-as 65320;
    neighbor 192.168.6.1 {
        family inet {
            unicast;
            labeled-unicast {
                rib {
                    inet.3;
                }
            }
        }
        export redistribute_ldp;
    }
}
```

Verification

IN THIS SECTION

- Verifying Label Advertisement Mode | **1488**

Verifying Label Advertisement Mode

Purpose

Confirm that the configuration is working properly.

Use the `show ldp session` command to verify the status of the label advertisement mode for the LDP session.

Action

Issue the `show ldp session` and `show ldp session detail` commands:

- The following command output for the `show ldp session` command indicates that the `Adv. Mode` (label advertisement mode) is `DOD` (meaning the LDP downstream on demand session is operational):

```
user@host> show ldp session
  Address          State        Connection     Hold time  Adv. Mode
172.16.1.2            Operational  Open             22          DOD
```

- The following command output for the `show ldp session detail` command indicates that the `Local Label Advertisement mode` is `Downstream unsolicited`, the default value (meaning downstream on demand is not configured on the local session). Conversely, the `Remote Label Advertisement mode` and the `Negotiated Label Advertisement mode` both indicate that `Downstream on demand` is configured on the remote session

```
user@host> show ldp session detail
Address: 172.16.1.2, State: Operational, Connection: Open, Hold time: 24
  Session ID: 10.1.1.1:0--10.1.1.2:0
  Next keepalive in 4 seconds
  Passive, Maximum PDU: 4096, Hold time: 30, Neighbor count: 1
  Neighbor types: configured-tunneled
  Keepalive interval: 10, Connect retry interval: 1
  Local address: 10.1.1.1, Remote address: 10.1.1.2
  Up for 17:54:52
  Capabilities advertised: none
  Capabilities received: none
  Protection: disabled
  Local - Restart: disabled, Helper mode: enabled,
  Remote - Restart: disabled, Helper mode: enabled
  Local maximum neighbor reconnect time: 120000 msec
  Local maximum neighbor recovery time: 240000 msec
  Local Label Advertisement mode: Downstream unsolicited
  Remote Label Advertisement mode: Downstream on demand
  Negotiated Label Advertisement mode: Downstream on demand
  Nonstop routing state: Not in sync
```

```
  Next-hop addresses received:
    10.1.1.2
```

Configuring LDP Native IPv6 Support

LDP is supported in an IPv6-only network, and in an IPv6 or IPv4 dual-stack network as described in *RFC 7552*. Configure the address family as `inet` for IPv4 or `inet6` for IPv6 or both, and the transport preference to be either `IPv4` or `IPv6`. The `dual-transport` statement allows Junos OS LDP to establish the TCP connection over IPv4 with IPv4 neighbors, and over IPv6 with IPv6 neighbors as a single-stack LSR. The `inet-lsr-id` and `inet6-lsr-id` IDs are the two LSR IDs that have to be configured to establish an LDP session over IPv4 and IPv6 TCP transport. These two IDs should be non-zero and must be configured with different values.

Before you configure IPv6 as dual-stack, be sure you configure the routing and signaling protocols.

To configure LDP native IPv6 support, you must do the following:

1. Enable forwarding equivalence class (FEC) deaggregation in order to use different labels for different address families.

```
[edit protocols ldp]
set deaggregate
```

2. Configure LDP address families.

```
[edit protocols ldp]
set family inet6
set family inet
```

3. Configure the `transport-preference` statement to select the preferred transport for the TCP connection when both IPv4 and IPv6 are enabled. By default, IPv6 is used as the TCP transport for establishing an LDP connection.

```
[edit protocols ldp]
set transport-preference ipv4
```

4. (Optional) Configure dual-transport to allow LDP to establish a separate IPv4 session with an IPv4 neighbor, and an IPv6 session with an IPv6 neighbor. Configure `inet-lsr-id` as the LSR ID for IPv4, and `inet6-lsr-id` as the LSR ID for IPv6.

```
[edit protocols ldp dual-transport]
set inet-lsr-id inet-lsr-id
set inet6-lsr-id inet6-lsr-id
```

For example, configure inet-lsr-id as 10.255.0.1, and inet6-lsr-id as 10.1.1.1.

```
[edit protocols ldp dual-transport]
set inet-lsr-id 10.255.0.1
set inet6-lsr-id 10.1.1.1
```

Example: Configuring LDP Native IPv6 Support

IN THIS SECTION

- Requirements | 1491
- Overview | 1492
- Configuration | 1492

This example shows how to allow the Junos OS Label Distribution Protocol (LDP) to establish the TCP connection over IPv4 with IPv4 neighbors, and over IPv6 with IPv6 neighbors as a single-stack LSR. This helps avoid tunneling of IPv6 over IPv4 MPLS core with IPv4-signaled MPLS label-switched paths (LSPs).

Requirements

This example uses the following hardware and software components:

- Two MX Series routers
- Junos OS Release 16.1 or later running on all devices

Before you configure IPv6 as dual-stack, be sure you configure the routing and signaling protocols.

Overview

IN THIS SECTION

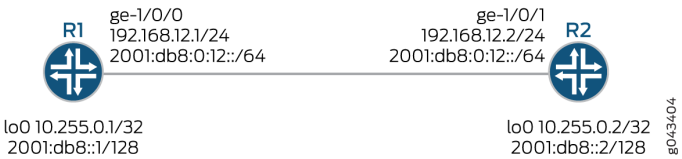
- Topology | 1492

LDP is supported in an IPv6 only network, and in an IPv6 or IPv4 dual-stack network as described in *RFC 7552*. Configure the address family as `inet` for IPv4 or `inet6` for IPv6. By default, IPv6 is used as the TCP transport for the LDP session with its peers when both IPv4 and IPv6 are enabled . The dual-transport statement allows Junos LDP to establish the TCP connection over IPv4 with IPv4 neighbors, and over IPv6 with IPv6 neighbors as a single-stack LSR. The `inet-lsr-id` and `inet6-lsr-id` are the two LSR IDs that have to be configured to establish an LDP session over IPv4 and IPv6 TCP transport. These two IDs should be non-zero and must be configured with different values.

Topology

Figure 97 on page 1492 shows the LDP IPv6 configured as dual-stack on Device R1 and Device R2.

Figure 97: Example LDP Native IPv6 Support



Configuration

IN THIS SECTION

- Verification | 1499
- Verification | 1505
- Verification | 1507

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter `commit` from configuration mode.

R1

```
set interfaces ge-1/0/0 unit 0 family inet address 192.168.12.1/24
set interfaces ge-1/0/0 unit 0 family iso
set interfaces ge-1/0/0 unit 0 family inet6 address 2001:db8:0:12::/64 eui-64
set interfaces ge-1/0/0 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.0.1/32
set interfaces lo0 unit 0 family iso address 49.0001.1720.1600.1010.00
set interfaces lo0 unit 0 family inet6 address 2001:db8::1/128
set protocols isis interface ge-1/0/0.0
set protocols isis interface lo0.0
set protocols mpls interface ge-1/0/0.0
set protocols ldp deaggregate
set protocols ldp interface ge-1/0/0.0
set protocols ldp interface lo0.0
set protocols ldp family inet6
set protocols ldp family inet
```

R2

```
set interfaces ge-1/0/1 unit 0 family inet address 192.168.12.2/24
set interfaces ge-1/0/1 unit 0 family iso
set interfaces ge-1/0/1 unit 0 family inet6 address 2001:db8:0:12::/64 eui-64
set interfaces ge-1/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.0.2/32
set interfaces lo0 unit 0 family iso address 49.0001.1720.1600.2020.00
set interfaces lo0 unit 0 family inet6 address 2001:db8::2/128
set protocols isis interface ge-1/0/1.0
set protocols isis interface lo0.0
set protocols mpls interface ge-1/0/1.0
set protocols ldp deaggregate
set protocols ldp interface ge-1/0/1.0
set protocols ldp interface lo0.0
set protocols ldp family inet6
set protocols ldp family inet
```

Configuring R1

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see " *Using the CLI Editor in Configuration Mode* " in the Junos OS CLI User Guide.

To configure Device R1:

1. Configure the interfaces.

```
[edit interfaces]
set ge-1/0/0 unit 0 family inet address 192.168.12.1/24
set ge-1/0/0 unit 0 family iso
set ge-1/0/0 unit 0 family inet6 address 2001:db8:0:12::/64 eui-64
set ge-1/0/0 unit 0 family mpls
```

2. Assign a loopback address to the device.

```
[edit interfaces lo0 unit 0]
set family inet address 10.255.0.1/32
set family iso address 49.0001.1720.1600.1010.00
set family inet6 address 2001:db8::1/128
```

3. Configure the IS-IS interfaces.

```
[edit protocols isis]
set interface ge-1/0/0.0
set interface lo0.0
```

4. Configure MPLS to use LDP interfaces on the device.

```
[edit protocols mpls]
set protocols mpls interface ge-1/0/0.0
set interface ge-1/0/0.0
set interface lo0.0
```

5. Enable forwarding equivalence class (FEC) deaggregation in order to use different labels for different address families.

```
[edit protocols ldp]
set deaggregate
```

6. Configure LDP address families.

```
[edit protocols ldp]
set family inet6
set family inet
```

Results

From configuration mode, confirm your configuration by entering the **show interfaces** and **show protocols** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R1# show interfaces
    ge-1/0/0 {
        unit 0 {
            family inet {
                address 192.168.12.1/24;
            }
            family iso;
            family inet6 {
                address 2001:db8:0:12::/64 {
                    eui-64;
                }
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.0.1/32;
```

```
            }
            family iso {
                address 49.0001.1720.1600.1010.00
            }
            family inet6 {
                address 2001:db8::1/128;
            }
        }
    }
```

```
user@R1# show protocols
    mpls {
        interface ge-1/0/0.0;
    }
    isis {
        interface ge-1/0/0.0;
        interface lo0.0;
    }
    ldp {
        deaggregate;
        interface ge-1/0/0.0;
        interface lo0.0;
        family {
            inet6;
            inet;
        }
    }
```

Configure transport-preference to Select the Preferred Transport

CLI Quick Configuration

Step-by-Step Procedure

You can configure the `transport-preference` statement to select the preferred transport for a TCP connection when both IPv4 and IPv6 are enabled. By default, IPv6 is used as TCP transport for establishing an LDP connection.

- (Optional) Configure the transport preference for an LDP connection.

```
[edit protocols ldp]
set transport-preference ipv4
```

Step-by-Step Procedure

Results

From configuration mode, confirm your configuration by entering the **show protocols** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R1# show protocols
    mpls {
        interface ge-1/0/0.0;
    }
    isis {
        interface ge-1/0/0.0;
        interface lo0.0;
    }
    ldp {
        deaggregate;
        interface ge-1/0/0.0;
        interface lo0.0;
        family {
            inet6;
            inet;
        }
        transport-preference ipv4;
    }
```

Configure dual-transport to Establish Separate Sessions for IPv4 with an IPv4 Neighbor and IPv6 with an IPv6 Neighbor

Step-by-Step Procedure

You can configure the `dual-transport` statement to allow LDP to establish a separate IPv4 session with an IPv4 neighbor, and an IPv6 session with an IPv6 neighbor. This requires the configuration of `inet-lsr-id` as the LSR ID for IPv4, and `inet6-lsr-id` as the LSR ID for IPv6.

- (Optional) Configure dual-transport to allow LDP to establish the TCP connection over IPv4 with IPv4 neighbors, and over IPv6 with IPv6 neighbors as a single-stack LSR.

```
[edit protocols ldp dual-transport]
set inet-lsr-id 10.255.0.1
set inet6-lsr-id 10.1.1.1
```

Results

From configuration mode, confirm your configuration by entering the **show protocols** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R1# show protocols
    mpls {
        interface ge-1/0/0.0;
    }
    isis {
        interface ge-1/0/0.0;
        interface lo0.0;
    }
    ldp {
        deaggregate;
        interface ge-1/0/0.0;
        interface lo0.0;
        family {
            inet6;
            inet;
        }
```

```
        dual-transport {
            inet-lsr-id 10.255.0.1;
            inet6-lsr-id 10.1.1.1;
        }
    }
```

Verification

IN THIS SECTION

- Verifying the Route Entries in the mpls.0 Table | **1499**
- Verifying the Route Entries in the inet.3 Table | **1500**
- Verifying the Route Entries in the inet6.3 Table | **1501**
- Verifying the LDP Database | **1501**
- Verifying the LDP Neighbor Information | **1502**
- Verifying the LDP Session Information | **1503**

Confirm that the configuration is working properly.

Verifying the Route Entries in the mpls.0 Table

Purpose

Display mpls.0 route table information.

Action

On Device R1, from operational mode, run the `show route table mpls.0` command to display mpls.0 route table information.

```
user@R1> show route table mpls.0
mpls.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 05:19:58, metric 1
                      Receive
1                  *[MPLS/0] 05:19:58, metric 1
                      Receive
```

```
2                  *[MPLS/0] 05:19:58, metric 1
                      Receive
13                 *[MPLS/0] 05:19:58, metric 1
                      Receive
299824             *[LDP/9] 04:28:45, metric 1
                    > to fe80::21f:1200:cb6:4c8d via ge-1/0/0.0, Pop
299824(S=0)        *[LDP/9] 04:28:45, metric 1
                    > to fe80::21f:1200:cb6:4c8d via ge-1/0/0.0, Pop
299888             *[LDP/9] 00:56:12, metric 1
                    > to 192.168.12.2 via ge-1/0/0.0, Pop
299888(S=0)        *[LDP/9] 00:56:12, metric 1
                    > to 192.168.12.2 via ge-1/0/0.0, Pop
```

Meaning

The output shows the mpls.0 route table information.

Verifying the Route Entries in the inet.3 Table

Purpose

Display inet.3 route table information.

Action

On Device R1, from operational mode, run the `show route table inet.3` command to display inet.3 route table information.

```
user@R1> show route table inet.3
inet.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.255.0.2/32      *[LDP/9] 00:58:38, metric 1
                    > to 192.168.12.2 via ge-1/0/0.0
```

Meaning

The output shows the inet.3 route table information.

Verifying the Route Entries in the inet6.3 Table

Purpose

Display inet6.3 route table information.

Action

On Device R1, from operational mode, run the `show route table inet6.3` command to display inet6.3 route table information.

```
user@R1> show route table inet6.3

inet6.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2001:db8::2/128    *[LDP/9] 04:31:17, metric 1
                    > to fe80::21f:1200:cb6:4c8d via ge-1/0/0.0
```

Meaning

The output shows the inet6.3 route table information.

Verifying the LDP Database

Purpose

Display the LDP database information.

Action

On Device R1, from operational mode, run the `show ldp database` command to display LDP database information.

```
user@R1> show ldp database

Input label database, 10.255.0.1:0--10.255.0.2:0
Labels received: 3
  Label     Prefix
 299840      10.255.0.1/32
```

```
      3      10.255.0.2/32
 299808      2001:db8::1/128
      3      2001:db8::2/128

Output label database, 10.255.0.1:0--10.255.0.2:0
Labels advertised: 3
  Label     Prefix
      3      10.255.0.1/32
 299888      10.255.0.2/32
      3      2001:db8::1/128
 299824      2001:db8::2/128
```

Meaning

The output shows the entries in the LDP database.

Verifying the LDP Neighbor Information

Purpose

Display the LDP neighbor information.

Action

On Device R1, from operational mode, run the `show ldp neighbor` and `show ldp neighbor extensive` commands to display LDP neighbor information.

```
user@R1> show ldp neighbor
Address                          Interface       Label space ID     Hold time
fe80::21f:1200:cb6:4c8d          ge-1/0/0.0      10.255.0.2:0         12
192.168.12.2                     ge-1/0/0.0      10.255.0.2:0         11


user@R1> show ldp neighbor extensive

Address                          Interface       Label space ID     Hold time
192.168.12.2                     ge-1/0/0.0      10.255.0.2:0         11
  Transport address: 10.255.0.2, Transport preference: IPv6, Configuration sequence: 10
  Up for 00:04:35
  Reference count: 1
  Hold time: 15, Proposed local/peer: 15/15
```

```
  Hello flags: none
  Neighbor types: discovered
Address                          Interface       Label space ID     Hold time
fe80::21f:1200:cb6:4c8d          ge-1/0/0.0      10.255.0.2:0         14
  Transport address: 2001:db8::2, Transport preference: IPv6, Configuration sequence: 10
  Up for 00:04:35
  Reference count: 1
  Hold time: 15, Proposed local/peer: 15/15
  Hello flags: none
  Neighbor types: discovered
```

Meaning

The output shows LDP neighbor information of both IPv4 and IPv6 addresses.

Verifying the LDP Session Information

Purpose

Display the LDP session information.

Action

On Device R1, from operational mode, run the `show ldp session` and `show ldp session extensive` commands to display LDP session information.

```
user@R1> show ldp session
session
  Address                        State       Connection  Hold time  Adv. Mode
2001:db8::2                      Operational Open          20         DU


user@R1> show ldp session extensive

Address: 2001:db8::2, State: Operational, Connection: Open, Hold time: 29
  Session ID: 10.255.0.1:0--10.255.0.2:0
  Next keepalive in 9 seconds
  Passive, Maximum PDU: 4096, Hold time: 30, Neighbor count: 1
  Neighbor types: discovered
  Keepalive interval: 10, Connect retry interval: 1
```

```
  Local address: 2001:db8::1, Remote address: 2001:db8::2
  Up for 00:05:31
  Capabilities advertised: none
  Capabilities received: none
  Protection: disabled
  Session flags: none
  Local - Restart: disabled, Helper mode: enabled
  Remote - Restart: disabled, Helper mode: enabled
  Local maximum neighbor reconnect time: 120000 msec
  Local maximum neighbor recovery time: 240000 msec
  Local Label Advertisement mode: Downstream unsolicited
  Remote Label Advertisement mode: Downstream unsolicited
  Negotiated Label Advertisement mode: Downstream unsolicited
  MTU discovery: disabled
  Nonstop routing state: Not in sync
  Next-hop addresses received:
    10.255.0.2
    192.168.12.2
    2001:db8::2
    fe80::21f:1200:cb6:4c8d
  Queue depth: 0
Message type               Total                 Last 5 seconds
                       Sent      Received          Sent      Received
Initialization            1             1             0             0
Keepalive                34            34             0             0
Notification              0             0             0             0
Address                   1             1             0             0
Address withdraw          0             0             0             0
Label mapping             3             3             0             0
Label request             0             0             0             0
Label withdraw            0             0             0             0
Label release             0             0             0             0
Label abort               0             0             0             0
```

Meaning

The output displays information for the LDP session using IPv6 as the TCP transport.

Verification

IN THIS SECTION

- Verifying the LDP Neighbor Information | **1505**
- Verifying the LDP Session Information | **1506**

Confirm that the configuration is working properly.

Verifying the LDP Neighbor Information

Purpose

Display the LDP neighbor information.

Action

On Device R1, from operational mode, run the `show ldp neighbor extensive` command to display LDP neighbor information.

```
user@R1> show ldp neighbor extensive
Address                          Interface       Label space ID     Hold time
192.168.12.2                     ge-1/0/0.0      10.255.0.2:0         14
  Transport address: 10.255.0.2, Transport preference: IPv4, Configuration sequence: 9
  Up for 00:00:14
  Reference count: 1
  Hold time: 15, Proposed local/peer: 15/15
  Hello flags: none
  Neighbor types: discovered
Address                          Interface       Label space ID     Hold time
fe80::21f:1200:cb6:4c8d          ge-1/0/0.0      10.255.0.2:0         14
  Transport address: 2001:db8::2, Transport preference: IPv4, Configuration sequence: 9
  Up for 00:00:14
  Reference count: 1
  Hold time: 15, Proposed local/peer: 15/15
  Hello flags: none
  Neighbor types: discovered
```

Meaning

The output shows LDP neighbor information for both the IPv4 and IPv6 addresses.

Verifying the LDP Session Information

Purpose

Display the LDP session information.

Action

On Device R1, from operational mode, run the `show ldp session extensive` command to display LDP session information.

```
user@R1> show ldp session extensive
Address: 10.255.0.2, State: Operational, Connection: Open, Hold time: 24
  Session ID: 10.255.0.1:0--10.255.0.2:0
  Next keepalive in 4 seconds
  Passive, Maximum PDU: 4096, Hold time: 30, Neighbor count: 2
  Neighbor types: discovered
  Keepalive interval: 10, Connect retry interval: 1
  Local address: 10.255.0.1, Remote address: 10.255.0.2
  Up for 00:05:26
  Capabilities advertised: none
  Capabilities received: none
  Protection: disabled
  Session flags: none
  Local - Restart: disabled, Helper mode: enabled
  Remote - Restart: disabled, Helper mode: enabled
  Local maximum neighbor reconnect time: 120000 msec
  Local maximum neighbor recovery time: 240000 msec
  Local Label Advertisement mode: Downstream unsolicited
  Remote Label Advertisement mode: Downstream unsolicited
  Negotiated Label Advertisement mode: Downstream unsolicited
  MTU discovery: disabled
  Nonstop routing state: Not in sync
  Next-hop addresses received:
    10.255.0.2
    192.168.12.2
    2001:db8::2
    fe80::21f:1200:cb6:4c8d
```

```
  Queue depth: 0
Message type              Total                   Last 5 seconds
                      Sent      Received          Sent      Received
Initialization           1             1             0             0
Keepalive               33            33             1             1
Notification             0             0             0             0
Address                  2             2             0             0
Address withdraw         0             0             0             0
Label mapping            6             6             0             0
Label request            0             0             0             0
Label withdraw           0             0             0             0
Label release            0             0             0             0
Label abort              0             0             0             0
```

Meaning

The output displays information for the LDP session using IPv6 as the TCP transport.

Verification

IN THIS SECTION

- Verifying the LDP Neighbor Information | **1507**
- Verifying the LDP Session Information | **1508**

Confirm that the configuration is working properly.

Verifying the LDP Neighbor Information

Purpose

Display the LDP neighbor information.

Action

On Device R1, from operational mode, run the `show ldp neighbor extensive` command to display LDP neighbor information.

```
user@R1> show ldp neighbor extensive
Address                          Interface       Label space ID     Hold time
192.168.12.2                     ge-1/0/0.0      10.255.0.2:0         11
  Transport address: 10.255.0.2, Configuration sequence: 10
  Up for 00:04:35
  Reference count: 1
  Hold time: 15, Proposed local/peer: 15/15
  Hello flags: none
  Neighbor types: discovered
Address                          Interface       Label space ID     Hold time
fe80::21f:1200:cb6:4c8d          ge-1/0/0.0      10.255.0.2:0         14
  Transport address: 2001:db8::2, Configuration sequence: 10
  Up for 00:04:35
  Reference count: 1
  Hold time: 15, Proposed local/peer: 15/15
  Hello flags: none
  Neighbor types: discovered
```

Meaning

The output shows LDP neighbor information for both the IPv4 and IPv6 addresses.

Verifying the LDP Session Information

Purpose

Display the LDP session information.

Action

On Device R1, from operational mode, run the `show ldp session extensive` command to display LDP neighbor information.

```
user@R1> show ldp session extensive
Address: 2001:db8::2, State: Operational, Connection: Open, Hold time: 29
```

```
  Session ID: 10.1.1.1:0--10.255.0.2:0
  Next keepalive in 9 seconds
  Passive, Maximum PDU: 4096, Hold time: 30, Neighbor count: 1
  Neighbor types: discovered
  Keepalive interval: 10, Connect retry interval: 1
  Local address: 2001:db8::1, Remote address: 2001:db8::2
  Up for 00:05:31
  Capabilities advertised: none
  Capabilities received: none
  Protection: disabled
  Session flags: none
  Local - Restart: disabled, Helper mode: enabled
  Remote - Restart: disabled, Helper mode: enabled
  Local maximum neighbor reconnect time: 120000 msec
  Local maximum neighbor recovery time: 240000 msec
  Local Label Advertisement mode: Downstream unsolicited
  Remote Label Advertisement mode: Downstream unsolicited
  Negotiated Label Advertisement mode: Downstream unsolicited
  MTU discovery: disabled
  Nonstop routing state: Not in sync
  Next-hop addresses received:
    2001:db8::2
    fe80::21f:1200:cb6:4c8d
  Queue depth: 0
Message type               Total                    Last 5 seconds
                       Sent      Received          Sent      Received
Initialization            1             1             0             0
Keepalive                34            34             0             0
Notification              0             0             0             0
Address                   1             1             0             0
Address withdraw          0             0             0             0
Label mapping             3             3             0             0
Label request             0             0             0             0
Label withdraw            0             0             0             0
Label release             0             0             0             0
Label abort               0             0             0             0

Address: 10.255.0.2, State: Operational, Connection: Open, Hold time: 29
  Session ID: 10.255.0.1:0--10.255.0.2:0
  Next keepalive in 9 seconds
  Passive, Maximum PDU: 4096, Hold time: 30, Neighbor count: 1
  Neighbor types: discovered
  Keepalive interval: 10, Connect retry interval: 1
```

```
  Local address: 10.255.0.1, Remote address: 10.255.0.2
  Up for 00:05:31
  Capabilities advertised: none
  Capabilities received: none
  Protection: disabled
  Session flags: none
  Local - Restart: disabled, Helper mode: enabled
  Remote - Restart: disabled, Helper mode: enabled
  Local maximum neighbor reconnect time: 120000 msec
  Local maximum neighbor recovery time: 240000 msec
  Local Label Advertisement mode: Downstream unsolicited
  Remote Label Advertisement mode: Downstream unsolicited
  Negotiated Label Advertisement mode: Downstream unsolicited
  MTU discovery: disabled
  Nonstop routing state: Not in sync
  Next-hop addresses received:
    10.255.0.2
    192.168.12.2
  Queue depth: 0
Message type               Total                 Last 5 seconds
                       Sent      Received          Sent      Received
Initialization            1             1             0             0
Keepalive                34            34             0             0
Notification              0             0             0             0
Address                   1             1             0             0
Address withdraw          0             0             0             0
Label mapping             3             3             0             0
Label request             0             0             0             0
Label withdraw            0             0             0             0
Label release             0             0             0             0
Label abort               0             0             0             0
```

Example: Configuring Multipoint LDP In-Band Signaling for Point-to-Multipoint LSPs

IN THIS SECTION

- Understanding Multipoint LDP Inband Signaling for Point-to-Multipoint LSPs | **1511**
- Example: Configuring Multipoint LDP In-Band Signaling for Point-to-Multipoint LSPs | **1522**

Understanding Multipoint LDP Inband Signaling for Point-to-Multipoint LSPs

IN THIS SECTION

- How M-LDP Works | **1512**
- Terminology | **1517**
- Ingress Join Translation and Pseudo Interface Handling | **1518**
- Ingress Splicing | **1518**
- Reverse Path Forwarding | **1518**
- LSP Root Detection | **1518**
- Egress Join Translation and Pseudo Interface Handling | **1519**
- Egress Splicing | **1519**
- Supported Functionality | **1519**
- Unsupported Functionality | **1520**
- LDP Functionality | **1520**
- Egress LER Functionality | **1521**
- Transit LSR Functionality | **1521**
- Ingress LER Functionality | **1521**

The Multipoint Label Distribution Protocol (M-LDP) for point-to-multipoint label-switched paths (LSPs) with in-band signaling is useful in a deployment with an existing IP/MPLS backbone, in which you need to carry multicast traffic, for IPTV for example.

For years, the most widely used solution for transporting multicast traffic has been to use native IP multicast in the service provider core with multipoint IP tunneling to isolate customer traffic. A multicast routing protocol, usually Protocol Independent Multicast (PIM), is deployed to set up the forwarding paths. IP multicast routing is used for forwarding, using PIM signaling in the core. For this model to work, the core network has to be multicast enabled. This allows for effective and stable deployments even in inter-autonomous system (AS) scenarios.

However, in an existing IP/MPLS network, deploying PIM might not be the first choice. Some service providers are interested in replacing IP tunneling with MPLS label encapsulation. The motivations for moving to MPLS label switching is to leverage MPLS traffic engineering and protection features and to reduce the amount of control traffic overhead in the provider core.

To do this, service providers are interested in leveraging the extension of the existing deployments to allow multicast traffic to pass through. The existing multicast extensions for IP/MPLS are point-to-

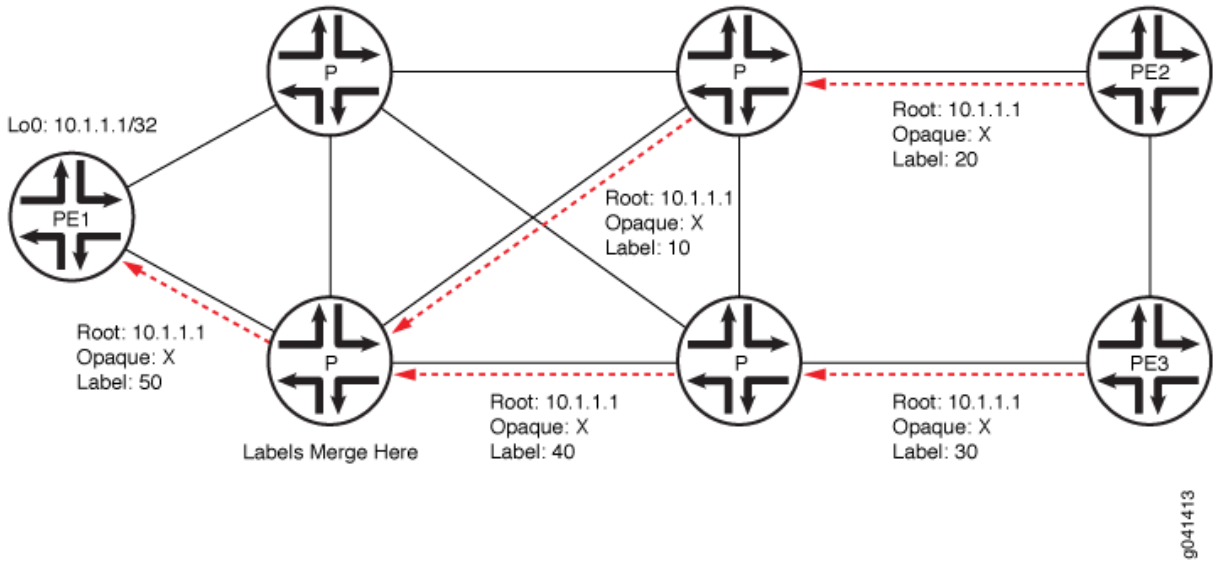
multipoint extensions for RSVP-TE and point-to-multipoint and multipoint-to-multipoint extensions for LDP. These deployment scenarios are discussed in RFC 6826, *Multipoint LDP In-Band Signaling for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths*. This feature overview is limited to point-to-multipoint extensions for LDP.

How M-LDP Works

Label Bindings in M-LDP Signaling

The multipoint extension to LDP uses point-to-multipoint and multipoint-to-multipoint forwarding equivalence class (FEC) elements (defined in RFC 5036, *LDP Specification*) along with capability advertisements, label mapping, and signaling procedures. The FEC elements include the idea of the LSP root, which is an IP address, and an "opaque" value, which is a selector that groups together the leaf nodes sharing the same opaque value. The opaque value is transparent to the intermediate nodes, but has meaning for the LSP root. Every LDP node advertises its local incoming label binding to the upstream LDP node on the shortest path to the root IP address found in the FEC. The upstream node receiving the label bindings creates its own local label and outgoing interfaces. This label allocation process might result in packet replication, if there are multiple outgoing branches. As shown in Figure 18, an LDP node merges the label bindings for the same opaque value if it finds downstream nodes sharing the same upstream node. This allows for effective building of point-to-multipoint LSPs and label conservation.

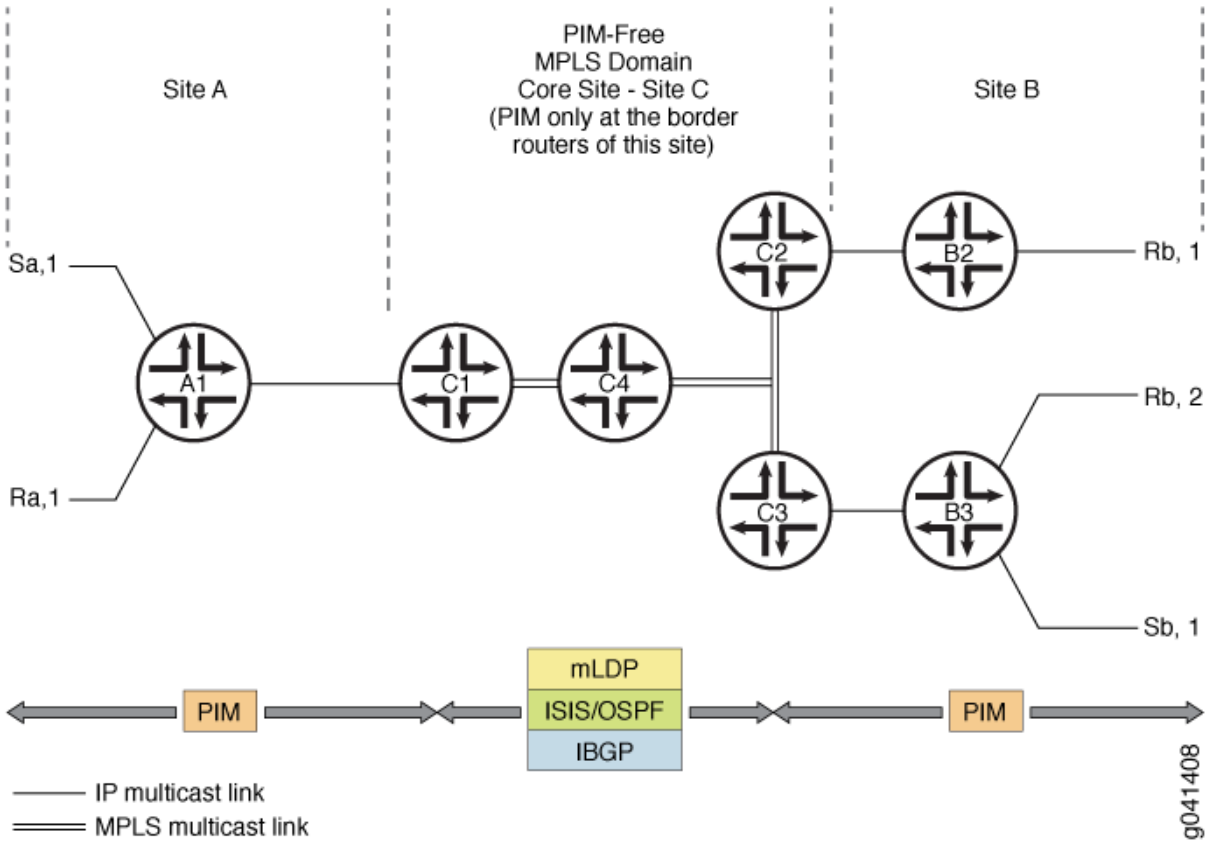
Figure 98: Label Bindings in M-LDP Signaling



M-LDP in PIM-Free MPLS Core

Figure 19 shows a scaled-down deployment scenario. Two separate PIM domains are interconnected by a PIM-free core site. The border routers in this core site support PIM on the border interfaces. Further, these border routers collect and distribute the routing information from the adjacent sites to the core network. The edge routers in Site C run BGP for root-node discovery. Interior gateway protocol (IGP) routes cannot be used for ingress discovery because in most cases the forwarding next hop provided by the IGP would not provide information about the ingress device toward the source. M-LDP inband signaling has a one-to-one mapping between the point-to-multipoint LSP and the (S,G) flow. With in-band signaling, PIM messages are directly translated into M-LDP FEC bindings. In contrast, out-of-band signaling is based on manual configuration. One application for M-LDP inband signaling is to carry IPTV multicast traffic in an MPLS backbone.

Figure 99: Sample M-LDP Topology in PIM-Free MPLS Core



Configuration

The *configuration statement* `mldp-inband-signalling` on the label-edge router (LER) enables PIM to use M-LDP in-band signaling for the upstream neighbors when the LER does not detect a PIM upstream

neighbor. Static configuration of the MPLS LSP root is included in the PIM configuration, using policy. This is needed when IBGP is not available in the core site or to override IBGP-based LSP root detection.

For example:

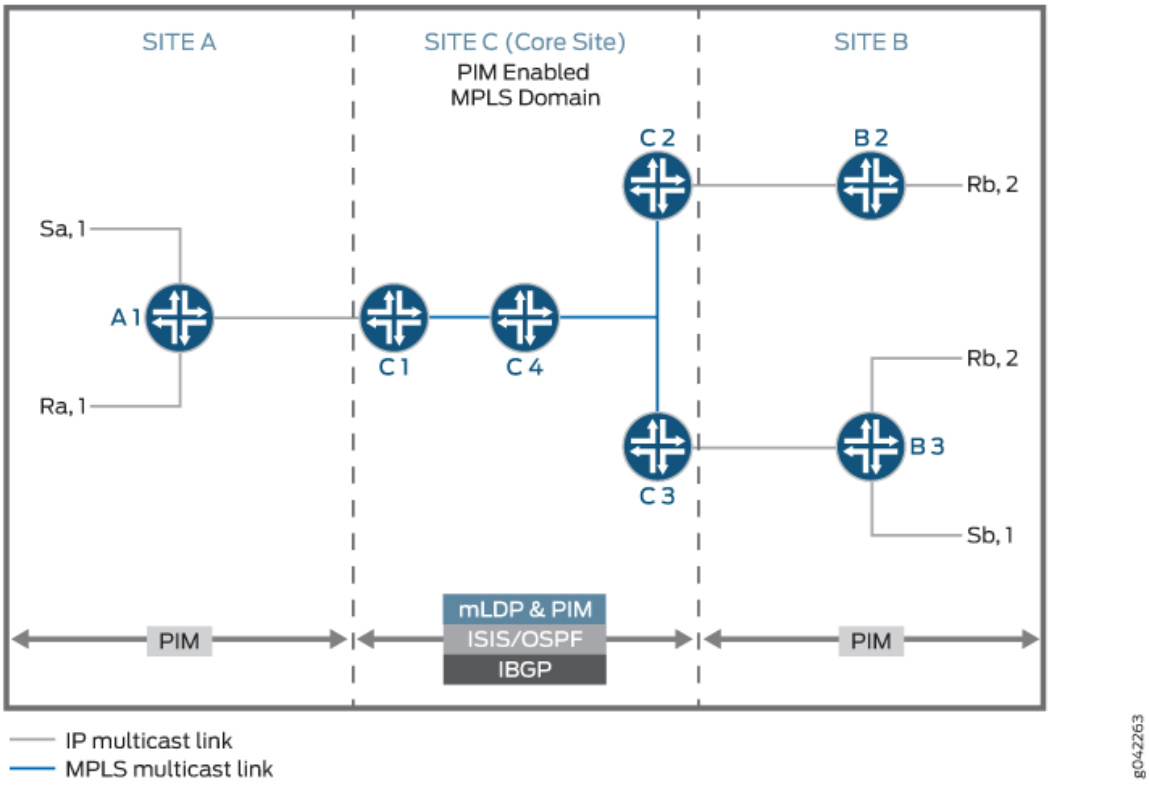
```
protocols {
    pim {
        mldp-inband-signalling {
            policy lsp-mapping-policy-example;
        }
    }
}
```

```
policy-options {
    policy-statement lsp-mapping-policy-example {
        term channel1 {
            from {
                source-address-filter ip-prefix</prefix-length>;  #policy filter for channel1
            }
            then {
                p2mp-lsp-root {
                    # Statically configured ingress address of edge
                    # used by channel1
                    address ip-address;
                }
                accept;
            }
        }
    }
}
```

M-LDP in PIM-Enabled MPLS Core

Starting in Junos OS Release 14.1, in order to migrate existing IPTV services from native IP multicast to MPLS multicast, you need to smoothly transition from PIM to M-LDP point-to-multipoint LSPs with minimal outage. Figure 20 shows a similar M-LDP topology as Figure 19, but with a different scenario. The core is enabled with PIM, with one source streaming all the IPTV channels. The TV channels are sent as ASM streams with each channel identified by its group address. Previously, these channels were streamed on the core as IP streams and signaled using PIM.

Figure 100: Sample M-LDP Topology in PIM-Enabled MPLS Core



By configuring the `mldp-inband-signaling` in this scenario, M-LDP signaling is initiated only when there is no PIM neighbor towards the source. However, because there is always a PIM neighbor towards the source unless PIM is deactivated on the upstream interfaces of the egress PE, PIM takes precedence over M-LDP and M-LDP does not take effect.

Configuration

To progressively migrate channel by channel to M-LDP MPLS core with few streams using M-LDP upstream and other streams using existing PIM upstream, include the `selected-mldp-egress` configuration statement along with group based filters in the policy filter for M-LDP inband signaling.

NOTE: The M-LDP inband signaling policy filter can include either the `source-address-filter` statement or the `route-filter` statement, or a combination of both.

For example:

```
protocols {
    pim {
```

```
        mldp-inband-signalling {
            policy lsp-mapping-policy-example;
        }
    }
}
```

```
policy-options {
    policy-statement lsp-mapping-policy-example {
        term channel1 {
            from {
                source-address-filter ip-prefix</prefix-length>;  #policy filter for channel1
            }
            then {
                selected-mldp-egress;
                accept;
            }
        }
        term channel2 {
            from {
                source-address-filter ip-prefix</prefix-length>;  #policy filter for channel2
                route-filter ip-prefix</prefix-length>;  #policy filter on multicast group
address
            }
            then {
                selected-mldp-egress;
                p2mp-lsp-root {
                    # Statically configured ingress address of edge
                    # used by channel2
                    address ip-address;
                }
                accept;
            }
        }
        term channel3 {
            from {
                route-filter ip-prefix</prefix-length>;  #policy filter on multicast group
address
            }
            then {
                selected-mldp-egress;
                accept;
```

```
            }
        }
    }
}
```

NOTE: Some of the limitations of the above configuration are as follows:

- The `selected-mldp-egress` statement should be configured only on the LER. Configuring the `selected-mldp-egress` statement on non-egress PIM routers can cause path setup failures.
- When policy changes are made to switch traffic from PIM upstream to M-LDP upstream and vice-versa, packet loss can be expected as break-and-make mechanism is performed at the control plane.

Terminology

The following terms are important for an understanding of M-LDP in-band signaling for multicast traffic.

Point-to-point LSP	An LSP that has one ingress label-switched router (LSR) and one egress LSR.
Multipoint LSP	Either a point-to-multipoint or a multipoint-to-multipoint LSP.
Point-to-multipoint LSP	An LSP that has one ingress LSR and one or more egress LSRs.
Multipoint-to-point LSP	An LSP that has one or more ingress LSRs and one unique egress LSR.
Multipoint-to-multipoint LSP	An LSP that connects a set of nodes, such that traffic sent by any node in the LSP is delivered to all others.
Ingress LSR	An ingress LSR for a particular LSP is an LSR that can send a data packet along the LSP. Multipoint-to-multipoint LSPs can have multiple ingress LSRs. Point-to-multipoint LSPs have only one, and that node is often referred to as the root node.
Egress LSR	An egress LSR for a particular LSP is an LSR that can remove a data packet from that LSP for further processing. Point-to-point and multipoint-to-point LSPs have only a single egress node. Point-to-multipoint and multipoint-to-multipoint LSPs can have multiple egress nodes.
Transit LSR	An LSR that has reachability to the root of the multipoint LSP through a directly connected upstream LSR and one or more directly connected downstream LSRs.

Bud LSR An LSR that is an egress but also has one or more directly connected downstream LSRs.

Leaf node Either an egress or bud LSR in the context of a point-to-multipoint LSP. In the context of a multipoint-to-multipoint LSP, an LSR is both ingress and egress for the same multipoint-to-multipoint LSP and can also be a bud LSR.

Ingress Join Translation and Pseudo Interface Handling

At the ingress LER, LDP notifies PIM about the (S,G) messages that are received over the in-band signaling. PIM associates each (S,G) messagewith a pseudo interface. Subsequently, a shortest-path-tree (SPT) join message is initiated toward the source. PIM treats this as a new type of local receiver. When the LSP is torn down, PIM removes this local receiver based on notification from LDP.

Ingress Splicing

LDP provides PIM with a next hop to be associated with each (S,G) entry. PIM installs a PIM (S,G) multicast route with the LDP next hop and other PIM receivers. The next hop is a composite next hop of local receivers + the list of PIM downstream neighbors + a sub-level next hopfor the LDP tunnel.

Reverse Path Forwarding

PIM's reverse-path-forwarding (RPF) calculation is performed at the egress node.

PIM performs M-LDP in-band signaling when all of the following conditions are true:

- There are no PIM neighbors toward the source.
- The M-LDP in-band signaling statement is configured.
- The next hop is learned through BGP, or is present in the static mapping (specified in an M-LDP in-band signaling policy).

Otherwise, if LSP root detection fails, PIM retains the (S,G) entry with an RPF state of unresolved.

PIM RPF registers this source address each time unicast routing information changes. Therefore, if the route toward the source changes, the RPF recalculation recurs. BGP protocol next hops toward the source too are monitored for changes in the LSP root. Such changes might cause traffic disruption for short durations.

LSP Root Detection

If the RPF operation detects the need for M-LDP in-band signaling upstream, the LSP root (ingress) is detected. This root is a parameter for LDP LSP signaling.

The root node is detected as follows:

1. If the existing static configuration specifies the source address, the root is taken as given in configuration.

2. A lookup is performed in the unicast routing table. If the source address is found, the protocol next hop toward the source is used as the LSP root.

   Prior to Junos OS Release 16.1, M-LDP point-to-multipoint LSP is signaled from an egress to ingress using the root address of the ingress LSR. This root address is reachable through IGP only, thereby confining the M-LDP point-to-multipoint LSP to a single autonomous system. If the root address is not reachable through an IGP, but reachable through BGP, and if that BGP route is recursively resolved over an MPLS LSP, then the point-to-multipoint LSP is not signaled further from that point towards the ingress LSR root address.

   There is a need for these non-segmented point-to-multipoint LSPs to be signaled across multiple autonomous systems, which can be used for the following applications:

   - Inter-AS MVPN with non-segmented point-to-multipoint LSPs.
   - Inter-AS M-LDP inband signaling between client networks connected by an MPLS core network.
   - Inter-area MVPN or M-LDP inband signaling with non-segmented point-to-multipoint LSPs (seamless MPLS multicast).

   Starting from Junos OS Release 16.1, M-LDP can signal point-to-multipoint LSPs at ASBR or transit or egress when root address is a BGP route which is further recursively resolved over an MPLS LSP.

Egress Join Translation and Pseudo Interface Handling

At the egress LER, PIM notifies LDP of the (S,G) message to be signaled along with the LSP root. PIM creates a pseudo interface as the upstream interface for this (S,G) message. When an (S,G) prune message is received, this association is removed.

Egress Splicing

At the egress node of the core network, where the (S,G) join message from the downstream site is received, this join message is translated to M-LDP in-band signaling parameters and LDP is notified. Further, LSP teardown occurs when the (S,G) entry is lost, when the LSP root changes, or when the (S,G) entry is reachable over a PIM neighbor.

Supported Functionality

For M-LDP in-band signaling, Junos OS supports the following functionality:

- Egress splicing of the PIM next hop with the LDP route

- Ingress splicing of the PIM route with the LDP next hop
- Translation of PIM join messages to LDP point-to-multipoint LSP setup parameters
- Translation of M-LDP in-band LSP parameters to set up PIM join messages
- Statically configured and BGP protocol next hop-based LSP root detection
- PIM (S,G) states in the PIM source-specific multicast (SSM) and anysource multicsast (ASM) ranges
- Configuration statements on ingress and egress LERs to enable them to act as edge routers
- IGMP join messages on LERs
- Carrying IPv6 source and group address as opaque information toward an IPv4 root node
- Static configuration to map an IPv6 (S,G) to an IPv4 root address

Unsupported Functionality

For M-LDP in-band signaling, Junos OS does *not* support the following functionality:

- Full support for PIM ASM
- The `mpls lsp point-to-multipoint ping` command with an (S,G) option
- *Nonstop active routing* (NSR)
- Make-before-break (MBB) for PIM
- IPv6 LSP root addresses (LDP does not support IPv6 LSPs.)
- Neighbor relationship between PIM speakers that are not directly connected
- Graceful restart
- PIM dense mode
- PIM bidirectional mode

LDP Functionality

The PIM (S,G) information is carried as M-LDP opaque type-length-value (TLV) encodings. The point-to-multipoint FEC element consists of the root-node address. In the case of next-generation multicast VPNs (NGEN MVPNs), the point-to-multipoint LSP is identified by the root node address and the LSP ID.

Egress LER Functionality

On the egress LER, PIM triggers LDP with the following information to create a point-to-multipoint LSP:

- Root node
- (S,G)
- Next hop

PIM finds the root node based on the source of the multicast tree. If the root address is configured for this (S,G) entry, the configured address is used as the point-to-multipoint LSP root. Otherwise, the routing table is used to look up the route to the source. If the route to the source of the multicast tree is a BGP-learned route, PIM retrieves the BGP next hop address and uses it as the root node for the point-to-multipoint LSP.

LDP finds the upstream node based on the root node, allocates a label, and sends the label mapping to the upstream node. LDP does not use penultimate hop popping (PHP) for in-band M-LDP signaling.

If the root addresses for the source of the multicast tree changes, PIM deletes the point-to-multipoint LSP and triggers LDP to create a new point-to-multipoint LSP. When this happens, the outgoing interface list becomes NULL, PIM triggers LDP to delete the point-to-multipoint LSP, and LDP sends a label withdraw message to the upstream node.

Transit LSR Functionality

The transit LSR advertises a label to the upstream LSR toward the source of the point-to-multipoint FEC and installs the necessary forwarding state to forward the packets. The transit LSR can be any M-LDP capable router.

Ingress LER Functionality

On the ingress LER, LDP provides the following information to PIM upon receiving the label mapping:

- (S,G)
- Flood next hop

Then PIM installs the forwarding state. If the new branches are added or deleted, the flood next hop is updated accordingly. If all branches are deleted due to a label being withdrawn, LDP sends updated information to PIM. If there are multiple links between the upstream and downstream neighbors, the point-to-multipoint LSP is not load balanced.

SEE ALSO

LDP Configuration | 1362

Example: Configuring Multipoint LDP In-Band Signaling for Point-to-Multipoint LSPs

IN THIS SECTION

- Requirements | 1522
- Overview | 1522
- Configuration | 1523
- Verification | 1535

This example shows how to configure multipoint LDP (M-LDP) in-band signaling for multicast traffic, as an extension to the Protocol Independent Multicast (PIM) protocol or as a substitute for PIM.

Requirements

This example can be configured using the following hardware and software components:

- Junos OS Release 13.2 or later
- MX Series 5G Universal Routing Platforms or M Series Multiservice Edge Routers for the Provider Edge (PE) Routers
- PTX Series Packet Transport Routers acting as transit label-switched routers
- T Series Core Routers for the Core Routers

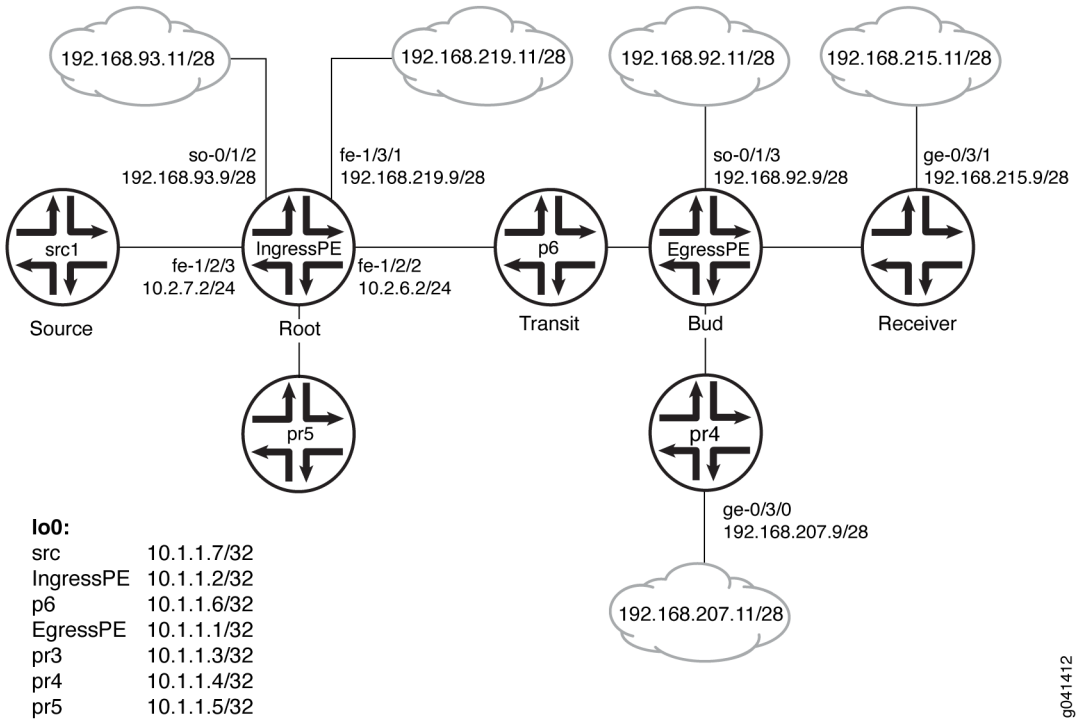
NOTE: The PE routers could also be T Series Core Routers but that is not typical. Depending on your scaling requirements, the core routers could also be MX Series 5G Universal Routing Platforms or M Series Multiservice Edge Routers. The Customer Edge (CE) devices could be other routers or switches from Juniper Networks or another vendor.

No special configuration beyond device initialization is required before configuring this example.

Overview

"CLI Quick Configuration" on page 1523 shows the configuration for all of the devices in Figure 101 on page 1523. The section "No Link Title" on page 1528 describes the steps on Device EgressPE.

Figure 101: M-LDP In-Band Signaling for Point-to-Multipoint LSPs Example Topology



Configuration

IN THIS SECTION

- Procedure | **1523**

Procedure

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the `[edit]` hierarchy level.

Device src1

```
set logical-systems src1 interfaces fe-1/2/0 unit 0 family inet address 10.2.7.7/24
set logical-systems src1 interfaces lo0 unit 0 family inet address 10.1.1.7/32
set logical-systems src1 protocols ospf area 0.0.0.0 interface all
```

Device IngressPE

```
set interfaces so-0/1/2 unit 0 family inet address 192.168.93.9/28
set interfaces fe-1/2/0 unit 0 family inet address 10.2.3.2/24
set interfaces fe-1/2/0 unit 0 family mpls
set interfaces fe-1/2/1 unit 0 family inet address 10.2.5.2/24
set interfaces fe-1/2/2 unit 0 family inet address 10.2.6.2/24
set interfaces fe-1/2/2 unit 0 family mpls
set interfaces fe-1/2/3 unit 0 family inet address 10.2.7.2/24
set interfaces fe-1/3/1 unit 0 family inet address 192.168.219.9/28
set interfaces lo0 unit 0 family inet address 10.1.1.2/32
set protocols igmp interface fe-1/2/1.0 version 3
set protocols igmp interface fe-1/2/1.0 static group 232.1.1.1 source 192.168.219.11
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.1.1.2
set protocols bgp group ibgp family inet any
set protocols bgp group ibgp family inet-vpn any
set protocols bgp group ibgp neighbor 10.1.1.3
set protocols bgp group ibgp neighbor 10.1.1.4
set protocols bgp group ibgp neighbor 10.1.1.1
set protocols ospf area 0.0.0.0 interface all
set protocols ldp interface fe-1/2/0.0
set protocols ldp interface fe-1/2/2.0
set protocols ldp interface lo0.0
set protocols ldp p2mp
set protocols pim mldp-inband-signalling policy mldppim-ex
set protocols pim rp static address 10.1.1.5
set protocols pim interface fe-1/3/1.0
set protocols pim interface lo0.0
set protocols pim interface fe-1/2/0.21
set protocols pim interface fe-1/2/3.0
set protocols pim interface fe-1/2/1.0
set protocols pim interface so-0/1/2.0
set policy-options policy-statement mldppim-ex term B from source-address-filter 192.168.0.0/24
orlonger
```

```
set policy-options policy-statement mldppim-ex term B from source-address-filter
192.168.219.11/32 orlonger
set policy-options policy-statement mldppim-ex term B then accept
set policy-options policy-statement mldppim-ex term A from source-address-filter 10.1.1.7/32
orlonger
set policy-options policy-statement mldppim-ex term A from source-address-filter 10.2.7.0/24
orlonger
set policy-options policy-statement mldppim-ex term A then accept
set routing-options autonomous-system 64510
```

Device EgressPE

```
set interfaces so-0/1/3 unit 0 point-to-point
set interfaces so-0/1/3 unit 0 family inet address 192.168.92.9/28
set interfaces fe-1/2/0 unit 0 family inet address 10.1.3.1/24
set interfaces fe-1/2/0 unit 0 family mpls
set interfaces fe-1/2/1 unit 0 family inet address 10.1.4.1/24
set interfaces fe-1/2/2 unit 0 family inet address 10.1.6.1/24
set interfaces fe-1/2/2 unit 0 family mpls
set interfaces fe-1/3/0 unit 0 family inet address 192.168.209.9/28
set interfaces lo0 unit 0 family inet address 10.1.1.1/32
set routing-options autonomous-system 64510
set protocols igmp interface fe-1/3/0.0 version 3
set protocols igmp interface fe-1/3/0.0 static group 232.1.1.1 group-count 3
set protocols igmp interface fe-1/3/0.0 static group 232.1.1.1 source 192.168.219.11
set protocols igmp interface fe-1/3/0.0 static group 227.1.1.1
set protocols igmp interface so-0/1/3.0 version 3
set protocols igmp interface so-0/1/3.0 static group 232.1.1.1 group-count 2
set protocols igmp interface so-0/1/3.0 static group 232.1.1.1 source 192.168.219.11
set protocols igmp interface so-0/1/3.0 static group 232.2.2.2 source 10.2.7.7
set protocols mpls interface fe-1/2/0.0
set protocols mpls interface fe-1/2/2.0
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.1.1.1
set protocols bgp group ibgp family inet any
set protocols bgp group ibgp neighbor 10.1.1.2
set protocols msdp local-address 10.1.1.1
set protocols msdp peer 10.1.1.5
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ldp interface fe-1/2/0.0
set protocols ldp interface fe-1/2/2.0
```

```
set protocols ldp interface lo0.0
set protocols ldp p2mp
set protocols pim mldp-inband-signalling policy mldppim-ex
set protocols pim rp local address 10.1.1.1
set protocols pim rp local group-ranges 227.0.0.0/8
set protocols pim rp static address 10.1.1.4
set protocols pim rp static address 10.2.7.7 group-ranges 226.0.0.0/8
set protocols pim interface lo0.0
set protocols pim interface fe-1/3/0.0
set protocols pim interface fe-1/2/0.0
set protocols pim interface fe-1/2/1.0
set protocols pim interface so-0/1/3.0
set policy-options policy-statement mldppim-ex term B from source-address-filter 192.168.0.0/24
orlonger
set policy-options policy-statement mldppim-ex term B from source-address-filter
192.168.219.11/32 orlonger
set policy-options policy-statement mldppim-ex term B then p2mp-lsp-root address 10.1.1.2
set policy-options policy-statement mldppim-ex term B then accept
set policy-options policy-statement mldppim-ex term A from source-address-filter 10.2.7.0/24
orlonger
set policy-options policy-statement mldppim-ex term A then accept
```

Device p6

```
set interfaces fe-1/2/0 unit 0 family inet address 10.1.6.6/24
set interfaces fe-1/2/0 unit 0 family mpls
set interfaces fe-1/2/1 unit 0 family inet address 10.2.6.6/24
set interfaces fe-1/2/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.1.1.6/32
set interfaces lo0 unit 0 family mpls
set protocols ospf area 0.0.0.0 interface all
set protocols ldp interface fe-1/2/0.0
set protocols ldp interface fe-1/2/1.0
set protocols ldp interface lo0.0
set protocols ldp p2mp
```

Device pr3

```
set interfaces ge-0/3/1 unit 0 family inet address 192.168.215.9/28
set interfaces fe-1/2/0 unit 0 family inet address 10.1.3.3/24
set interfaces fe-1/2/0 unit 0 family mpls
```

```
set interfaces fe-1/2/1 unit 0 family inet address 10.2.3.3/24
set interfaces fe-1/2/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.1.1.3/32
set protocols igmp interface ge-0/3/1.0 version 3
set protocols igmp interface ge-0/3/1.0 static group 232.1.1.2 source 192.168.219.11
set protocols igmp interface ge-0/3/1.0 static group 232.2.2.2 source 10.2.7.7
set protocols bgp group ibgp local-address 10.1.1.3
set protocols bgp group ibgp type internal
set protocols bgp group ibgp neighbor 10.1.1.2
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fe-1/2/1.0 metric 2
set protocols ldp interface fe-1/2/0.0
set protocols ldp interface fe-1/2/1.0
set protocols ldp interface lo0.0
set protocols ldp p2mp
set protocols pim mldp-inband-signalling policy mldppim-ex
set protocols pim interface fe-0/3/1.0
set protocols pim interface lo0.0
set policy-options policy-statement mldppim-ex term B from source-address-filter 192.168.0.0/24
orlonger
set policy-options policy-statement mldppim-ex term B from source-address-filter
192.168.219.11/32 orlonger
set policy-options policy-statement mldppim-ex term B then p2mp-lsp-root address 10.1.1.2
set policy-options policy-statement mldppim-ex term B then accept
set policy-options policy-statement mldppim-ex term B from source-address-filter 192.168.0.0/24
orlonger
set policy-options policy-statement mldppim-ex term B from source-address-filter
192.168.219.11/32 orlonger
set policy-options policy-statement mldppim-ex term B from source-address-filter 10.2.7.7/32
orlonger
set policy-options policy-statement mldppim-ex term B then p2mp-lsp-root address 10.1.1.2
set policy-options policy-statement mldppim-ex term B then accept
set routing-options autonomous-system 64510
```

Device pr4

```
set interfaces ge-0/3/0 unit 0 family inet address 192.168.207.9/28
set interfaces fe-1/2/0 unit 0 family inet address 10.1.4.4/24
set interfaces fe-1/2/0 unit 0 family iso
set interfaces lo0 unit 0 family inet address 10.1.1.4/32
set protocols igmp interface ge-0/3/0.0 version 3
set protocols igmp interface ge-0/3/0.0 static group 232.1.1.2 source 192.168.219.11
```

```
set protocols igmp interface ge-0/3/0.0 static group 225.1.1.1
set protocols bgp group ibgp local-address 10.1.1.4
set protocols bgp group ibgp type internal
set protocols bgp group ibgp neighbor 10.1.1.2
set protocols msdp local-address 10.1.1.4
set protocols msdp peer 10.1.1.5
set protocols ospf area 0.0.0.0 interface all
set protocols pim rp local address 10.1.1.4
set protocols pim interface ge-0/3/0.0
set protocols pim interface lo0.0
set protocols pim interface fe-1/2/0.0
set routing-options autonomous-system 64510
```

Device pr5

```
set interfaces fe-1/2/0 unit 0 family inet address 10.2.5.5/24
set interfaces lo0 unit 0 family inet address 10.1.1.5/24
set protocols igmp interface lo0.0 version 3
set protocols igmp interface lo0.0 static group 232.1.1.1 source 192.168.219.11
set protocols msdp local-address 10.1.1.5
set protocols msdp peer 10.1.1.4
set protocols msdp peer 10.1.1.1
set protocols ospf area 0.0.0.0 interface all
set protocols pim rp local address 10.1.1.5
set protocols pim interface all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure Device EgressPE:

1. Configure the interfaces.

   Enable MPLS on the core-facing interfaces. On the egress next hops, you do not need to enable MPLS.

   ```
   [edit interfaces]
   user@EgressPE# set fe-1/2/0 unit 0 family inet address 10.1.3.1/24
   user@EgressPE# set fe-1/2/0 unit 0 family mpls
   ```

```
user@EgressPE# set fe-1/2/2 unit 0 family inet address 10.1.6.1/24
user@EgressPE# set fe-1/2/2 unit 0 family mpls
user@EgressPE# set so-0/1/3 unit 0 point-to-point
user@EgressPE# set so-0/1/3 unit 0 family inet address 192.168.92.9/28
user@EgressPE# set fe-1/2/1 unit 0 family inet address 10.1.4.1/24
user@EgressPE# set fe-1/3/0 unit 0 family inet address 192.168.209.9/28
user@EgressPE# set lo0 unit 0 family inet address 10.1.1.1/32
```

2. Configure IGMP on the egress interfaces.

   For testing purposes, this example includes static group and source addresses.

```
[edit protocols igmp]
user@EgressPE# set interface fe-1/3/0.0 version 3
user@EgressPE# set interface fe-1/3/0.0 static group 232.1.1.1 group-count 3
user@EgressPE# set interface fe-1/3/0.0 static group 232.1.1.1 source 192.168.219.11
user@EgressPE# set interface fe-1/3/0.0 static group 227.1.1.1
user@EgressPE# set interface so-0/1/3.0 version 3
user@EgressPE# set interface so-0/1/3.0 static group 232.1.1.1 group-count 2
user@EgressPE# set interface so-0/1/3.0 static group 232.1.1.1 source 192.168.219.11
user@EgressPE# set interface so-0/1/3.0 static group 232.2.2.2 source 10.2.7.7
```

3. Configure MPLS on the core-facing interfaces.

```
[edit protocols mpls]
user@EgressPE# set interface fe-1/2/0.0
user@EgressPE# set interface fe-1/2/2.0
```

4. Configure BGP.

   BGP is a policy-driven protocol, so also configure and apply any needed routing policies.

   For example, you might want to export static routes into BGP.

```
[edit protocols bgp group ibgp]
user@EgressPE# set type internal
user@EgressPE# set local-address 10.1.1.1
user@EgressPE# set family inet any
user@EgressPE# set neighbor 10.1.1.2
```

5. (Optional) Configure an MSDP peer connection with Device pr5 in order to interconnect the disparate PIM domains, thus enabling redundant RPs.

```
[edit protocols msdp]
user@EgressPE# set local-address 10.1.1.1
user@EgressPE# set peer 10.1.1.5
```

6. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@EgressPE# set interface all
user@EgressPE# set interface fxp0.0 disable
```

7. Configure LDP on the core-facing interfaces and on the loopback interface.

```
[edit protocols ldp]
user@EgressPE# set interface fe-1/2/0.0
user@EgressPE# set interface fe-1/2/2.0
user@EgressPE# set interface lo0.0
```

8. Enable point-to-multipoint MPLS LSPs.

```
[edit protocols ldp]
user@EgressPE# set p2mp
```

9. Configure PIM on the downstream interfaces.

```
[edit protocols pim]
user@EgressPE# set interface lo0.0
user@EgressPE# set interface fe-1/3/0.0
user@EgressPE# set interface fe-1/2/1.0
user@EgressPE# set interface so-0/1/3.0
```

10. Configure the RP settings because this device serves as the PIM rendezvous point (RP).

```
[edit protocols pim]
user@EgressPE# set rp local address 10.1.1.1
```

```
user@EgressPE# set rp local group-ranges 227.0.0.0/8
user@EgressPE# set rp static address 10.1.1.4
user@EgressPE# set rp static address 10.2.7.7 group-ranges 226.0.0.0/8
```

11. Enable M-LDP in-band signaling and set the associated policy.

```
[edit protocols pim]
user@EgressPE# set mldp-inband-signalling policy mldppim-ex
```

12. Configure the routing policy that specifies the root address for the point-to-multipoint LSP and the associated source addresses.

```
[edit policy-options policy-statement mldppim-ex]
user@EgressPE# set term B from source-address-filter 192.168.0.0/24 orlonger
user@EgressPE# set term B from source-address-filter 192.168.219.11/32 orlonger
user@EgressPE# set term B then p2mp-lsp-root address 10.1.1.2
user@EgressPE# set term B then accept
user@EgressPE# set term A from source-address-filter 10.2.7.0/24 orlonger
user@EgressPE# set term A then accept
```

13. Configure the autonomous system (AS) ID.

```
[edit routing-options]
user@EgressPE# set autonomous-system 64510
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols`, `show policy-options`, and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

Device EgressPE

```
user@EgressPE# show interfaces
so-0/1/3 {
    unit 0 {
        point-to-point;
        family inet {
            address 192.168.92.9/28;
```

```
        }
    }
}
fe-1/2/0 {
    unit 0 {
        family inet {
            address 10.1.3.1/24;
        }
        family mpls;
    }
}
fe-1/2/1 {
    unit 0 {
        family inet {
            address 10.1.4.1/24;
        }
    }
}
fe-1/2/2 {
    unit 0 {
        family inet {
            address 10.1.6.1/24;
        }
        family mpls;
    }
}
fe-1/3/0 {
    unit 0 {
        family inet {
            address 192.168.209.9/28;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.1.1.1/32;
        }
```

```
    }
}
```

```
user@EgressPE# show protocols
igmp {
    interface fe-1/3/0.0 {
        version 3;
        static {
            group 232.1.1.1 {
                group-count 3;
                source 192.168.219.11;
            }
            group 227.1.1.1;
        }
    }
    interface so-0/1/3.0 {
        version 3;
        static {
            group 232.1.1.1 {
                group-count 2;
                source 192.168.219.11;
            }
            group 232.2.2.2 {
                source 10.2.7.7;
            }
        }
    }
}
mpls {
    interface fe-1/2/0.0;
    interface fe-1/2/2.0;
}
bgp {
    group ibgp {
        type internal;
        local-address 10.1.1.1;
        family inet {
            any;
        }
        neighbor 10.1.1.2;
    }
```

```
}
msdp {
    local-address 10.1.1.1;
    peer 10.1.1.5;
}
ospf {
    area 0.0.0.0 {
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
}
ldp {
    interface fe-1/2/0.0;
    interface fe-1/2/2.0;
    interface lo0.0;
    p2mp;
}
pim {
    mldp-inband-signalling {
        policy mldppim-ex;
    }
    rp {
        local {
            address 10.1.1.1;
            group-ranges {
                227.0.0.0/8;
            }
        }
        static {
            address 10.1.1.4;
            address 10.2.7.7 {
                group-ranges {
                    226.0.0.0/8;
                }
            }
        }
    }
    interface lo0.0;
    interface fe-1/3/0.0;
    interface fe-1/2/0.0;
    interface fe-1/2/1.0;
```

```
    interface so-0/1/3.0;
}
```

```
user@EgressPE# show policy-options
policy-statement mldppim-ex {
    term B {
        from {
            source-address-filter 192.168.0.0/24 orlonger;
            source-address-filter 192.168.219.11/32 orlonger;
        }
        then {
            p2mp-lsp-root {
                address 10.1.1.2;
            }
            accept;
        }
    }
    term A {
        from {
            source-address-filter 10.2.7.0/24 orlonger;
        }
        then accept;
    }
}
```

```
user@EgressPE# show routing-options
autonomous-system 64510;
```

Similarly, configure the other egress devices.

If you are done configuring the devices, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Checking the PIM Join States | **1536**
- Checking the PIM Sources | **1541**
- Checking the LDP Database | **1543**

Looking Up the Route Information for the MPLS Label | 1547
Checking the LDP Traffic Statistics | 1548

Confirm that the configuration is working properly.

Checking the PIM Join States

Purpose

Display information about PIM join states to verify the M-LDP in-band upstream and downstream details. On the ingress device, the `show pim join extensive` command displays `Pseudo-MLDP` for the downstream interface. On the egress, the `show pim join extensive` command displays `Pseudo-MLDP` for the upstream interface.

Action

From operational mode, enter the `show pim join extensive` command.

```
user@IngressPE> show pim join extensive

Instance: PIM.master Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 232.1.1.1
    Source: 192.168.219.11
    Flags: sparse,spt
    Upstream interface: fe-1/3/1.0
    Upstream neighbor: Direct
    Upstream state: Local Source
    Keepalive timeout:
    Uptime: 1d 23:00:12
    Downstream neighbors:
        Interface: Pseudo-MLDP
        Interface: fe-1/2/1.0
            10.2.5.2 State: Join Flags: S   Timeout: Infinity
            Uptime: 1d 23:00:12 Time since last Join: 1d 23:00:12

Group: 232.1.1.2
```

```
    Source: 192.168.219.11
    Flags: sparse,spt
    Upstream interface: fe-1/3/1.0
    Upstream neighbor: Direct
    Upstream state: Local Source
    Keepalive timeout:
    Uptime: 1d 22:59:59
    Downstream neighbors:
        Interface: Pseudo-MLDP

Group: 232.1.1.3
    Source: 192.168.219.11
    Flags: sparse,spt
    Upstream interface: fe-1/3/1.0
    Upstream neighbor: Direct
    Upstream state: Local Source
    Keepalive timeout:
    Uptime: 1d 22:07:31
    Downstream neighbors:
        Interface: Pseudo-MLDP

Group: 232.2.2.2
    Source: 10.2.7.7
    Flags: sparse,spt
    Upstream interface: fe-1/2/3.0
    Upstream neighbor: Direct
    Upstream state: Local Source
    Keepalive timeout:
    Uptime: 1d 22:59:59
    Downstream neighbors:
        Interface: Pseudo-MLDP


user@EgressPE> show pim join extensive

Instance: PIM.master Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 227.1.1.1
    Source: *
    RP: 10.1.1.1
    Flags: sparse,rptree,wildcard
    Upstream interface: Local
```

```
    Upstream neighbor: Local
    Upstream state: Local RP
    Uptime: 1d 23:14:21
    Downstream neighbors:
        Interface: fe-1/3/0.0
            192.168.209.9 State: Join Flags: SRW  Timeout: Infinity
            Uptime: 1d 23:14:21 Time since last Join: 1d 20:12:35

Group: 232.1.1.1
    Source: 192.168.219.11
    Flags: sparse,spt
    Upstream protocol: MLDP
    Upstream interface: Pseudo MLDP
    Upstream neighbor: MLDP LSP root <10.1.1.2>
    Upstream state: Join to Source
    Keepalive timeout:
    Uptime: 1d 23:14:22
    Downstream neighbors:
        Interface: so-0/1/3.0
            192.168.92.9 State: Join Flags: S   Timeout: Infinity
            Uptime: 1d 20:12:35 Time since last Join: 1d 20:12:35
    Downstream neighbors:
        Interface: fe-1/3/0.0
            192.168.209.9 State: Join Flags: S   Timeout: Infinity
            Uptime: 1d 20:12:35 Time since last Join: 1d 20:12:35

Group: 232.1.1.2
    Source: 192.168.219.11
    Flags: sparse,spt
    Upstream protocol: MLDP
    Upstream interface: Pseudo MLDP
    Upstream neighbor: MLDP LSP root <10.1.1.2>
    Upstream state: Join to Source
    Keepalive timeout:
    Uptime: 1d 23:14:22
    Downstream neighbors:
        Interface: so-0/1/3.0
            192.168.92.9 State: Join Flags: S   Timeout: Infinity
            Uptime: 1d 20:12:35 Time since last Join: 1d 20:12:35
    Downstream neighbors:
        Interface: fe-1/2/1.0
            10.1.4.4 State: Join Flags: S Timeout: 198
            Uptime: 1d 22:59:59 Time since last Join: 00:00:12
```

```
    Downstream neighbors:
        Interface: fe-1/3/0.0
            192.168.209.9 State: Join Flags: S   Timeout: Infinity
            Uptime: 1d 20:12:35 Time since last Join: 1d 20:12:35

Group: 232.1.1.3
    Source: 192.168.219.11
    Flags: sparse,spt
    Upstream protocol: MLDP
    Upstream interface: Pseudo MLDP
    Upstream neighbor: MLDP LSP root <10.1.1.2>
    Upstream state: Join to Source
    Keepalive timeout:
    Uptime: 1d 20:12:35
    Downstream neighbors:
        Interface: fe-1/3/0.0
            192.168.209.9 State: Join Flags: S   Timeout: Infinity
            Uptime: 1d 20:12:35 Time since last Join: 1d 20:12:35

Group: 232.2.2.2
    Source: 10.2.7.7
    Flags: sparse,spt
    Upstream protocol: MLDP
    Upstream interface: Pseudo MLDP
    Upstream neighbor: MLDP LSP root <10.1.1.2>
    Upstream state: Join to Source
    Keepalive timeout:
    Uptime: 1d 20:12:35
    Downstream neighbors:
        Interface: so-0/1/3.0
            192.168.92.9 State: Join Flags: S   Timeout: Infinity
            Uptime: 1d 20:12:35 Time since last Join: 1d 20:12:35

user@pr3> show pim join extensive

Instance: PIM.master Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 232.1.1.2
    Source: 192.168.219.11
    Flags: sparse,spt
    Upstream protocol: MLDP
    Upstream interface: Pseudo MLDP
```

```
    Upstream neighbor: MLDP LSP root <10.1.1.2>
    Upstream state: Join to Source
    Keepalive timeout:
    Uptime: 1d 20:14:40
    Downstream neighbors:
        Interface: Pseudo-GMP
            ge-0/3/1.0

Group: 232.2.2.2
    Source: 10.2.7.7
    Flags: sparse,spt
    Upstream protocol: MLDP
    Upstream interface: Pseudo MLDP
    Upstream neighbor: MLDP LSP root <10.1.1.2>
    Upstream state: Join to Source
    Keepalive timeout:
    Uptime: 1d 20:14:40
    Downstream neighbors:
        Interface: Pseudo-GMP
            ge-0/3/1.0


user@pr4> show pim join extensive
Instance: PIM.master Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 225.1.1.1
    Source: *
    RP: 10.1.1.4
    Flags: sparse,rptree,wildcard
    Upstream interface: Local
    Upstream neighbor: Local
    Upstream state: Local RP
    Uptime: 1d 23:13:43
    Downstream neighbors:
        Interface: ge-0/3/0.0
            192.168.207.9 State: Join Flags: SRW  Timeout: Infinity
            Uptime: 1d 23:13:43 Time since last Join: 1d 23:13:43

Group: 232.1.1.2
    Source: 192.168.219.11
    Flags: sparse,spt
    Upstream interface: fe-1/2/0.0
```

```
    Upstream neighbor: 10.1.4.1
    Upstream state: Local RP, Join to Source
    Keepalive timeout: 0
    Uptime: 1d 23:13:43
    Downstream neighbors:
        Interface: ge-0/3/0.0
            192.168.207.9 State: Join Flags: S   Timeout: Infinity
            Uptime: 1d 23:13:43 Time since last Join: 1d 23:13:43


user@pr5> show pim join extensive
            ge-0/3/1.0

Instance: PIM.master Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Instance: PIM.master Family: INET6
R = Rendezvous Point Tree, S = Sparse, W = Wildcard
```

Checking the PIM Sources

Purpose

Verify that the PIM sources have the expected M-LDP in-band upstream and downstream details.

Action

From operational mode, enter the `show pim source` command.

```
user@IngressPE> show pim source

Instance: PIM.master Family: INET

Source 10.1.1.1
    Prefix 10.1.1.1/32
    Upstream interface Local
    Upstream neighbor Local

Source 10.2.7.7
    Prefix 10.2.7.0/24
    Upstream protocol MLDP
```

```
    Upstream interface Pseudo MLDP
    Upstream neighbor MLDP LSP root <10.1.1.2>

Source 192.168.219.11
    Prefix 192.168.219.0/28
    Upstream protocol MLDP
    Upstream interface Pseudo MLDP
    Upstream neighbor MLDP LSP root <10.1.1.2>
```

```
user@EgressPE> show pim source
Instance: PIM.master Family: INET

Source 10.2.7.7
    Prefix 1.2.7.0/24
    Upstream interface fe-1/2/3.0
    Upstream neighbor 10.2.7.2

Source 10.2.7.7
    Prefix 10.2.7.0/24
    Upstream interface fe-1/2/3.0
    Upstream neighbor Direct

Source 192.168.219.11
    Prefix 192.168.219.0/28
    Upstream interface fe-1/3/1.0
    Upstream neighbor 192.168.219.9

Source 192.168.219.11
    Prefix 192.168.219.0/28
    Upstream interface fe-1/3/1.0
    Upstream neighbor Direct
```

```
user@pr3> show pim source

Instance: PIM.master Family: INET

Source 10.2.7.7
    Prefix 1.2.7.0/24
    Upstream protocol MLDP
    Upstream interface Pseudo MLDP
```

```
    Upstream neighbor MLDP LSP root <10.1.1.2>

Source 192.168.219.11
    Prefix 192.168.219.0/28
    Upstream protocol MLDP
    Upstream interface Pseudo MLDP
    Upstream neighbor MLDP LSP root <10.1.1.2>
```

```
user@pr4> show pim source
Instance: PIM.master Family: INET

Source 10.1.1.4
    Prefix 10.1.1.4/32
    Upstream interface Local
    Upstream neighbor Local

Source 192.168.219.11
    Prefix 192.168.219.0/28
    Upstream interface fe-1/2/0.0
    Upstream neighbor 10.1.4.1
```

Checking the LDP Database

Purpose

Make sure that the `show ldp database` command displays the expected root-to-(S,G) bindings.

Action

```
user@IngressPE> show ldp database
Input label database, 10.255.2.227:0--10.1.1.3:0
  Label     Prefix
 300096      10.1.1.2/32
      3      10.1.1.3/32
 299856      10.1.1.6/32
 299776      10.255.2.227/32
```

```
Output label database, 10.255.2.227:0--10.1.1.3:0
  Label     Prefix
 300144      10.1.1.2/32
 299776      10.1.1.3/32
 299856      10.1.1.6/32
      3      10.255.2.227/32

Input label database, 10.255.2.227:0--10.1.1.6:0
  Label     Prefix
 299936      10.1.1.2/32
 299792      10.1.1.3/32
      3      10.1.1.6/32
 299776      10.255.2.227/32

Output label database, 10.255.2.227:0--10.1.1.6:0
  Label     Prefix
 300144      10.1.1.2/32
 299776      10.1.1.3/32
 299856      10.1.1.6/32
      3      10.255.2.227/32
 300432      P2MP root-addr 10.1.1.2, grp: 232.2.2.2, src: 10.2.7.7
 300288      P2MP root-addr 10.1.1.2, grp: 232.1.1.1, src: 192.168.219.11
 300160      P2MP root-addr 10.1.1.2, grp: 232.1.1.2, src: 192.168.219.11
 300480      P2MP root-addr 10.1.1.2, grp: 232.1.1.3, src: 192.168.219.11
```

```
user@EgressPE> show ldp database

Input label database, 10.1.1.2:0--10.1.1.3:0
  Label     Prefix
 300096      10.1.1.2/32
      3      10.1.1.3/32
 299856      10.1.1.6/32
 299776      10.255.2.227/32
 300144      P2MP root-addr 10.1.1.2, grp: 232.2.2.2, src: 10.2.7.7
 300128      P2MP root-addr 10.1.1.2, grp: 232.1.1.2, src: 192.168.219.11

Output label database, 10.1.1.2:0--10.1.1.3:0
  Label     Prefix
      3      10.1.1.2/32
 299776      10.1.1.3/32
 299808      10.1.1.6/32
```

```
 299792      10.255.2.227/32

Input label database, 10.1.1.2:0--10.1.1.6:0
  Label     Prefix
 299936      10.1.1.2/32
 299792      10.1.1.3/32
      3      10.1.1.6/32
 299776      10.255.2.227/32
 300128      P2MP root-addr 10.1.1.2, grp: 232.2.2.2, src: 10.2.7.7
 299984      P2MP root-addr 10.1.1.2, grp: 232.1.1.1, src: 192.168.219.11
 299952      P2MP root-addr 10.1.1.2, grp: 232.1.1.2, src: 192.168.219.11
 300176      P2MP root-addr 10.1.1.2, grp: 232.1.1.3, src: 192.168.219.11
 300192      P2MP root-addr 10.1.1.2, grp: ff3e::1:2, src: 2001:db8:abcd::10:2:7:7

Output label database, 10.1.1.2:0--10.1.1.6:0
  Label     Prefix
      3      10.1.1.2/32
 299776      10.1.1.3/32
 299808      10.1.1.6/32
 299792      10.255.2.227/32
-----

logical-system: default

Input label database, 10.255.2.227:0--10.1.1.3:0
  Label     Prefix
 300096      10.1.1.2/32
      3      10.1.1.3/32
 299856      10.1.1.6/32
 299776      10.255.2.227/32

Output label database, 10.255.2.227:0--10.1.1.3:0
  Label     Prefix
 300144      10.1.1.2/32
 299776      10.1.1.3/32
 299856      10.1.1.6/32
      3      10.255.2.227/32

Input label database, 10.255.2.227:0--10.1.1.6:0
  Label     Prefix
 299936      10.1.1.2/32
 299792      10.1.1.3/32
      3      10.1.1.6/32
```

```
 299776      10.255.2.227/32

Output label database, 10.255.2.227:0--10.1.1.6:0
  Label     Prefix
 300144      10.1.1.2/32
 299776      10.1.1.3/32
 299856      10.1.1.6/32
      3      10.255.2.227/32
 300432      P2MP root-addr 10.1.1.2, grp: 232.2.2.2, src: 10.2.7.7
 300288      P2MP root-addr 10.1.1.2, grp: 232.1.1.1, src: 192.168.219.11
 300160      P2MP root-addr 10.1.1.2, grp: 232.1.1.2, src: 192.168.219.11
 300480      P2MP root-addr 10.1.1.2, grp: 232.1.1.3, src: 192.168.219.11
 300496      P2MP root-addr 10.1.1.2, grp: ff3e::1:2, src: 2001:db8:abcd::10:2:7:7
```

```
user@p6> show ldp database

Input label database, 10.1.1.6:0--10.1.1.2:0
  Label     Prefix
      3      10.1.1.2/32
 299776      10.1.1.3/32
 299808      10.1.1.6/32

Output label database, 10.1.1.6:0--10.1.1.2:0
  Label     Prefix
 299776      10.1.1.2/32
 299792      10.1.1.3/32
      3      10.1.1.6/32
```

```
user@pr3> show ldp database

Input label database, 10.1.1.3:0--10.1.1.2:0
  Label     Prefix
      3      10.1.1.2/32
 299776      10.1.1.3/32
 299808      10.1.1.6/32
 299792      10.255.2.227/32

Output label database, 10.1.1.3:0--10.1.1.2:0
  Label     Prefix
 300096      10.1.1.2/32
```

```
      3      10.1.1.3/32
 299856      10.1.1.6/32
 299776      10.255.2.227/32
 300144      P2MP root-addr 10.1.1.2, grp: 232.2.2.2, src: 10.2.7.7
 300128      P2MP root-addr 10.1.1.2, grp: 232.1.1.2, src: 192.168.219.11

Input label database, 10.1.1.3:0--10.255.2.227:0
  Label     Prefix
 300144      10.1.1.2/32
 299776      10.1.1.3/32
 299856      10.1.1.6/32
      3      10.255.2.227/32

Output label database, 10.1.1.3:0--10.255.2.227:0
  Label     Prefix
 300096      10.1.1.2/32
      3      10.1.1.3/32
 299856      10.1.1.6/32
 299776      10.255.2.227/32
```

Looking Up the Route Information for the MPLS Label

Purpose

Display the point-to-multipoint FEC information.

Action

```
user@EgressPE> show route label 299808 detail

mpls.0: 14 destinations, 14 routes (14 active, 0 holddown, 0 hidden)
299808 (1 entry, 1 announced)
        *LDP    Preference: 9
                Next hop type: Flood
                Address: 0x931922c
                Next-hop reference count: 3
                Next hop type: Router, Next hop index: 1109
                Address: 0x9318b0c
                Next-hop reference count: 2
                Next hop: via so-0/1/3.0
                Label operation: Pop
```

```
                Next hop type: Router, Next hop index: 1110
                Address: 0x93191e0
                Next-hop reference count: 2
                Next hop: 192.168.209.11 via fe-1/3/0.0
                Label operation: Pop
                State: **Active Int AckRequest>
                Local AS:    10
                Age: 13:08:15   Metric: 1
                Validation State: unverified
                Task: LDP
                Announcement bits (1): 0-KRT
                AS path: I
                FECs bound to route: P2MP root-addr 10.1.1.2, grp: 232.1.1.1, src: 192.168.219.11
```

Checking the LDP Traffic Statistics

Purpose

Monitor the data traffic statistics for the point-to-multipoint LSP.

Action

```
user@EgressPE> show ldp traffic-statistics p2mp
P2MP FEC Statistics:

FEC(root_addr:lsp_id/grp,src)     Nexthop              Packets             Bytes Shared
10.1.1.2:232.2.2.2,10.2.7.7         so-0/1/3.0                 0                 0 No
10.1.1.2:232.1.1.1,192.168.219.11  so-0/1/3.0                 0                 0 No
                                   fe-1/3/0.0                 0                 0 No
10.1.1.2:232.1.1.2,192.168.219.11  so-0/1/3.0                 0                 0 No
                                   fe-1/3/0.0                 0                 0 No
                                   lt-1/2/0.14                0                 0 No
10.1.1.2:232.1.1.3,192.168.219.11  fe-1/3/0.0                 0                 0 No
10.1.1.2:ff3e::1:2,2001:db8:abcd::1:2:7:7   fe-1/3/0.0                0                 0 No
```

Mapping Client and Server for Segment Routing to LDP Interoperability

IN THIS SECTION

- Overview of Segment Routing to LDP Interoperability | 1549
- Segment Routing to LDP Interoperability Using OSPF | 1550
- Interoperability of Segment Routing with LDP Using ISIS | 1552

Segment routing mapping server and client support enables interoperability between network islands that run LDP and segment routing (SR or SPRING). This interoperability is useful during a migration from LDP to SR. During the transition there can be islands (or domains) with devices that support either only LDP, or only segment routing. For these devices to interwork the LDP segment routing mapping server (SRMS) and segment routing mapping client (SRMC) functionality is required. You enable these server and client functions on a device in the segment routing network.

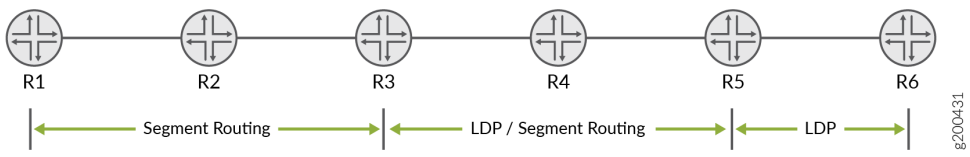
SR mapping server and client functionality is supported with either OSPF or ISIS.

Overview of Segment Routing to LDP Interoperability

Figure 102 on page 1549 shows a simple LDP network topology to illustrate how interoperability of segment routing devices with LDP devices works. Keep in mind that both OSPF and ISIS are supported, so for now we'll keep things agnostic with regard to the IGP. The sample topology has six devices, R1 through R6, in a network that is undergoing a migration from LDP to segment routing.

In the topology, devices R1, R2, and R3 are configured for segment routing only. Devices R5 and R6 are part of a legacy LDP domain and do not currently support SR. Device R4 supports both LDP and segment routing. The loopback addresses of all devices are shown. These loopbacks are advertised as egress FECs in the LDP domain and as SR node IDs in the SR domain. Interoperability is based on mapping a LDP FEC into a SR node ID, and vice versa.

Figure 102: Sample Segment Routing to LDP Interoperation Topology



For R1 to interwork with R6, both an LDP segment routing mapping server (SRMS) and a segment routing mapping client (SRMC) are needed. Its easier to understand the role of the SRMS and SRMC by looking at the traffic flow in a unidirectional manner. Based on Figure 102 on page 1549, we'll say that traffic flowing from left to right originates in the SR domain and terminates in the LDP domain. In like fashion, traffic that flows from right to left originates in the LDP domain and terminates in the SR domain.

The SRMS provides the information needed to stitch traffic in the left to right direction. The SRMC provides mapping for traffic that flows from right to left.

- Left to right Traffic Flow: The Segment Routing Mapping Server

  The SRMS facilitates LSP stitching between the SR and LDP domains. The server maps LDP FECs into SR node IDs. You configure the LDP FECs to be mapped under the `[edit routing-options source-packet-routing]` hierarchy level. Normally you need to map all LDP node loopback addresses for full connectivity. As shown below, you can map contiguous prefixes in a single range statement. If the LDP node loopbacks are not contiguous you need to define multiple mapping statements.

  You apply the SRMS mapping configuration under the `[edit protocols ospf]` or `[edit protocols isis]` hierarchy level. This choice depends on which IGP is being used. Note that both the SR and LDP nodes share a common, single area/level, IGP routing domain.

  The SRMS generates an extended prefix list LSA (or LSP in the case of ISIS). The information in this LSA allows the SR nodes to map LDP prefixes (FECs) to SR Node IDs. The mapped routes for the LDP prefixes are installed in the `inet.3` and `mpls.0` routing tables of the SR nodes to facilitate LSP ingress and stitching operations for traffic in the left to right direction.

  The extended LSA (or LSP) is flooded throughout the (single) IGP area. This means you are free to place the SRMS configuration on any router in the SR domain. The SRMS node does not have to run LDP.

- Right to Left Traffic Flow: The Segment Routing Mapping Client

  To interoperate in the right to left direction, that is, from the LDP island to the SR island, you simply enable segment routing mapping client functionality on a node that speaks both SR and LDP. In our example that is R4. You activate SRMC functionality with the `mapping-client` statement at the `[edit protocols ldp]` hierarchy.

  The SRMC configuration automatically activates an LDP egress policy to advertise the SR domain's node and prefix SIDs as LDP egress FECs. This provides the LDP nodes with LSP reachability to the nodes in the SR domain.

- The SRMC function must be configured on a router that attaches to both the SR and LSP domains. If desired, the same node can also function as the SRMS.

Segment Routing to LDP Interoperability Using OSPF

Refer to Figure 102 on page 1549, assume that device R2 (in the segment routing network) is the SRMS.

1. Define the SRMS function:

```
[edit routing-options source-packet-routing ]
user@R2# set mapping-server-entry ospf-mapping-server prefix-segment-range ldp-lo0s start-prefix 192.168.0.5
user@R2# set mapping-server-entry ospf-mapping-server prefix-segment-range ldp-lo0s start-index 1000
user@R2# set mapping-server-entry ospf-mapping-server prefix-segment-range ldp-lo0s size 2
```

This configuration creates a mapping block for both the LDP device loopback addresses in the sample topology. The initial Segment ID (SID) index mapped to R5's loopback is `1000`. Specifying size `2` results in SID index 10001 being mapped to R6's loopback address.

NOTE: The IP address used as the `start-prefix` is a loopback address of a device in the LDP network (R5, in this example). For full connectivity you must map all the loopback addresses of the LDP routers into the SR domain. If the loopback addresses are contiguous, you can do this with a single `prefix-segment-range` statement. Non-contiguous loopbacks requires definition of multiple prefix mapping statements.

Our example uses contiguous loopbacks so a single `prefix-segment-range` is shown above. Here's an example of multiple mappings to support the case of two LDP nodes with non-contiguous loopback addressing:

```
[edit routing-options source-packet-routing]
show
    mapping-server-entry map-server-name {
        prefix-segment-range lo1 {
            start-prefix 192.168.0.5/32;
            start-index 1000;
            size 1;
        }
        prefix-segment-range lo2 {
            start-prefix 192.168.0.10/32;
            start-index 2000;
            size 1;
        }
    }
}
```

2. Next, configure OSPF support for the extended LSA used to flood the mapped prefixes.

```
[edit protocols]
user@R2# set ospf source-packet-routing mapping-server ospf-mapping-server
```

   Once the mapping server configuration is committed on device R2, the extended prefix range TLV is flooded across the OSPF area. The devices capable of segment routing (R1, R2, and R3) install OSPF segment routing routes for the specified loopback address (R5 and R6 in this example), with a segment ID (SID) index. The SID index is also updated in the `mpls.0` routing table by the segment routing devices.

3. Enable SRMC functionality. For our sample topology you must enable SRMC functionality on R4.

```
[edit protocols]
user@R4# set ldp sr-mapping-client
```

   Once the mapping client configuration is committed on device R4, the SR node IDs and label blocks are advertised as egress FECs to router R5, which then re-advertises them to R6.

Support for stitching segment routing and LDP next-hops with OSPF began in Junos OS 19.1R1.

Unsupported Features and Functionality for Segment Routing interoperability with LDP using OSPF

- IPv6 prefixes are not supported.
- Flooding of the OSPF Extended Prefix Opaque LSA across AS boundaries (inter-AS) is not supported.
- Inter-area LDP mapping server functionality is not supported.
- ABR functionality of Extended Prefix Opaque LSA is not supported.
- ASBR functionality of Extended Prefix Opaque LSA is not supported.
- The segment routing mapping server Preference TLV is not supported.

Interoperability of Segment Routing with LDP Using ISIS

Refer to Figure 102 on page 1549, assume that device R2 (in the segment routing network) is the SRMS. The following configuration is added for the mapping function:

1. Define the SRMS function:

```
[edit routing-options source-packet-routing ]
user@R2# set mapping-server-entry isis-mapping-server prefix-segment-range ldp-lo0s start-prefix 192.168.0.5
user@R2# set mapping-server-entry isis-mapping-server prefix-segment-range ldp-lo0s start-index 1000
user@R2# set mapping-server-entry isis-mapping-server prefix-segment-range ldp-lo0s size 2
```

This configuration creates a mapping block for both the LDP device loopback addresses in the sample topology. The initial segment ID (SID) index mapped to R5's loopback is `1000`. Specifying size `2` results in SID index 10001 being mapped to R6's loopback address.

NOTE: The IP address used as the `start-prefix` is a loopback address of a device in the LDP network (R5, in this example). For full connectivity you must map all the loopback addresses of the LDP routers in the SR domain. If the loopback addresses are contiguous, you can do this with a `prefix-segment-range` statement. Non-contiguous loopbacks require the definition of multiple mapping statements.

Our example uses contiguous loopbacks so a single `prefix-segment-range` is shown above. Here is an example of prefix mappings to handle the case of two LDP routers with non-contiguous loopback addressing:

```
[edit routing-options source-packet-routing]
show
    mapping-server-entry map-server-name {
        prefix-segment-range lo1 {
            start-prefix 192.168.0.5/32;
            start-index 1000;
            size 1;
        }
        prefix-segment-range lo2 {
            start-prefix 192.168.0.10/32;
            start-index 2000;
            size 1;
        }
    }
}
```

2. Next, configure ISIS support for the extended LSP used to flood the mapped prefixes.

```
[edit protocols]
user@R2# set isis source-packet-routing mapping-server isis-mapping-server
```

   Once the mapping server configuration is committed on device R2, the extended prefix range TLV is flooded across the OSPF area. The devices capable of segment routing (R1, R2, and R3) install ISIS segment routing routes for the specified loopback address (R5 and R6 in this example), with a segment ID (SID) index. The SID index is also updated in the `mpls.0` routing table by the segment routing devices.

3. Enable SRMC functionality. For our sample topology you must enable SRMC functionality on R4.

```
[edit protocols]
user@R4# set ldp sr-mapping-client
```

   Once the mapping client configuration is committed on device R4, the SR node IDs and label blocks are advertised as egress FECs to router R5, and from there on to R6.

Support for stitching segment routing and LDP next-hops with ISIS began in Junos OS 17.4R1.

Unsupported Features and Functionality for Interoperability of Segment Routing with LDP using ISIS

- Penultimate-hop popping behavior for label binding TLV is not supported.
- Advertising of range of prefixes in label binding TLV is not supported.
- Segment Routing Conflict Resolution is not supported.
- LDP traffic statistics does not work.
- Nonstop active routing (NSR) and graceful Routing Engine switchover (GRES) is not supported.
- ISIS inter-level is not supported.
- RFC 7794, *IS-IS Prefix Attributes for Extended IPv4* is not supported.
- Redistributing LDP route as a prefix-sid at the stitching node is not supported.

Miscellaneous LDP Properties

IN THIS SECTION

- Configure LDP to Use the IGP Route Metric | 1555
- Prevent Addition of Ingress Routes to the inet.0 Routing Table | 1555
- Multiple-Instance LDP and Carrier-of-Carriers VPNs | 1556
- Configure MPLS and LDP to Pop the Label on the Ultimate-Hop Router | 1556
- Enable LDP over RSVP-Established LSPs | 1557
- Enable LDP over RSVP-Established LSPs in Heterogeneous Networks | 1557
- Configure the TCP MD5 Signature for LDP Sessions | 1558
- Configuring LDP Session Protection | 1560
- Disabling SNMP Traps for LDP | 1560
- Configuring LDP Synchronization with the IGP on LDP Links | 1561
- Configuring LDP Synchronization with the IGP on the Router | 1561
- Configuring the Label Withdrawal Timer | 1562
- Ignoring the LDP Subnet Check | 1562

The following sections describe how to configure a number of miscellaneous LDP properties.

Configure LDP to Use the IGP Route Metric

Use the `track-igp-metric` statement if you want the interior gateway protocol (IGP) route metric to be used for the LDP routes instead of the default LDP route metric (the default LDP route metric is 1).

To use the IGP route metric, include the `track-igp-metric` statement:

```
track-igp-metric;
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

Prevent Addition of Ingress Routes to the inet.0 Routing Table

By configuring the `no-forwarding` statement, you can prevent ingress routes from being added to the inet.0 routing table instead of the inet.3 routing table even if you enabled the `traffic-engineering bgp-igp`

statement at the [edit protocols mpls] or the [edit logical-systems *logical-system-name* protocols mpls] hierarchy level. By default, the no-forwarding statement is disabled.

NOTE: ACX Series routers do not support the [edit logical-systems] hierarchy level.

To omit ingress routes from the inet.0 routing table, include the no-forwarding statement:

```
no-forwarding;
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

Multiple-Instance LDP and Carrier-of-Carriers VPNs

By configuring multiple LDP routing instances, you can use LDP to advertise labels in a carrier-of-carriers VPN from a service provider provider edge (PE) router to a customer carrier customer edge (CE) router. This is especially useful when the carrier customer is a basic Internet service provider (ISP) and wants to restrict full Internet routes to its PE routers. By using LDP instead of BGP, the carrier customer shields its other internal routers from the Internet. Multiple-instance LDP is also useful when a carrier customer wants to provide Layer 2 or Layer 3 VPN services to its customers.

For an example of how to configure multiple LDP routing instances for carrier-of-carriers VPNs, see the *Multiple Instances for Label Distribution Protocol User Guide*.

Configure MPLS and LDP to Pop the Label on the Ultimate-Hop Router

The default advertised label is label 3 (Implicit Null label). If label 3 is advertised, the penultimate-hop router removes the label and sends the packet to the egress router. If ultimate-hop popping is enabled, label 0 (IPv4 Explicit Null label) is advertised. Ultimate-hop popping ensures that any packets traversing an MPLS network include a label.

To configure ultimate-hop popping, include the explicit-null statement:

```
explicit-null;
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

NOTE: Juniper Networks routers queue packets based on the incoming label. Routers from other vendors might queue packets differently. Keep this in mind when working with networks containing routers from multiple vendors.

For more information about labels, see "MPLS Label Overview" on page 524 and "MPLS Label Allocation" on page 524.

Enable LDP over RSVP-Established LSPs

You can run LDP over LSPs established by RSVP, effectively tunneling the LDP-established LSP through the one established by RSVP. To do so, enable LDP on the lo0.0 interface (see "Enabling and Disabling LDP" on page 1364). You must also configure the LSPs over which you want LDP to operate by including the `ldp-tunneling` statement at the `[edit protocols mpls label-switched-path` *`lsp-name`*`]` hierarchy level:

```
[edit]
protocols {
    mpls {
         label-switched-path lsp-name {
            from source;
            to destination;
            ldp-tunneling;
        }
    }
}
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

NOTE: LDP can be tunneled over a RSVP session that has link protection enabled. Starting with Junos OS Release 21.1R1, displaying details about the LDP tunneled route displays both the primary and bypass LSP next hops. In prior Junos OS releases, the bypass LSP next hop displayed the next hop for the primary LSP.

Enable LDP over RSVP-Established LSPs in Heterogeneous Networks

Some other vendors use an OSPF metric of 1 for the loopback address. Juniper Networks routers use an OSPF metric of 0 for the loopback address. This might require that you manually configure the RSVP metric when deploying LDP tunneling over RSVP LSPs in heterogeneous networks.

When a Juniper Networks router is linked to another vendor's router through an RSVP tunnel, and LDP tunneling is also enabled, by default the Juniper Networks router might not use the RSVP tunnel to route traffic to the LDP destinations downstream of the other vendor's egress router if the RSVP path has a metric of 1 larger than the physical OSPF path.

To ensure that LDP tunneling functions properly in heterogeneous networks, you can configure OSPF to ignore the RSVP LSP metric by including the `ignore-lsp-metrics` statement:

```
ignore-lsp-metrics;
```

You can configure this statement at the following hierarchy levels:

- `[edit protocols ospf traffic-engineering shortcuts]`
- `[edit logical-systems` *`logical-system-name`* `protocols ospf traffic-engineering shortcuts]`

NOTE: ACX Series routers do not support the `[edit logical-systems]` hierarchy level.

To enable LDP over RSVP LSPs, you also still need to complete the procedure in Section "Enable LDP over RSVP-Established LSPs" on page 1557.

Configure the TCP MD5 Signature for LDP Sessions

You can configure an MD5 signature for an LDP TCP connection to protect against the introduction of spoofed TCP segments into LDP session connection streams. For more information about TCP authentication, see TCP. For how to use TCP Authentication Option (TCP-AO) instead of TCP MD5, see TCP Authentication Option (TCP-AO).

A router using the MD5 signature option is configured with a password for each peer for which authentication is required. The password is stored encrypted.

LDP hello adjacencies can still be created even when peering interfaces are configured with different security signatures. However, the TCP session cannot be authenticated and is never established.

You can configure Hashed Message Authentication Code (HMAC) and MD5 authentication for LDP sessions as a per-session configuration or a subnet match (that is, longest prefix match) configuration. The support for subnet-match authentication provides flexibility in configuring authentication for automatically targeted LDP (TLDP) sessions. This makes the deployment of remote loop-free alternate (LFA) and FEC 129 pseudowires easy.

To configure an MD5 signature for an LDP TCP connection, include the `authentication-key` statement as part of the session group:

```
[edit protocols ldp]
session-group prefix-length {
     authentication-key md5-authentication-key;
}
```

Use the `session-group` statement to configure the address for the remote end of the LDP session.

The *md5-authentication-key*, or password, in the configuration can be up to 69 characters long. Characters can include any ASCII strings. If you include spaces, enclose all characters in quotation marks.

You can also configure an authentication key update mechanism for the LDP routing protocol. This mechanism allows you to update authentication keys without interrupting associated routing and signaling protocols such as Open Shortest Path First (*OSPF*) and Resource Reservation Setup Protocol (*RSVP*).

To configure the authentication key update mechanism, include the `key-chain` statement at the `[edit security authentication-key-chains]` hierarchy level, and specify the `key` option to create a keychain consisting of several authentication keys.

```
[edit security authentication-key-chains]
key-chain key-chain-name {
    key key {
        secret secret-data;
        start-time yyyy-mm-dd.hh:mm:ss;
    }
}
```

To configure the authentication key update mechanism for the LDP routing protocol, include the `authentication-key-chain` statement at the `[edit protocols ldp]` hierarchy level to associate the protocol with the `[edit security suthentication-key-chains]` authentication keys. You must also configure the authentication algorithm by including the `authentication-algorithm` *algorithm* statement the `[edit protocols ldp]` hierarchy level.

```
[edit protocols ldp]
group group-name {
    neighbor address {
        authentication-algorithm algorithm;
        authentication-key-chain key-chain-name;
```

```
    }
}
```

For more information about the authentication key update feature, see *Configuring the Authentication Key Update Mechanism for BGP and LDP Routing Protocols*.

Configuring LDP Session Protection

An LDP session is normally created between a pair of routers that are connected by one or more links. The routers form one hello adjacency for every link that connects them and associate all the adjacencies with the corresponding LDP session. When the last hello adjacency for an LDP session goes away, the LDP session is terminated. You might want to modify this behavior to prevent an LDP session from being unnecessarily terminated and reestablished.

You can configure the Junos OS to leave the LDP session between two routers up even if there are no hello adjacencies on the links connecting the two routers by configuring the `session-protection` statement. You can optionally specify a time in seconds using the `timeout` option. The session remains up for the duration specified as long as the routers maintain IP network connectivity.

```
session-protection {
    timeout seconds;
}
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section.

Disabling SNMP Traps for LDP

Whenever an LDP LSP makes a transition from up to down, or down to up, the router sends an SNMP trap. However, it is possible to disable the LDP SNMP traps on a router, logical system, or routing instance.

For information about the LDP SNMP traps and the proprietary LDP MIB, see the SNMP MIB Explorer..

To disable SNMP traps for LDP, specify the `trap disable` option for the `log-updown` statement:

```
log-updown {
    trap disable;
}
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

Configuring LDP Synchronization with the IGP on LDP Links

LDP is a protocol for distributing labels in non-traffic-engineered applications. Labels are distributed along the best path determined by the IGP. If synchronization between LDP and the IGP is not maintained, the LSP goes down. When LDP is not fully operational on a given link (a session is not established and labels are not exchanged), the IGP advertises the link with the maximum cost metric. The link is not preferred but remains in the network topology.

LDP synchronization is supported only on active point-to-point interfaces and LAN interfaces configured as point-to-point under the IGP. LDP synchronization is not supported during graceful restart.

To advertise the maximum cost metric until LDP is operational for synchronization, include the `ldp-synchronization` statement:

```
ldp-synchronization {
    disable;
    hold-time seconds;
}
```

To disable synchronization, include the `disable` statement. To configure the time period to advertise the maximum cost metric for a link that is not fully operational, include the `hold-time` statement.

For a list of hierarchy levels at which you can configure this statement, see the statement summary section for this statement.

Configuring LDP Synchronization with the IGP on the Router

You can configure the time the LDP waits before informing the IGP that the LDP neighbor and session for an interface are operational. For large networks with numerous FECs, you might need to configure a longer value to allow enough time for the LDP label databases to be exchanged.

To configure the time the LDP waits before informing the IGP that the LDP neighbor and session are operational, include the `igp-synchronization` statement and specify a time in seconds for the `holddown-interval` option:

```
igp-synchronization holddown-interval seconds;
```

For a list of hierarchy levels at which you can configure this statement, see the statement summary section for this statement.

Configuring the Label Withdrawal Timer

The label withdrawal timer delays sending a label withdrawal message for a FEC to a neighbor. When an IGP link to a neighbor fails, the label associated with the FEC has to be withdrawn from all the upstream routers if the neighbor is the next hop for the FEC. After the IGP converges and a label is received from a new next hop, the label is readvertised to all the upstream routers. This is the typical network behavior. By delaying label withdrawal by a small amount of time (for example, until the IGP converges and the router receives a new label for the FEC from the downstream next hop), the label withdrawal and sending a label mapping soon could be avoided. The `label-withdrawal-delay` statement allows you to configure this delay time. By default, the delay is 60 seconds.

If the router receives the new label before the timer runs out, the label withdrawal timer is canceled. However, if the timer runs out, the label for the FEC is withdrawn from all of the upstream routers.

By default, LDP waits for 60 seconds before withdrawing labels to avoid resignaling LSPs multiple times while the IGP is reconverging. To configure the label withdrawal delay time in seconds, include the `label-withdrawal-delay` statement:

```
label-withdrawal-delay seconds;
```

For a list of hierarchy levels at which you can configure this statement, see the statement summary section for this statement.

Ignoring the LDP Subnet Check

In Junos OS Release 8.4 and later releases, an LDP source address subnet check is performed during the neighbor establishment procedure. The source address in the LDP link hello packet is matched against the interface address. This causes an interoperability issue with some other vendors' equipment.

To disable the subnet check, include the `allow-subnet-mismatch` statement:

```
allow-subnet-mismatch;
```

This statement can be included at the following hierarchy levels:

- [edit protocols ldp `interface` *interface-name*]
- [edit logical-systems *logical-system-name* protocols ldp `interface` *interface-name*]

NOTE: ACX Series routers do not support [`edit logical-systems`] hierarchy level.

SEE ALSO

Tunneling LDP LSPs in RSVP LSPs Overview | 1324

Configuring LDP LSP Traceroute

You can trace the route followed by an LDP-signaled LSP. LDP LSP traceroute is based on RFC 4379, *Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures*. This feature allows you to periodically trace all paths in a FEC. The FEC topology information is stored in a database accessible from the CLI.

A topology change does not automatically trigger a trace of an LDP LSP. However, you can manually initiate a traceroute. If the traceroute request is for an FEC that is currently in the database, the contents of the database are updated with the results.

The periodic traceroute feature applies to all FECs specified by the `oam` statement configured at the `[edit protocols ldp]` hierarchy level. To configure periodic LDP LSP traceroute, include the `periodic-traceroute` statement:

```
periodic-traceroute {
    disable;
    exp exp-value;
    fanout fanout-value;
    frequency minutes;
    paths number-of-paths;
    retries retry-attempts;
    source address;
    ttl ttl-value;
    wait seconds;
}
```

You can configure this statement at the following hierarchy levels:

- `[edit protocols ldp oam]`
- `[edit protocols ldp oam fec address]`

You can configure the `periodic-traceroute` statement by itself or with any of the following options:

- `exp`—Specify the class of service to use when sending probes.
- `fanout`—Specify the maximum number of next hops to search per node.
- `frequency`—Specify the interval between traceroute attempts.
- `paths`—Specify the maximum number of paths to search.

- `retries`—Specify the number of attempts to send a probe to a specific node before giving up.
- `source`—Specify the IPv4 source address to use when sending probes.
- `ttl`—Specify the maximum time-to-live value. Nodes that are beyond this value are not traced.
- `wait`—Specify the wait interval before resending a probe packet.

Collecting LDP Statistics

IN THIS SECTION

- LDP Statistics Output | **1565**
- Disabling LDP Statistics on the Penultimate-Hop Router | **1565**
- LDP Statistics Limitations | **1566**

LDP traffic statistics show the volume of traffic that has passed through a particular FEC on a router.

When you configure the `traffic-statistics` statement at the `[edit protocols ldp]` hierarchy level, the LDP traffic statistics are gathered periodically and written to a file. You can configure how often statistics are collected (in seconds) by using the `interval` option. The default collection interval is 5 minutes. You must configure an LDP statistics file; otherwise, LDP traffic statistics are not gathered. If the LSP goes down, the LDP statistics are reset.

To collect LDP traffic statistics, include the `traffic-statistics` statement:

```
traffic-statistics {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    interval interval;
    no-penultimate-hop;
}
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

This section includes the following topics:

LDP Statistics Output

The following sample output is from an LDP statistics file:

```
FEC                 Type       Packets     Bytes    Shared
10.255.350.448/32   Transit          0         0        No
                    Ingress          0         0        No
10.255.350.450/32   Transit          0         0       Yes
                    Ingress          0         0        No
10.255.350.451/32   Transit          0         0        No
                    Ingress          0         0        No
220.220.220.1/32    Transit          0         0       Yes
                    Ingress          0         0        No
220.220.220.2/32    Transit          0         0       Yes
                    Ingress          0         0        No
220.220.220.3/32    Transit          0         0       Yes
                    Ingress          0         0        No
May 28 15:02:05, read 12 statistics in  00:00:00 seconds
```

The LDP statistics file includes the following columns of data:

- `FEC`—FEC for which LDP traffic statistics are collected.
- `Type`—Type of traffic originating from a router, either `Ingress` (originating from this router) or `Transit` (forwarded through this router).
- `Packets`—Number of packets passed by the FEC since its LSP came up.
- `Bytes`—Number of bytes of data passed by the FEC since its LSP came up.
- `Shared`—A `Yes` value indicates that several prefixes are bound to the same label (for example, when several prefixes are advertised with an egress policy). The LDP traffic statistics for this case apply to all the prefixes and should be treated as such.
- `read`—This number (which appears next to the date and time) might differ from the actual number of the statistics displayed. Some of the statistics are summarized before being displayed.

Disabling LDP Statistics on the Penultimate-Hop Router

Gathering LDP traffic statistics at the penultimate-hop router can consume excessive system resources, on next-hop routes in particular. This problem is exacerbated if you have configured the `deaggregate`

statement in addition to the traffic-statistics statement. For routers reaching their limit of next-hop route usage, we recommend configuring the no-penultimate-hop option for the traffic-statistics statement:

```
traffic-statistics {
    no-penultimate-hop;
}
```

For a list of hierarchy levels at which you can configure the traffic-statistics statement, see the statement summary section for this statement.

NOTE: When you configure the no-penultimate-hop option, no statistics are available for the FECs that are the penultimate hop for this router.

Whenever you include or remove this option from the configuration, the LDP sessions are taken down and then restarted.

The following sample output is from an LDP statistics file showing routers on which the no-penultimate-hop option is configured:

```
FEC                 Type           Packets          Bytes       Shared
10.255.245.218/32   Transit              0              0           No
                    Ingress              4            246           No
10.255.245.221/32   Transit     statistics disabled
                    Ingress     statistics disabled
13.1.1.0/24         Transit     statistics disabled
                    Ingress     statistics disabled
13.1.3.0/24         Transit     statistics disabled
                    Ingress     statistics disabled
```

LDP Statistics Limitations

The following are issues related to collecting LDP statistics by configuring the traffic-statistics statement:

- You cannot clear the LDP statistics.
- If you shorten the specified interval, a new LDP statistics request is issued only if the statistics timer expires later than the new interval.
- A new LDP statistics collection operation cannot start until the previous one has finished. If the interval is short or if the number of LDP statistics is large, the time gap between the two statistics collections might be longer than the interval.

When an LSP goes down, the LDP statistics are reset.

Tracing LDP Protocol Traffic

IN THIS SECTION

- Tracing LDP Protocol Traffic at the Protocol and Routing Instance Levels | 1567
- Tracing LDP Protocol Traffic Within FECs | 1568
- Examples: Tracing LDP Protocol Traffic | 1569

The following sections describe how to configure the trace options to examine LDP protocol traffic:

Tracing LDP Protocol Traffic at the Protocol and Routing Instance Levels

To trace LDP protocol traffic, you can specify options in the global `traceoptions` statement at the `[edit routing-options]` hierarchy level, and you can specify LDP-specific options by including the `traceoptions` statement:

```
traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
}
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

Use the `file` statement to specify the name of the file that receives the output of the tracing operation. All files are placed in the directory /var/log. We recommend that you place LDP-tracing output in the file **ldp-log**.

The following trace flags display the operations associated with the sending and receiving of various LDP messages. Each can carry one or more of the following modifiers:

- `address`—Trace the operation of address and address withdrawal messages.
- `binding`—Trace label-binding operations.
- `error`—Trace error conditions.
- `event`—Trace protocol events.

- initialization—Trace the operation of initialization messages.
- label—Trace the operation of label request, label map, label withdrawal, and label release messages.
- notification—Trace the operation of notification messages.
- packets—Trace the operation of address, address withdrawal, initialization, label request, label map, label withdrawal, label release, notification, and periodic messages. This modifier is equivalent to setting the address, initialization, label, notification, and periodic modifiers.

  You can also configure the filter flag modifier with the match-on address sub-option for the packets flag. This allows you to trace based on the source and destination addresses of the packets.
- path—Trace label-switched path operations.
- path—Trace label-switched path operations.
- periodic—Trace the operation of hello and keepalive messages.
- route—Trace the operation of route messages.
- state—Trace protocol state transitions.

Tracing LDP Protocol Traffic Within FECs

LDP associates a forwarding equivalence class (FEC) with each LSP it creates. The FEC associated with an LSP specifies which packets are mapped to that LSP. LSPs are extended through a network as each router chooses the label advertised by the next hop for the FEC and splices it to the label it advertises to all other routers.

You can trace LDP protocol traffic within a specific FEC and filter LDP trace statements based on an FEC. This is useful when you want to trace or troubleshoot LDP protocol traffic associated with an FEC. The following trace flags are available for this purpose: route, path, and binding.

The following example illustrates how you might configure the LDP traceoptions statement to filter LDP trace statements based on an FEC:

```
[edit protocols ldp traceoptions]
set flag route filter match-on fec policy "filter-policy-for-ldp-fec";
```

This feature has the following limitations:

- The filtering capability is only available for FECs composed of IP version 4 (IPv4) prefixes.
- Layer 2 circuit FECs cannot be filtered.

- When you configure both route tracing and filtering, MPLS routes are not displayed (they are blocked by the filter).
- Filtering is determined by the policy and the configured value for the `match-on` option. When configuring the policy, be sure that the default behavior is always `reject`.
- The only `match-on` option is `fec`. Consequently, the only type of policy you should include is a route-filter policy.

Examples: Tracing LDP Protocol Traffic

Trace LDP path messages in detail:

```
[edit]
protocols {
    ldp {
        traceoptions {
            file ldp size 10m files 5;
            flag path;
        }
    }
}
```

Trace all LDP outgoing messages:

```
[edit]
protocols {
    ldp {
        traceoptions {
            file ldp size 10m files 5;
            flag packets;
        }
    }
}
```

Trace all LDP error conditions:

```
[edit]
protocols {
    ldp {
        traceoptions {
```

```
            file ldp size 10m files 5;
            flag error;
        }
    }
}
```

Trace all LDP incoming messages and all label-binding operations:

```
[edit]
protocols {
    ldp {
        traceoptions {
            file ldp size 10m files 5 world-readable;
            flag packets receive;
            flag binding;
        }
        interface all {
        }
    }
}
```

Trace LDP protocol traffic for an FEC associated with the LSP:

```
[edit]
protocols {
    ldp {
        traceoptions {
            flag route filter match-on fec policy filter-policy-for-ldp-fec;
        }
    }
}
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
22.4R1	Starting in Junos OS Evolved Release 22.4R1, you can configure TCP-AO or TCP MD5 authentication with an IP subnet to include the entire range of addresses under that subnet.

22.4R1	Starting in Junos OS Evolved Release 22.4R1, TCP authentication is VRF aware.
16.1	Starting from Junos OS Release 16.1, M-LDP can signal point-to-multipoint LSPs at ASBR or transit or egress when root address is a BGP route which is further recursively resolved over an MPLS LSP.
14.1	Starting in Junos OS Release 14.1, in order to migrate existing IPTV services from native IP multicast to MPLS multicast, you need to smoothly transition from PIM to M-LDP point-to-multipoint LSPs with minimal outage.

Example: Configuring Multiple-Instance LDP

IN THIS SECTION

- Verifying Your Work | **1582**

The primary LDP instance is configured at the `[edit protocols]` hierarchy level.

You can configure a specific instance of LDP by using the `ldp` statement at the `[edit routing-instances` *`routing-instance-name`* `protocols]` hierarchy level. This creates an instance of LDP for the particular VRF routing instance. You must specify all the required VRF statements and apply export and import policies to your LDP instance for the configuration to commit properly.

Most of the LDP hierarchy levels available in a primary instance are also available for specific instances of LDP. However, the `no-forwarding` option does not work in a VRF-based instance of LDP.

Figure 103: Multiple-Instance LDP Topology Diagram

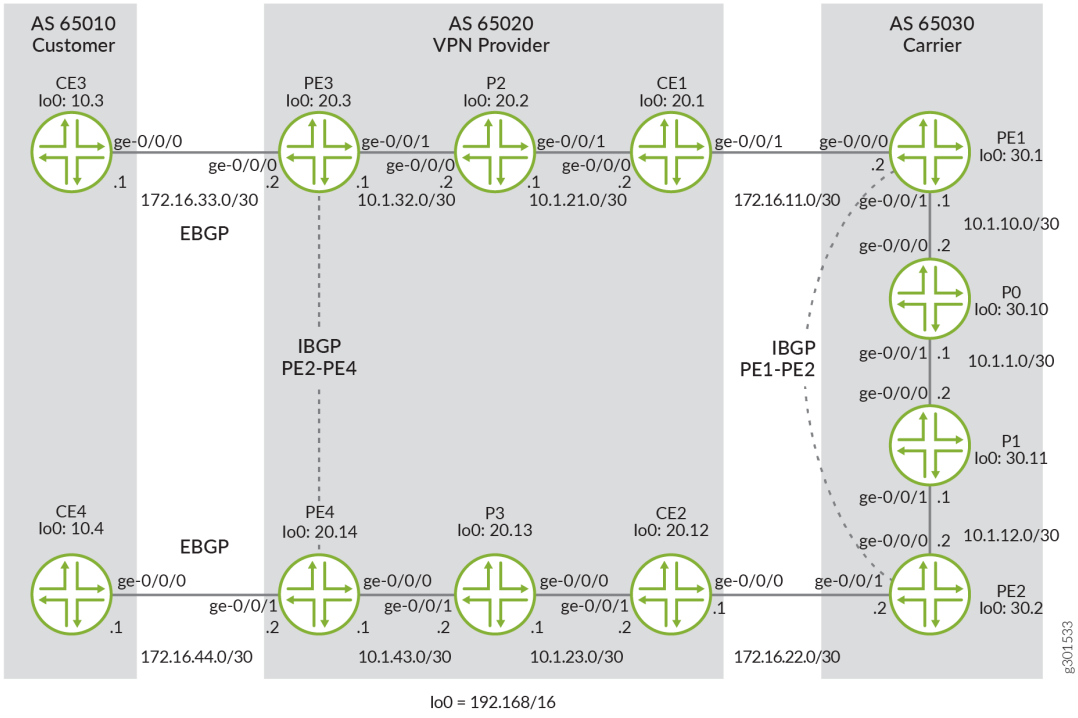


Figure 103 on page 1572 shows an example of a carrier-of-carriers network. CE3 and CE4 are end customer CE routers residing in AS 65010. The VPN provider in AS 65020 has three types of routers: PE3 and PE4 are PE routers that connect to the end customer, CE1 and CE2 act as the intermediate carrier CE routers, and P2 and P3 are internal transit routers. PE1 and PE2 in AS 65030 are PE routers servicing the intermediate VPN provider, and P0 and P1 are transit routers for the top-tier carrier.

To make this configuration work, you must complete three major tasks:

1. Configure external BGP between the VPN customer CE and the VPN provider PE.
2. Configure internal BGP using the VPN family between both pairs of PE routers (one IBGP connection between PE1 and PE2 and a second IBGP connection between Router PE3 and Router PE4).
3. Establish LDP and Interior Gateway Protocol (IGP) connections on all remaining links. This example uses OSPF as the IGP, but you can use the IGP of your choice.

Information supporting this carrier-of-carriers multiple-instance LDP example is summarized in Table 33 on page 1573.

Table 33: Multiple-Instance LDP Example—Routing Protocol Summary

Connection	Protocols
CE3 - PE3	EBGP family inet
PE3 - P2 - CE1	OSPF and LDP
CE1 - PE1	OSPF and LDP
PE1 - P0 - P1 - PE2	OSPF and LDP
PE1 - PE2	IBGP family inet-vpn
PE2 - CE2	OSPF and LDP
CE2 - P3 - PE4	OSPF and LDP
PE4 - CE4	EBGP family inet
PE3 - PE4	IBGP family inet-vpn

Your configuration tasks start at Router CE3 and move router by router through the first part of the VPN provider network, into the carrier AS, through the second VPN provider cluster of AS 65020, and end at the second VPN customer Router CE4.

Since Router CE3 is the first customer router, configure EBGP between Router CE3 and the connected VPN provider Router PE3. You must also advertise your loopback address into BGP with a routing policy to allow IP reachability with Router CE4.

Router CE3

```
user@CE3#
set interfaces ge-0/0/0 description to-PE3
set interfaces ge-0/0/0 unit 0 family inet address 172.16.33.1/30
set interfaces lo0 unit 0 family inet address 192.168.10.3/32
set policy-options policy-statement loopback term 1 from route-filter 192.168.10.3/32 exact
set policy-options policy-statement loopback term 1 then accept
```

```
set policy-options policy-statement loopback term 3 then reject
set protocols bgp group to-PE3 export loopback
set protocols bgp group to-PE3 peer-as 65020
set protocols bgp group to-PE3 neighbor 172.16.33.2
set routing-options router-id 192.168.10.3
set routing-options autonomous-system 65010
```

On Router PE3, the configuration tasks are more involved. You need to complete the EBGP connection to Router CE3 in a VRF instance, enable MPLS and LDP on the interface pointing toward the VPN provider Router CE1, and configure a primary instance of IBGP to reach Router PE4 at the far edge of AS 65020.

Finally, set up an outbound VRF policy that places all BGP traffic and directly connected interfaces into a BGP community and an inbound VRF policy that accepts similar BGP community traffic from Router PE4.

Router PE3

```
user@PE3#
set interfaces ge-0/0/0 description to-CE3
set interfaces ge-0/0/0 unit 0 family inet address 172.16.33.2/30
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 description to-P2
set interfaces ge-0/0/1 unit 0 family inet address 10.1.32.1/30
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.20.3/32
set policy-options policy-statement vpn-customer-export term 1 from protocol bgp
set policy-options policy-statement vpn-customer-export term 1 from protocol direct
set policy-options policy-statement vpn-customer-export term 1 then community add vpn-customer-
comm
set policy-options policy-statement vpn-customer-export term 1 then accept
set policy-options policy-statement vpn-customer-export term 2 then reject
set policy-options policy-statement vpn-customer-import term 1 from protocol bgp
set policy-options policy-statement vpn-customer-import term 1 from community vpn-customer-comm
set policy-options policy-statement vpn-customer-import term 1 then accept
set policy-options policy-statement vpn-customer-import term 2 then reject
set policy-options community vpn-customer-comm members target:65020:1
set routing-instances vpn-customer instance-type vrf
set routing-instances vpn-customer protocols bgp group customer peer-as 65010
set routing-instances vpn-customer protocols bgp group customer as-override
set routing-instances vpn-customer protocols bgp group customer neighbor 172.16.33.1
set routing-instances vpn-customer interface ge-0/0/0.0
set routing-instances vpn-customer route-distinguisher 192.168.20.3:1
```

```
set routing-instances vpn-customer vrf-import vpn-customer-import
set routing-instances vpn-customer vrf-export vpn-customer-export
set protocols bgp group to-PE4 type internal
set protocols bgp group to-PE4 local-address 192.168.20.3
set protocols bgp group to-PE4 neighbor 192.168.20.14 family inet-vpn unicast
set protocols ldp interface ge-0/0/1.0
set protocols mpls interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface lo0.0
set routing-options router-id 192.168.20.3
set routing-options autonomous-system 65020
```

On Router P2, enable LDP and the IGP used for transporting labels (in this case, OSPF). You will repeat these tasks on all transit core routers, both in the VPN provider network and the core carrier network.

Router P2

```
user@P2#
set interfaces ge-0/0/0 description to-PE3
set interfaces ge-0/0/0 unit 0 family inet address 10.1.32.2/30
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 description to-CE1
set interfaces ge-0/0/1 unit 0 family inet address 10.1.21.1/30
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.20.2/32
set protocols ldp interface ge-0/0/0.0
set protocols ldp interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface lo0.0
set routing-options router-id 192.168.20.2
set routing-options autonomous-system 65020
```

For Router CE1, configure LDP and OSPF in the same manner that you configured Router P2.

Router CE1

```
user@CE1#
set interfaces ge-0/0/0 description to-P2
set interfaces ge-0/0/0 unit 0 family inet address 10.1.21.2/30
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 description to-PE1
```

```
set interfaces ge-0/0/1 unit 0 family inet address 172.16.11.1/30
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.20.1/32
set protocols ldp interface ge-0/0/0.0
set protocols ldp interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface lo0.0
set routing-options router-id 192.168.20.1
set routing-options autonomous-system 65020
```

On core carrier Router PE1, configure a primary instance for OSPF, LDP, MPLS, and IBGP (with the `family inet-vpn` option) to connect the router to neighbor Router PE2. Next, implement multiple-instance LDP by establishing a secondary instance. Enable LDP and OSPF in this instance for Router PE1 to communicate with Router CE1. MPLS is not required in the secondary instance.

Finally, set up an outbound VRF policy that places all LDP traffic coming from Router CE1 into a BGP community, an export policy that sends this community traffic to Router PE2, and an inbound VRF policy that accepts similar BGP community traffic from Router PE2. This step tunnels the VPN provider's LDP traffic into the carrier's BGP session.

Router PE1

```
user@PE1#
set interfaces ge-0/0/0 description to-CE1
set interfaces ge-0/0/0 unit 0 family inet address 172.16.11.2/30
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 description to-P0
set interfaces ge-0/0/1 unit 0 family inet address 10.1.10.1/30
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.30.1/32
set policy-options policy-statement bgp-routes-export term 1 from protocol bgp
set policy-options policy-statement bgp-routes-export term 1 from community vpn-provider-comm
set policy-options policy-statement bgp-routes-export term 1 then accept
set policy-options policy-statement bgp-routes-export term 2 then reject
set policy-options policy-statement vpn-provider-export term 1 from protocol ldp
set policy-options policy-statement vpn-provider-export term 1 from protocol ospf
set policy-options policy-statement vpn-provider-export term 1 then community add vpn-provider-
comm
set policy-options policy-statement vpn-provider-export term 1 then accept
set policy-options policy-statement vpn-provider-export term 2 then reject
set policy-options policy-statement vpn-provider-import term 1 from protocol bgp
set policy-options policy-statement vpn-provider-import term 1 from community vpn-provider-comm
```

```
set policy-options policy-statement vpn-provider-import term 1 then accept
set policy-options policy-statement vpn-provider-import term 2 then reject
set policy-options community vpn-provider-comm members target:65030:1
set routing-instances vpn-provider instance-type vrf
set routing-instances vpn-provider protocols ldp egress-policy bgp-routes-export
set routing-instances vpn-provider protocols ldp interface ge-0/0/0.0
set routing-instances vpn-provider protocols mpls traffic-engineering bgp-igp
set routing-instances vpn-provider protocols mpls interface ge-0/0/0.0
set routing-instances vpn-provider protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set routing-instances vpn-provider protocols ospf export bgp-routes-export
set routing-instances vpn-provider interface ge-0/0/0.0
set routing-instances vpn-provider route-distinguisher 192.168.30.1:1
set routing-instances vpn-provider vrf-import vpn-provider-import
set routing-instances vpn-provider vrf-export vpn-provider-export
set protocols bgp group pe type internal
set protocols bgp group pe local-address 192.168.30.1
set protocols bgp group pe family inet-vpn unicast
set protocols bgp group pe neighbor 192.168.30.2
set protocols ldp interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set routing-options router-id 192.168.30.1
set routing-options autonomous-system 65030
```

On Router P0, enable LDP and OSPF in the same manner that you configured these protocols on Router P2. You will repeat these tasks on Router P1 and Router P3.

Router P0

```
user@P0#
set interfaces ge-0/0/0 description to-PE1
set interfaces ge-0/0/0 unit 0 family inet address 10.1.10.2/30
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 description to-P1
set interfaces ge-0/0/1 unit 0 family inet address 10.1.1.1/30
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.30.10/32
set protocols ldp interface ge-0/0/0.0
set protocols ldp interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

```
set routing-options router-id 192.168.30.10
set routing-options autonomous-system 65030
```

On Router P1, enable LDP and the IGP used for transporting labels (OSPF in this case).

Router P1

```
user@P1#
set interfaces ge-0/0/0 description to-P0
set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.2/30
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 description to-PE2
set interfaces ge-0/0/1 unit 0 family inet address 10.1.12.1/30
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.30.11/32
set protocols ldp interface ge-0/0/0.0
set protocols ldp interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface lo0.0
set routing-options router-id 192.168.30.11
set routing-options autonomous-system 65030
```

Core carrier Router PE2 is a mirror image of Router PE1. First, configure a primary instance for OSPF, LDP, MPLS, and IBGP (with the `family inet-vpn` option) to connect Router PE2 to neighbor Router PE1. Next, implement multiple-instance LDP by establishing a secondary instance. Enable LDP and OSPF in this instance for Router PE2 to communicate with Router CE2. MPLS is not required in the secondary instance.

Finally, set up an outbound VRF policy that places all LDP traffic coming from Router CE2 into a BGP community, an export policy that sends this community traffic to Router PE1, and an inbound VRF policy that accepts similar BGP community traffic from Router PE1. This step tunnels the VPN provider's LDP traffic into the carrier's BGP session.

Router PE2

```
user@PE2#
set interfaces ge-0/0/0 description to-P1
set interfaces ge-0/0/0 unit 0 family inet address 10.1.12.2/30
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 description to-CE2
set interfaces ge-0/0/1 unit 0 family inet address 172.16.22.2/30
```

```
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.30.2/32
set policy-options policy-statement bgp-routes-export term 1 from protocol bgp
set policy-options policy-statement bgp-routes-export term 1 from community vpn-provider-comm
set policy-options policy-statement bgp-routes-export term 1 then accept
set policy-options policy-statement bgp-routes-export term 2 then reject
set policy-options policy-statement vpn-provider-export term 1 from protocol ldp
set policy-options policy-statement vpn-provider-export term 1 from protocol ospf
set policy-options policy-statement vpn-provider-export term 1 then community add vpn-provider-
comm
set policy-options policy-statement vpn-provider-export term 1 then accept
set policy-options policy-statement vpn-provider-export term 2 then reject
set policy-options policy-statement vpn-provider-import term 1 from protocol bgp
set policy-options policy-statement vpn-provider-import term 1 from community vpn-provider-comm
set policy-options policy-statement vpn-provider-import term 1 then accept
set policy-options policy-statement vpn-provider-import term 2 then reject
set policy-options community vpn-provider-comm members target:65030:1
set routing-instances vpn-provider instance-type vrf
set routing-instances vpn-provider protocols ldp egress-policy bgp-routes-export
set routing-instances vpn-provider protocols ldp interface ge-0/0/1.0
set routing-instances vpn-provider protocols mpls traffic-engineering bgp-igp
set routing-instances vpn-provider protocols mpls interface ge-0/0/1.0
set routing-instances vpn-provider protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set routing-instances vpn-provider protocols ospf export bgp-routes-export
set routing-instances vpn-provider interface ge-0/0/1.0
set routing-instances vpn-provider route-distinguisher 192.168.30.2:1
set routing-instances vpn-provider vrf-import vpn-provider-import
set routing-instances vpn-provider vrf-export vpn-provider-export
set protocols bgp group pe type internal
set protocols bgp group pe local-address 192.168.30.2
set protocols bgp group pe family inet-vpn unicast
set protocols bgp group pe neighbor 192.168.30.1
set protocols ldp interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set routing-options router-id 192.168.30.2
set routing-options autonomous-system 65030
```

For Router CE2, configure LDP and OSPF as you did on Router CE1 and the transit P routers.

Router CE2

```
user@CE2#
set interfaces ge-0/0/0 description to-PE2
set interfaces ge-0/0/0 unit 0 family inet address 172.16.22.1/30
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 description to-P3
set interfaces ge-0/0/1 unit 0 family inet address 10.1.23.2/30
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.20.12/32
set protocols ldp interface ge-0/0/0.0
set protocols ldp interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface lo0.0
set routing-options router-id 192.168.20.12
set routing-options autonomous-system 65020
```

Since Router P3 is another core provider router, enable LDP and OSPF on all transit interfaces.

Router P3

```
user@P3#
set interfaces ge-0/0/0 description to-CE2
set interfaces ge-0/0/0 unit 0 family inet address 10.1.23.1/30
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 description to-PE4
set interfaces ge-0/0/1 unit 0 family inet address 10.1.43.2/30
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.20.13/32
set protocols ldp interface ge-0/0/0.0
set protocols ldp interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface lo0.0
set routing-options router-id 192.168.20.13
set routing-options autonomous-system 65020
```

On Router PE4, complete the IBGP connection initiated on Router PE3 to connect the edge routers in AS 65020. Also, enable LDP and MPLS on the interface pointing toward the VPN provider Router CE2 and establish an EBGP connection to Router CE4 through use of a VRF instance.

Finally, set up an outbound VRF policy that places all BGP traffic and directly connected interfaces into a BGP community and an inbound VRF policy that accepts similar BGP community traffic from Router PE3.

Router PE4

```
user@PE4#
set interfaces ge-0/0/0 description to-P3
set interfaces ge-0/0/0 unit 0 family inet address 10.1.43.1/30
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 description to-CE4
set interfaces ge-0/0/1 unit 0 family inet address 172.16.44.2/30
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.20.14/32
set policy-options policy-statement vpn-customer-export term 1 from protocol bgp
set policy-options policy-statement vpn-customer-export term 1 from protocol direct
set policy-options policy-statement vpn-customer-export term 1 then community add vpn-customer-
comm
set policy-options policy-statement vpn-customer-export term 1 then accept
set policy-options policy-statement vpn-customer-export term 2 then reject
set policy-options policy-statement vpn-customer-import term 1 from protocol bgp
set policy-options policy-statement vpn-customer-import term 1 from community vpn-customer-comm
set policy-options policy-statement vpn-customer-import term 1 then accept
set policy-options policy-statement vpn-customer-import term 2 then reject
set policy-options community vpn-customer-comm members target:65020:1
set routing-instances vpn-customer instance-type vrf
set routing-instances vpn-customer protocols bgp group customer peer-as 65010
set routing-instances vpn-customer protocols bgp group customer as-override
set routing-instances vpn-customer protocols bgp group customer neighbor 172.16.44.1
set routing-instances vpn-customer interface ge-0/0/1.0
set routing-instances vpn-customer route-distinguisher 192.168.20.14:1
set routing-instances vpn-customer vrf-import vpn-customer-import
set routing-instances vpn-customer vrf-export vpn-customer-export
set protocols bgp group int type internal
set protocols bgp group int local-address 192.168.20.14
set protocols bgp group int neighbor 192.168.20.3 family inet-vpn unicast
set protocols ldp interface ge-0/0/0.0
set protocols mpls interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface lo0.0
set routing-options router-id 192.168.20.14
set routing-options autonomous-system 65020
```

Router CE4 is the destination VPN customer router. Configure EBGP between Router CE4 and the connected VPN provider Router PE4 to complete the configuration. Remember to advertise the loopback address into BGP by using a routing policy to allow IP reachability with Router CE3.

Router CE4

```
user@CE4#
set interfaces ge-0/0/0 description to-PE4
set interfaces ge-0/0/0 unit 0 family inet address 172.16.44.1/30
set interfaces lo0 unit 0 family inet address 192.168.10.4/32
set policy-options policy-statement loopback term 1 from route-filter 192.168.10.4/32 exact
set policy-options policy-statement loopback term 1 then accept
set policy-options policy-statement loopback term 3 then reject
set protocols bgp group provider export loopback
set protocols bgp group provider peer-as 65020
set protocols bgp group provider neighbor 172.16.44.2
set routing-options router-id 192.168.10.4
set routing-options autonomous-system 65010
```

Verifying Your Work

To verify the proper operation of your multiple-instance LDP configuration, use the following commands:

- `show ldp database`
- `show ldp interface`
- `show ldp neighbor`
- `show ldp path`
- `show ldp route`
- `show ldp session`
- `show ldp statistics`

The display output for these commands is the same as in previous Junos OS Releases, except for one difference. An instance name can now be used as an argument.

If you include an instance name with these commands, you display information for the specified LDP instance. For example, the command `show ldp neighbor instance crockett` shows all the LDP neighbors for a

VRF instance named `crockett`. Conversely, `show ldp neighbor` without an instance name displays the LDP neighbors associated with the primary instance.

The following sections show the output of these commands used with the configuration example:

Router CE3 Status

```
user@CE3> show bgp summary
Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0
                      2          2          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
172.16.33.2           65020      19653      19724       0       0  6d 3:53:37 Establ
  inet.0: 2/2/2/0

user@CE3>  show route protocol bgp
inet.0: 11 destinations, 11 routes (11 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

172.16.44.0/30     *[BGP/170] 6d 00:53:43, localpref 100
                      AS path: 65020 I, validation-state: unverified
                    >  to 172.16.33.2 via ge-0/0/0.0
192.168.10.4/32    *[BGP/170] 6d 00:53:42, localpref 100
                      AS path: 65020 65020 I, validation-state: unverified
                    >  to 172.16.33.2 via ge-0/0/0.0

inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

user@CE3>  ping 192.168.10.4 source 192.168.10.3 count 2
PING 192.168.10.4 (192.168.10.4): 56 data bytes
64 bytes from 192.168.10.4: icmp_seq=0 ttl=54 time=24.744 ms
64 bytes from 192.168.10.4: icmp_seq=1 ttl=54 time=16.336 ms

--- 192.168.10.4 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 16.336/20.540/24.744/4.204 ms
```

Router PE3 Status

```
user@PE3> show bgp summary
Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 2 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l3vpn.0
                       2          2          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
172.16.33.1           65010      19783      19708       0       0  6d 4:19:07 Establ
  vpn-customer.inet.0: 1/1/1/0
192.168.20.14         65020      19299      19297       0       1  6d 1:17:05 Establ
  bgp.l3vpn.0: 2/2/2/0
  vpn-customer.inet.0: 2/2/2/0

user@PE3>  show route protocol ldp
inet.0: 20 destinations, 20 routes (20 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

224.0.0.2/32       *[LDP/9] 6d 04:55:21, metric 1
                       MultiRecv

inet.3: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.23.0/30       *[LDP/9] 6d 01:18:46, metric 1
                    >  to 10.1.32.2 via ge-0/0/1.0, Push 299968
10.1.43.0/30       *[LDP/9] 6d 01:18:46, metric 1
                    >  to 10.1.32.2 via ge-0/0/1.0, Push 300000
192.168.20.1/32    *[LDP/9] 6d 01:18:46, metric 1
                    >  to 10.1.32.2 via ge-0/0/1.0, Push 299952
192.168.20.2/32    *[LDP/9] 6d 04:22:00, metric 1
                    >  to 10.1.32.2 via ge-0/0/1.0
192.168.20.12/32   *[LDP/9] 6d 01:18:46, metric 1
                    >  to 10.1.32.2 via ge-0/0/1.0, Push 299984
192.168.20.13/32   *[LDP/9] 6d 01:18:46, metric 1
                    >  to 10.1.32.2 via ge-0/0/1.0, Push 300016
192.168.20.14/32   *[LDP/9] 6d 01:18:46, metric 1
                    >  to 10.1.32.2 via ge-0/0/1.0, Push 300032
```

```
vpn-customer.inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)

mpls.0: 13 destinations, 13 routes (13 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

299856             *[LDP/9] 6d 04:22:00, metric 1
                    >  to 10.1.32.2 via ge-0/0/1.0, Pop
299856(S=0)        *[LDP/9] 6d 04:22:00, metric 1
                    >  to 10.1.32.2 via ge-0/0/1.0, Pop
299984             *[LDP/9] 6d 01:18:46, metric 1
                    >  to 10.1.32.2 via ge-0/0/1.0, Swap 299952
300000             *[LDP/9] 6d 01:18:46, metric 1
                    >  to 10.1.32.2 via ge-0/0/1.0, Swap 299968
300016             *[LDP/9] 6d 01:18:46, metric 1
                    >  to 10.1.32.2 via ge-0/0/1.0, Swap 299984
300032             *[LDP/9] 6d 01:18:46, metric 1
                    >  to 10.1.32.2 via ge-0/0/1.0, Swap 300000
300048             *[LDP/9] 6d 01:18:46, metric 1
                    >  to 10.1.32.2 via ge-0/0/1.0, Swap 300016
300064             *[LDP/9] 6d 01:18:46, metric 1
                    >  to 10.1.32.2 via ge-0/0/1.0, Swap 300032

bgp.l3vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

vpn-customer.inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

user@PE3>  show route protocol bgp
inet.0: 20 destinations, 20 routes (20 active, 0 holddown, 0 hidden)

inet.3: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)

vpn-customer.inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

172.16.44.0/30     *[BGP/170] 6d 01:19:31, localpref 100, from 192.168.20.14
                      AS path: I, validation-state: unverified
                    >  to 10.1.32.2 via ge-0/0/1.0, Push 299968, Push 300032(top)
192.168.10.3/32    *[BGP/170] 6d 04:21:33, localpref 100
                      AS path: 65010 I, validation-state: unverified
                    >  to 172.16.33.1 via ge-0/0/0.0
192.168.10.4/32    *[BGP/170] 6d 01:19:30, localpref 100, from 192.168.20.14
```

```
                      AS path: 65010 I, validation-state: unverified
                    >  to 10.1.32.2 via ge-0/0/1.0, Push 299968, Push 300032(top)

mpls.0: 13 destinations, 13 routes (13 active, 0 holddown, 0 hidden)

bgp.l3vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.20.14:1:172.16.44.0/30
                   *[BGP/170] 6d 01:19:31, localpref 100, from 192.168.20.14
                      AS path: I, validation-state: unverified
                    >  to 10.1.32.2 via ge-0/0/1.0, Push 299968, Push 300032(top)
192.168.20.14:1:192.168.10.4/32
                   *[BGP/170] 6d 01:19:30, localpref 100, from 192.168.20.14
                      AS path: 65010 I, validation-state: unverified
                    >  to 10.1.32.2 via ge-0/0/1.0, Push 299968, Push 300032(top)

inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

vpn-customer.inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
```

Router CE1 Status

```
user@CE1> show ldp neighbor
Address                          Interface       Label space ID     Hold time
172.16.11.2                      ge-0/0/1.0      172.16.11.2:0        12
10.1.21.1                        ge-0/0/0.0      192.168.20.2:0       12

user@CE1>  show route
inet.0: 15 destinations, 15 routes (15 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.21.0/30       *[Direct/0] 6d 01:32:24
                    >  via ge-0/0/0.0
10.1.21.2/32       *[Local/0] 6d 01:32:24
                       Local via ge-0/0/0.0
10.1.23.0/30       *[OSPF/10] 6d 03:34:28, metric 3
                    >  to 172.16.11.2 via ge-0/0/1.0
10.1.32.0/30       *[OSPF/10] 6d 01:31:41, metric 2
                    >  to 10.1.21.1 via ge-0/0/0.0
10.1.43.0/30       *[OSPF/10] 6d 03:28:49, metric 4
```

```
                    >  to 172.16.11.2 via ge-0/0/1.0
172.16.11.0/30      *[Direct/0] 6d 04:57:56
                    >  via ge-0/0/1.0
172.16.11.1/32      *[Local/0] 6d 04:57:56
                       Local via ge-0/0/1.0
192.168.20.1/32     *[Direct/0] 6d 04:40:45
                    >  via lo0.0
192.168.20.2/32     *[OSPF/10] 6d 01:31:41, metric 1
                    >  to 10.1.21.1 via ge-0/0/0.0
192.168.20.3/32     *[OSPF/10] 6d 01:31:41, metric 2
                    >  to 10.1.21.1 via ge-0/0/0.0
192.168.20.12/32    *[OSPF/150] 6d 03:34:27, metric 1, tag 3489725958
                    >  to 172.16.11.2 via ge-0/0/1.0
192.168.20.13/32    *[OSPF/150] 6d 03:28:46, metric 1, tag 3489725958
                    >  to 172.16.11.2 via ge-0/0/1.0
192.168.20.14/32    *[OSPF/150] 6d 02:51:40, metric 1, tag 3489725958
                    >  to 172.16.11.2 via ge-0/0/1.0
224.0.0.2/32        *[LDP/9] 6d 04:57:56, metric 1
                       MultiRecv
224.0.0.5/32        *[OSPF/10] 6d 04:57:56, metric 1
                       MultiRecv

inet.3: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.23.0/30        *[LDP/9] 6d 01:32:49, metric 1
                    >  to 172.16.11.2 via ge-0/0/1.0, Push 300032
10.1.43.0/30        *[LDP/9] 6d 01:32:49, metric 1
                    >  to 172.16.11.2 via ge-0/0/1.0, Push 300064
192.168.20.2/32     *[LDP/9] 6d 01:31:39, metric 1
                    >  to 10.1.21.1 via ge-0/0/0.0
192.168.20.3/32     *[LDP/9] 6d 01:31:39, metric 1
                    >  to 10.1.21.1 via ge-0/0/0.0, Push 299856
192.168.20.12/32    *[LDP/9] 6d 01:32:49, metric 1
                    >  to 172.16.11.2 via ge-0/0/1.0, Push 300048
192.168.20.13/32    *[LDP/9] 6d 01:32:49, metric 1
                    >  to 172.16.11.2 via ge-0/0/1.0, Push 300080
192.168.20.14/32    *[LDP/9] 6d 01:32:49, metric 1
                    >  to 172.16.11.2 via ge-0/0/1.0, Push 300096

mpls.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
299888             *[LDP/9] 6d 01:32:49, metric 1
                    >  to 172.16.11.2 via ge-0/0/1.0, Swap 300032
299904             *[LDP/9] 6d 01:32:49, metric 1
                    >  to 172.16.11.2 via ge-0/0/1.0, Swap 300048
299920             *[LDP/9] 6d 01:32:49, metric 1
                    >  to 172.16.11.2 via ge-0/0/1.0, Swap 300064
299936             *[LDP/9] 6d 01:32:49, metric 1
                    >  to 172.16.11.2 via ge-0/0/1.0, Swap 300080
299952             *[LDP/9] 6d 01:32:49, metric 1
                    >  to 172.16.11.2 via ge-0/0/1.0, Swap 300096
299968             *[LDP/9] 6d 01:31:39, metric 1
                    >  to 10.1.21.1 via ge-0/0/0.0, Pop
299968(S=0)        *[LDP/9] 6d 01:31:39, metric 1
                    >  to 10.1.21.1 via ge-0/0/0.0, Pop
299984             *[LDP/9] 6d 01:31:39, metric 1
                    >  to 10.1.21.1 via ge-0/0/0.0, Swap 299856

inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

ff02::2/128        *[INET6/0] 1w0d 02:45:00
                       MultiRecv
```

Router PE1 Status

```
user@PE1> show ldp neighbor instance vpn-provider
Address                            Interface       Label space ID     Hold time
172.16.11.1                        ge-0/0/0.0      192.168.20.1:0       14

user@PE1>  show ldp database instance vpn-provider
Input label database, 172.16.11.2:0--192.168.20.1:0
Labels received: 8
  Label     Prefix
 299888      10.1.23.0/30
 299920      10.1.43.0/30
      3      192.168.20.1/32
 299968      192.168.20.2/32
 299984      192.168.20.3/32
 299904      192.168.20.12/32
 299936      192.168.20.13/32
 299952      192.168.20.14/32
```

```
Output label database, 172.16.11.2:0--192.168.20.1:0
Labels advertised: 8
  Label     Prefix
 300032      10.1.23.0/30
 300064      10.1.43.0/30
 299824      192.168.20.1/32
 300112      192.168.20.2/32
 300128      192.168.20.3/32
 300048      192.168.20.12/32
 300080      192.168.20.13/32
 300096      192.168.20.14/32

user@PE1>  show ldp interface instance vpn-provider
Interface          Address                         Label space ID   Nbr   Next
                                                                   count  hello
ge-0/0/0.0         172.16.11.2                     172.16.11.2:0     1      2

user@PE1>  show ldp path instance vpn-provider
Output Session (label)          Input Session (label)
192.168.20.1:0(299824)(                           ) 192.168.20.1:0(3)( )
  Attached route:  192.168.20.1/32, Ingress route
192.168.20.1:0(300032)          ( )
  Attached route:  10.1.23.0/30
192.168.20.1:0(300048)          ( )
  Attached route:  192.168.20.12/32
192.168.20.1:0(300064)          ( )
  Attached route:  10.1.43.0/30
192.168.20.1:0(300080)          ( )
  Attached route:  192.168.20.13/32
192.168.20.1:0(300096)          ( )
  Attached route:  192.168.20.14/32
192.168.20.1:0(300112)          192.168.20.1:0(299968)
  Attached route:  192.168.20.2/32, Ingress route
192.168.20.1:0(300128)          192.168.20.1:0(299984)
  Attached route:  192.168.20.3/32, Ingress route

user@PE1>  show ldp route instance vpn-provider
Destination                          Next-hop intf/lsp/table  Next-hop address
 10.1.21.0/30                        ge-0/0/0.0               172.16.11.1
 10.1.23.0/30                                                 192.168.30.2
 10.1.32.0/30                        ge-0/0/0.0               172.16.11.1
 10.1.43.0/30                                                 192.168.30.2
```

```
 172.16.11.0/30                     ge-0/0/0.0
 172.16.11.2/32
 192.168.20.1/32                    ge-0/0/0.0              172.16.11.1
 192.168.20.2/32                    ge-0/0/0.0              172.16.11.1
 192.168.20.3/32                    ge-0/0/0.0              172.16.11.1
 192.168.20.12/32                                           192.168.30.2
 192.168.20.13/32                                           192.168.30.2
 192.168.20.14/32                                           192.168.30.2
 224.0.0.5/32

user@PE1>  show ldp session instance vpn-provider
  Address                          State       Connection  Hold time  Adv. Mode
192.168.20.1                       Operational Open          21         DU

user@PE1>  show bgp summary
Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l3vpn.0
                      5          5          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
192.168.30.2          65030      19795      19797       0       0  6d 4:21:31 Establ
  bgp.l3vpn.0: 5/5/5/0
  vpn-provider.inet.0: 5/5/5/0
```

Router PE2 Status

```
user@PE2> show ldp neighbor instance vpn-provider
Address                            Interface       Label space ID     Hold time
172.16.22.1                        ge-0/0/1.0      192.168.20.12:0      12

user@PE2>  show ldp database instance vpn-provider
Input label database, 172.16.22.2:0--192.168.20.12:0
Labels received: 8
  Label     Prefix
 299888      10.1.21.0/30
 299904      10.1.32.0/30
 299808      192.168.20.1/32
 299920      192.168.20.2/32
```

```
 299936      192.168.20.3/32
      3      192.168.20.12/32
 299856      192.168.20.13/32
 299872      192.168.20.14/32

Output label database, 172.16.22.2:0--192.168.20.12:0
Labels advertised: 8
  Label     Prefix
 300000      10.1.21.0/30
 300032      10.1.32.0/30
 299888      192.168.20.1/32
 300016      192.168.20.2/32
 300048      192.168.20.3/32
 299920      192.168.20.12/32
 299952      192.168.20.13/32
 299984      192.168.20.14/32

user@PE2>  show ldp interface instance vpn-provider
Interface          Address                         Label space ID   Nbr   Next
                                                                    count  hello
ge-0/0/1.0         172.16.22.2                     172.16.22.2:0     1      4

user@PE2>  show ldp path instance vpn-provider
Output Session (label)          Input Session (label)
192.168.20.12:0(299888)(                           )( )
                                ( )
  Attached route:  192.168.20.1/32
192.168.20.12:0(299920)         192.168.20.12:0(3)
  Attached route:  192.168.20.12/32, Ingress route
192.168.20.12:0(299952)         192.168.20.12:0(299856)
  Attached route:  192.168.20.13/32, Ingress route
192.168.20.12:0(299984)         192.168.20.12:0(299872)
  Attached route:  192.168.20.14/32, Ingress route
192.168.20.12:0(300000)         ( )
  Attached route:  10.1.21.0/30
192.168.20.12:0(300016)         ( )
  Attached route:  192.168.20.2/32
192.168.20.12:0(300032)         ( )
  Attached route:  10.1.32.0/30
192.168.20.12:0(300048)         ( )
  Attached route:  192.168.20.3/32

user@PE2>  show ldp route instance vpn-provider
```

```
Destination                          Next-hop intf/lsp/table  Next-hop address
 10.1.21.0/30                                                 192.168.30.1
 10.1.23.0/30                        ge-0/0/1.0               172.16.22.1
 10.1.32.0/30                                                 192.168.30.1
 10.1.43.0/30                        ge-0/0/1.0               172.16.22.1
 172.16.22.0/30                      ge-0/0/1.0
 172.16.22.2/32
 192.168.20.1/32                                              192.168.30.1
 192.168.20.2/32                                              192.168.30.1
 192.168.20.3/32                                              192.168.30.1
 192.168.20.12/32                    ge-0/0/1.0               172.16.22.1
 192.168.20.13/32                    ge-0/0/1.0               172.16.22.1
 192.168.20.14/32                    ge-0/0/1.0               172.16.22.1
 224.0.0.5/32

user@PE2>  show ldp session instance vpn-provider
  Address                          State       Connection  Hold time  Adv. Mode
192.168.20.12                      Operational Open          21         DU

user@PE2>  show bgp summary
Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l3vpn.0
                      5          5          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
192.168.30.1          65030      35560      35557       0       0 1w4d 2:35:55 Establ
  bgp.l3vpn.0: 5/5/5/0
  vpn-provider.inet.0: 5/5/5/0
```

Router CE2 Status

```
user@CE2> show ldp neighbor
Address                          Interface       Label space ID     Hold time
172.16.22.2                      ge-0/0/0.0      172.16.22.2:0        12
10.1.23.1                        ge-0/0/1.0      192.168.20.13:0      11

user@CE2>  show route
```

```
inet.0: 16 destinations, 16 routes (16 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.21.0/30       *[OSPF/10] 1w4d 00:42:23, metric 3
                    >  to 172.16.22.2 via ge-0/0/0.0
10.1.23.0/30       *[Direct/0] 1w4d 02:44:36
                    >  via ge-0/0/1.0
10.1.23.2/32       *[Local/0] 1w4d 02:44:36
                       Local via ge-0/0/1.0
10.1.32.0/30       *[OSPF/10] 1w4d 00:41:40, metric 4
                    >  to 172.16.22.2 via ge-0/0/0.0
10.1.43.0/30       *[OSPF/10] 1w4d 02:38:50, metric 2
                    >  to 10.1.23.1 via ge-0/0/1.0
172.16.22.0/30     *[Direct/0] 1w4d 02:44:36
                    >  via ge-0/0/0.0
172.16.22.1/32     *[Local/0] 1w4d 02:44:36
                       Local via ge-0/0/0.0
192.168.0.0/16     *[Static/5] 1w5d 01:55:12
                    >  to 10.93.31.254 via fxp0.0
192.168.20.1/32    *[OSPF/150] 1w4d 02:44:26, metric 1, tag 3489725958
                    >  to 172.16.22.2 via ge-0/0/0.0
192.168.20.2/32    *[OSPF/150] 1w4d 00:41:37, metric 1, tag 3489725958
                    >  to 172.16.22.2 via ge-0/0/0.0
192.168.20.3/32    *[OSPF/150] 1w4d 00:41:37, metric 1, tag 3489725958
                    >  to 172.16.22.2 via ge-0/0/0.0
192.168.20.12/32   *[Direct/0] 1w4d 02:44:36
                    >  via lo0.0
192.168.20.13/32   *[OSPF/10] 1w4d 02:38:50, metric 1
                    >  to 10.1.23.1 via ge-0/0/1.0
192.168.20.14/32   *[OSPF/10] 1w4d 02:01:44, metric 2
                    >  to 10.1.23.1 via ge-0/0/1.0
224.0.0.2/32       *[LDP/9] 1w4d 02:44:36, metric 1
                       MultiRecv
224.0.0.5/32       *[OSPF/10] 1w4d 02:44:36, metric 1
                       MultiRecv

inet.3: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.21.0/30       *[LDP/9] 1w4d 00:42:23, metric 1
                    >  to 172.16.22.2 via ge-0/0/0.0, Push 300000
10.1.32.0/30       *[LDP/9] 1w4d 00:41:40, metric 1
                    >  to 172.16.22.2 via ge-0/0/0.0, Push 300032
```

```
192.168.20.1/32    *[LDP/9] 1w4d 02:44:26, metric 1
                    >  to 172.16.22.2 via ge-0/0/0.0, Push 299888
192.168.20.2/32    *[LDP/9] 1w4d 00:41:40, metric 1
                    >  to 172.16.22.2 via ge-0/0/0.0, Push 300016
192.168.20.3/32    *[LDP/9] 1w4d 00:41:40, metric 1
                    >  to 172.16.22.2 via ge-0/0/0.0, Push 300048
192.168.20.13/32   *[LDP/9] 1w4d 02:38:47, metric 1
                    >  to 10.1.23.1 via ge-0/0/1.0
192.168.20.14/32   *[LDP/9] 1w4d 02:01:41, metric 1
                    >  to 10.1.23.1 via ge-0/0/1.0, Push 299872

mpls.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

299808             *[LDP/9] 1w4d 02:44:26, metric 1
                    >  to 172.16.22.2 via ge-0/0/0.0, Swap 299888
299856             *[LDP/9] 1w4d 02:38:47, metric 1
                    >  to 10.1.23.1 via ge-0/0/1.0, Pop
299856(S=0)        *[LDP/9] 1w4d 02:38:47, metric 1
                    >  to 10.1.23.1 via ge-0/0/1.0, Pop
299872             *[LDP/9] 1w4d 02:01:41, metric 1
                    >  to 10.1.23.1 via ge-0/0/1.0, Swap 299872
299888             *[LDP/9] 1w4d 00:42:23, metric 1
                    >  to 172.16.22.2 via ge-0/0/0.0, Swap 300000
299904             *[LDP/9] 1w4d 00:41:40, metric 1
                    >  to 172.16.22.2 via ge-0/0/0.0, Swap 300032
299920             *[LDP/9] 1w4d 00:41:40, metric 1
                    >  to 172.16.22.2 via ge-0/0/0.0, Swap 300016
299936             *[LDP/9] 1w4d 00:41:40, metric 1
                    >  to 172.16.22.2 via ge-0/0/0.0, Swap 300048

inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

ff02::2/128        *[INET6/0] 1w5d 01:55:12
                       MultiRecv
```

Router PE4 Status

```
user@PE4> show bgp summary
Threading mode: BGP I/O
```

```
Default eBGP mode: advertise - accept, receive - accept
Groups: 2 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l3vpn.0
                       2          2          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
172.16.44.1           65010      35462      35336       0       0 1w4d 1:56:16 Establ
  vpn-customer.inet.0: 1/1/1/0
192.168.20.3          65020      35168      35168       0       1 1w4d 0:42:52 Establ
  bgp.l3vpn.0: 2/2/2/0
  vpn-customer.inet.0: 2/2/2/0

user@PE4>  show route protocol bgp
inet.0: 20 destinations, 20 routes (20 active, 0 holddown, 0 hidden)

inet.3: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)

vpn-customer.inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

172.16.33.0/30     *[BGP/170] 1w4d 00:43:33, localpref 100, from 192.168.20.3
                      AS path: I, validation-state: unverified
                    >  to 10.1.43.2 via ge-0/0/0.0, Push 300080, Push 299936(top)
192.168.10.3/32    *[BGP/170] 1w4d 00:43:32, localpref 100, from 192.168.20.3
                      AS path: 65010 I, validation-state: unverified
                    >  to 10.1.43.2 via ge-0/0/0.0, Push 300080, Push 299936(top)
192.168.10.4/32    *[BGP/170] 1w4d 01:56:57, localpref 100
                      AS path: 65010 I, validation-state: unverified
                    >  to 172.16.44.1 via ge-0/0/1.0

mpls.0: 13 destinations, 13 routes (13 active, 0 holddown, 0 hidden)

bgp.l3vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.20.3:1:172.16.33.0/30
                   *[BGP/170] 1w4d 00:43:33, localpref 100, from 192.168.20.3
                      AS path: I, validation-state: unverified
                    >  to 10.1.43.2 via ge-0/0/0.0, Push 300080, Push 299936(top)
192.168.20.3:1:192.168.10.3/32
                   *[BGP/170] 1w4d 00:43:32, localpref 100, from 192.168.20.3
                      AS path: 65010 I, validation-state: unverified
```

```
                    >  to 10.1.43.2 via ge-0/0/0.0, Push 300080, Push 299936(top)

inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

vpn-customer.inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

user@PE4>  show route protocol ldp
inet.0: 20 destinations, 20 routes (20 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

224.0.0.2/32       *[LDP/9] 1w4d 02:05:35, metric 1
                       MultiRecv

inet.3: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.21.0/30       *[LDP/9] 1w4d 00:46:06, metric 1
                    >  to 10.1.43.2 via ge-0/0/0.0, Push 299888
10.1.32.0/30       *[LDP/9] 1w4d 00:45:23, metric 1
                    >  to 10.1.43.2 via ge-0/0/0.0, Push 299904
192.168.20.1/32    *[LDP/9] 1w4d 02:05:25, metric 1
                    >  to 10.1.43.2 via ge-0/0/0.0, Push 299824
192.168.20.2/32    *[LDP/9] 1w4d 00:45:23, metric 1
                    >  to 10.1.43.2 via ge-0/0/0.0, Push 299920
192.168.20.3/32    *[LDP/9] 1w4d 00:45:23, metric 1
                    >  to 10.1.43.2 via ge-0/0/0.0, Push 299936
192.168.20.12/32   *[LDP/9] 1w4d 02:05:25, metric 1
                    >  to 10.1.43.2 via ge-0/0/0.0, Push 299776
192.168.20.13/32   *[LDP/9] 1w4d 02:05:25, metric 1
                    >  to 10.1.43.2 via ge-0/0/0.0

vpn-customer.inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)

mpls.0: 13 destinations, 13 routes (13 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

299776             *[LDP/9] 1w4d 02:05:25, metric 1
                    >  to 10.1.43.2 via ge-0/0/0.0, Pop
299776(S=0)        *[LDP/9] 1w4d 02:05:25, metric 1
                    >  to 10.1.43.2 via ge-0/0/0.0, Pop
299792             *[LDP/9] 1w4d 02:05:25, metric 1
                    >  to 10.1.43.2 via ge-0/0/0.0, Swap 299776
299840             *[LDP/9] 1w4d 02:05:25, metric 1
```

```
                    >  to 10.1.43.2 via ge-0/0/0.0, Swap 299824
299904              *[LDP/9] 1w4d 00:46:06, metric 1
                    >  to 10.1.43.2 via ge-0/0/0.0, Swap 299888
299920              *[LDP/9] 1w4d 00:45:23, metric 1
                    >  to 10.1.43.2 via ge-0/0/0.0, Swap 299904
299936              *[LDP/9] 1w4d 00:45:23, metric 1
                    >  to 10.1.43.2 via ge-0/0/0.0, Swap 299920
299952              *[LDP/9] 1w4d 00:45:23, metric 1
                    >  to 10.1.43.2 via ge-0/0/0.0, Swap 299936

bgp.l3vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

vpn-customer.inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
```

Router CE4 Status

```
user@CE4> show route protocol bgp
inet.0: 11 destinations, 11 routes (11 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

172.16.33.0/30     *[BGP/170] 1w4d 00:46:22, localpref 100
                      AS path: 65020 I, validation-state: unverified
                    >  to 172.16.44.2 via ge-0/0/0.0
192.168.10.3/32    *[BGP/170] 1w4d 00:46:21, localpref 100
                      AS path: 65020 65020 I, validation-state: unverified
                    >  to 172.16.44.2 via ge-0/0/0.0

inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

user@CE4>  show bgp summary
Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0
                      2          2          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
172.16.44.2           65020      35346      35470       0       0 1w4d 2:00:25 Establ
```

```
  inet.0: 2/2/2/0

user@CE4>  ping 192.168.10.3 source 192.168.10.4 count 2
PING 192.168.10.3 (192.168.10.3): 56 data bytes
64 bytes from 192.168.10.3: icmp_seq=0 ttl=54 time=63.857 ms
64 bytes from 192.168.10.3: icmp_seq=1 ttl=54 time=19.586 ms

--- 192.168.10.3 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 19.586/41.721/63.857/22.135 ms
```

Tunneling LDP over SR-TE

IN THIS SECTION

- Benefits of Tunneling LDP over SR-TE | **1598**
- Tunneling LDP over SR-TE Overview | **1599**

Learn about the benefits and get an overview of tunneling LDP over SR-TE.

Benefits of Tunneling LDP over SR-TE

- Enables seamless integration of LDP over SR-TE in the core network.
- Provides flexible connectivity options to accommodate multiple topologies, protocols, and domains.
- Enables interoperability between LDP and SR capable devices.
- Leverages SR-TE load sharing capabilities.
- Provides faster restoration of network connectivity using Topology Independent Loop-Free Alternate (TI-LFA) within the SR-TE domain. SR using TI-LFA routes the traffic instantly to a backup or an alternate path if the primary path fails or becomes unavailable.

Tunneling LDP over SR-TE Overview

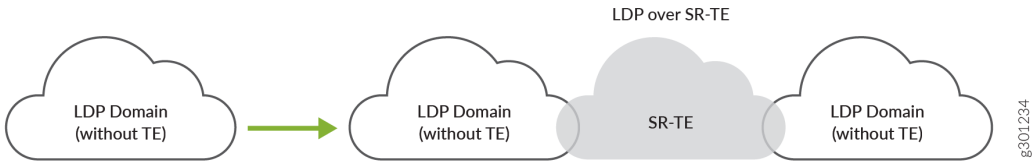
It's common for service providers to use the LDP signaling protocol with MPLS transport at the edges of their networks. LDP offers the advantage of being simple, but LDP lacks traffic engineering (TE) and sophisticated path repair capabilities that are often desired in the network's core. Many service providers are migrating from RSVP to segment routing traffic engineering (SR-TE) in the core. SR-TE is also referred to as source routing in packet networks (SPRING).

It's possible that the routers running LDP at the edge may not support SR capabilities. The service provider may wish to continue using LDP on these routers to avoid the need for an upgrade. In such scenarios, the LDP over SR-TE tunneling feature provides the ability to integrate routers that are not SR capable (running LDP) with routers that are SR capable (running SR-TE).

The LDP LSPs are tunneled through the SR-TE network, enabling interworking of LDP LSPs with SR-TE LSPs. For example, if you have LDP domains on the provider edge network and SR-TE in the core network, then you can connect the LDP domains over SR-TE, as shown in Figure 104 on page 1599.

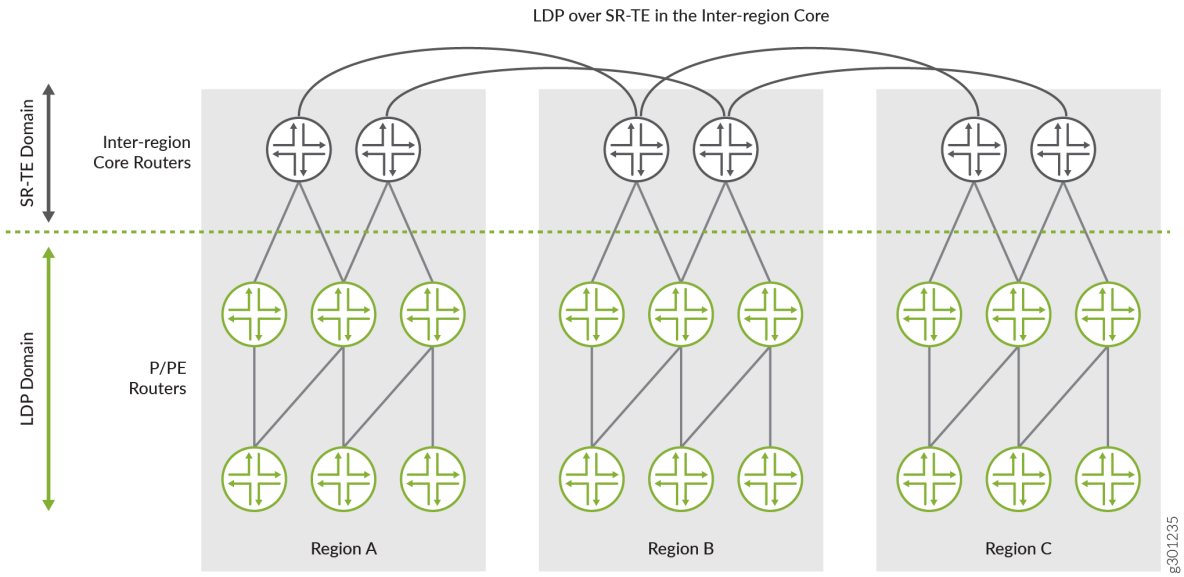
Tunneling LDP over SR-TE supports co-existence of both LDP LSPs and SR-TE LSPs.

Figure 104: Interconnect LDP Domains over SR-TE in the Core Network



You can also tunnel LDP over SR-TE between LDP domains connected to inter-region core networks. For example, if you have multiple regional LDP domains connected to the inter-region SR-TE core networks, you can tunnel LDP across the inter-region SR-TE core network, as shown in Figure 105 on page 1600.

Figure 105: LDP over SR-TE between Inter-region Core Networks



In Figure 105 on page 1600, you have three regional networks (A, B, and C) running LDP. These regional LDP domains are connected to their respective regional core networks running SR-TE. The regional SR-TE core networks are further interconnected to other regional SR-TE core networks (inter-region core network). You can tunnel LDP over these inter-region SR-TE core networks and deploy services, such as Layer 3 VPNs, seamlessly. This scenario could be used in a mobile backhaul network, where the core aggregation layer runs LDP tunneled over SR-TE while the access layer runs LDP only.

To enable LDP tunneling over SR-TE in IS-IS networks, you need to configure the following configuration statements:

- `ldp-tunneling` at the [`edit protocols source-packet-routing source-routing-path` *`source-routing-path-name`*] hierarchy level to enable LDP tunneling over SR-TE.
- `spring-te` at the [`edit protocols isis traffic-engineering tunnel-source-protocol`] hierarchy level selects LDP over SR-TE LSPs as the tunnel source protocol.

To enable LDP tunneling over SR-TE in OSPF networks, you need to configure the following configuration statements:

- `ldp-tunneling` at the [`edit protocols source-packet-routing source-routing-path` *`source-routing-path-name`*] hierarchy level to enable LDP tunneling over SR-TE.
- `spring-te` at the [`edit protocols ospf traffic-engineering tunnel-source-protocol`] hierarchy level selects LDP over SR-TE LSPs as the tunnel source protocol.

You can configure more than one tunnel source protocol for IGPs (IS-IS and OSPF) to create shortcut routes. When more than one tunnel source protocol is configured and if the tunnels from more than one

protocol are available to a destination, the tunnel with the most preferred route is established. For example, if the core network has both RSVP LSPs and SR-TE LSPs and LDP tunneling is enabled for both RSVP and SR-TE LSPs, then the `tunnel-source-protocol` configuration selects the tunnel based on the preference value. The tunnel with the lowest preference value is most preferred. You can override this route preference with a specific protocol for all destinations by configuring the preference value, as shown in the following example:

```
[edit]
user@host#set protocols isis traffic-engineering tunnel-source-protocol spring-te preference 2
user@host#set protocols isis traffic-engineering tunnel-source-protocol rsvp preference 5
```

```
[edit]
user@host#set protocols ospf traffic-engineering tunnel-source-protocol spring-te preference 2
user@host#set protocols ospf traffic-engineering tunnel-source-protocol rsvp preference 5
```

In this example, you can see the preference value configured for the SR-TE tunnel source protocol is 2 and the preference value for RSVP tunnel source protocol is 5. In this case, the SR-TE tunnel are preferred because they have the lowest preference value as compared to RSVP tunnel source protocol.

NOTE: It is not mandatory to configure the tunnel source protocol preference value. If more than one tunnel source protocol has the same preference value, then the tunnel is established based on the preferred route to the destination.

The targeted LDP session is established and is triggered when the SR-TE LSP comes up. The LSP session remains established until the LDP tunneling (`ldp-tunneling`) configuration is removed, or the SR-TE LSP is removed from the configuration.

NOTE: Junos OS currently does not support LDP over colored SR-TE LSPs.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
22.4R1	Starting in Junos OS and Junos OS Evolved Release 22.4R1, you can tunnel LDP LSPs over Segment Routing Traffic Engineering (SR-TE) in OSPF networks.

Example: Tunneling LDP over SR-TE in IS-IS Network

IN THIS SECTION

- Requirements | 1602
- Overview | 1602
- Configuration | 1603
- Verification | 1624

Use this example to learn how to tunnel LDP LSPs over SR-TE in your core network.

NOTE: Our content testing team has validated and updated this example.

Requirements

This example uses the following hardware and software components:

- MX Series routers as CE, PE, and core routers.
- Junos OS Release 20.3R1 or later running on all devices.
  - Updated and revalidated using vMX on Junos OS Release 21.1R1.

NOTE: Are you interested in getting hands-on experience on this feature?
Visit Juniper vLabs to reserve your pre-configured vLab Sandbox: Segment Routing - Basic and try it out for free!

Overview

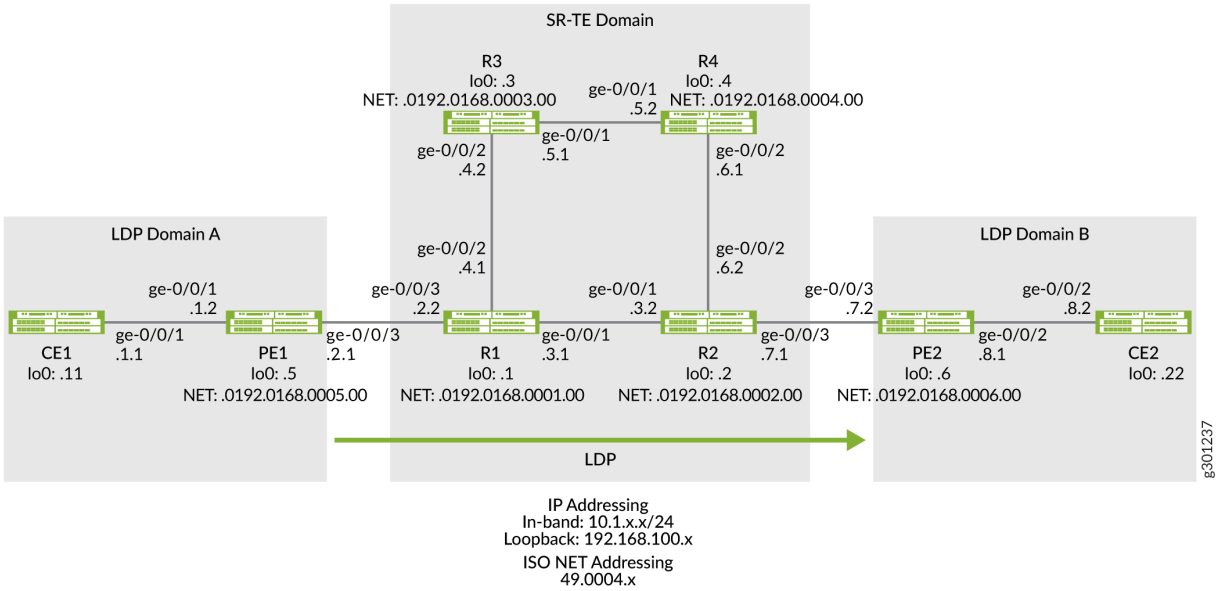
IN THIS SECTION

- Topology | 1603

The following topology (Figure 106 on page 1603) shows two LDP domains (LDP Domain A and LDP Domain B) connected to the SR-TE core network, which extends the LSP session over the core by tunneling them over SR-TE.

Topology

Figure 106: Tunneling LDP over SR-TE in the Core Network



Configuration

IN THIS SECTION

- CLI Quick Configuration | **1604**
- Configuring PE1 | **1611**
- Configuring R1 Device | **1617**

To tunnel LDP LSP over SR-TE in your core network, perform these tasks:

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

Device CE1

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/1 description CE1-to-PE1
set interfaces ge-0/0/1 unit 0 family inet address 10.1.1.1/24
set interfaces lo0 unit 0 family inet address 192.168.100.11/32
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set routing-options router-id 192.168.100.11
```

Device PE1

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/1 description PE1-to-CE1
set interfaces ge-0/0/1 unit 0 family inet address 10.1.1.2/24
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/3 description PE1-to-R1
set interfaces ge-0/0/3 unit 0 family inet address 10.1.2.1/24
set interfaces ge-0/0/3 unit 0 family iso
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.100.5/32
set interfaces lo0 unit 0 family iso address 49.0004.0192.0168.0005.00
set interfaces lo0 unit 0 family mpls
set policy-options policy-statement export_bgp term a from protocol bgp
set policy-options policy-statement export_bgp term a from protocol direct
set policy-options policy-statement export_bgp term a then accept
set routing-instances CE1_vpn1 instance-type vrf
set routing-instances CE1_vpn1 protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set routing-instances CE1_vpn1 protocols ospf export export_bgp
set routing-instances CE1_vpn1 interface ge-0/0/1.0
set routing-instances CE1_vpn1 route-distinguisher 192.168.100.5:1
set routing-instances CE1_vpn1 vrf-target target:100:4
set routing-instances CE1_vpn1 vrf-table-label
set protocols bgp group ibgp1 type internal
```

```
set protocols bgp group ibgp1 local-address 192.168.100.5
set protocols bgp group ibgp1 family inet unicast
set protocols bgp group ibgp1 family inet-vpn unicast
set protocols bgp group ibgp1 neighbor 192.168.100.6
set protocols isis interface ge-0/0/3.0 point-to-point
set protocols isis interface lo0.0 passive
set protocols ldp interface ge-0/0/3.0
set protocols ldp interface lo0.0
set protocols mpls interface ge-0/0/3.0
set protocols mpls interface lo0.0
set routing-options router-id 192.168.100.5
set routing-options autonomous-system 65410
```

Device R1

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/1 description R1-to-R2
set interfaces ge-0/0/1 unit 0 family inet address 10.1.3.1/24
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/2 description R1-to-R3
set interfaces ge-0/0/2 unit 0 family inet address 10.1.4.1/24
set interfaces ge-0/0/2 unit 0 family iso
set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/3 description R1-to-PE1
set interfaces ge-0/0/3 unit 0 family inet address 10.1.2.2/24
set interfaces ge-0/0/3 unit 0 family iso
set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
set interfaces lo0 unit 0 family inet address 192.168.100.1/32
set interfaces lo0 unit 0 family iso address 49.0004.0192.0168.0001.00
set interfaces lo0 unit 0 family mpls
set protocols isis interface ge-0/0/1.0 level 2 ipv4-adjacency-segment protected index 108
set protocols isis interface ge-0/0/1.0 level 2 ipv4-adjacency-segment unprotected index 110
set protocols isis interface ge-0/0/1.0 level 2 ipv6-adjacency-segment protected index 109
set protocols isis interface ge-0/0/1.0 level 2 ipv6-adjacency-segment unprotected index 111
set protocols isis interface ge-0/0/1.0 level 2 post-convergence-lfa
set protocols isis interface ge-0/0/1.0 point-to-point
set protocols isis interface ge-0/0/2.0 level 2 ipv4-adjacency-segment protected index 104
set protocols isis interface ge-0/0/2.0 level 2 ipv4-adjacency-segment unprotected index 106
set protocols isis interface ge-0/0/2.0 level 2 ipv6-adjacency-segment protected index 105
set protocols isis interface ge-0/0/2.0 level 2 ipv6-adjacency-segment unprotected index 107
```

```
set protocols isis interface ge-0/0/2.0 level 2 post-convergence-lfa
set protocols isis interface ge-0/0/2.0 point-to-point
set protocols isis interface ge-0/0/3.0 level 2 ipv4-adjacency-segment protected index 100
set protocols isis interface ge-0/0/3.0 level 2 ipv4-adjacency-segment unprotected index 102
set protocols isis interface ge-0/0/3.0 level 2 ipv6-adjacency-segment protected index 101
set protocols isis interface ge-0/0/3.0 level 2 ipv6-adjacency-segment unprotected index 103
set protocols isis interface ge-0/0/3.0 level 2 post-convergence-lfa
set protocols isis interface ge-0/0/3.0 point-to-point
set protocols isis interface lo0.0 passive
set protocols isis source-packet-routing srgb start-label 80000
set protocols isis source-packet-routing srgb index-range 50000
set protocols isis source-packet-routing node-segment ipv4-index 5001
set protocols isis source-packet-routing node-segment ipv6-index 5501
set protocols isis level 1 disable
set protocols isis backup-spf-options use-post-convergence-lfa
set protocols isis backup-spf-options use-source-packet-routing
set protocols isis traffic-engineering l3-unicast-topology
set protocols isis traffic-engineering credibility-protocol-preference
set protocols isis traffic-engineering tunnel-source-protocol spring-te
set protocols ldp auto-targeted-session
set protocols ldp preference 1
set protocols ldp interface ge-0/0/1.0
set protocols ldp interface ge-0/0/3.0
set protocols ldp interface lo0.0
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols mpls interface ge-0/0/3.0
set protocols mpls interface lo0.0
set protocols source-packet-routing segment-list seg1 inherit-label-nexthops
set protocols source-packet-routing segment-list seg1 auto-translate
set protocols source-packet-routing segment-list seg1 hop1 ip-address 10.1.4.2
set protocols source-packet-routing segment-list seg1 hop2 ip-address 10.1.5.2
set protocols source-packet-routing segment-list seg1 hop3 ip-address 10.1.6.2
set protocols source-packet-routing source-routing-path sr_static_r5 ldp-tunneling
set protocols source-packet-routing source-routing-path sr_static_r5 to 192.168.100.2
set protocols source-packet-routing source-routing-path sr_static_r5 binding-sid 1003001
set protocols source-packet-routing source-routing-path sr_static_r5 primary seg1
set routing-options router-id 192.168.100.1
```

Device R2

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/1 description R2-to-R1
set interfaces ge-0/0/1 unit 0 family inet address 10.1.3.2/24
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/2 description R2-to-R4
set interfaces ge-0/0/2 unit 0 family inet address 10.1.6.2/24
set interfaces ge-0/0/2 unit 0 family iso
set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/3 description R2-to-PE2
set interfaces ge-0/0/3 unit 0 family inet address 10.1.7.1/24
set interfaces ge-0/0/3 unit 0 family iso
set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
set interfaces lo0 unit 0 family inet address 192.168.100.2/32
set interfaces lo0 unit 0 family iso address 49.0004.0192.0168.0002.00
set interfaces lo0 unit 0 family mpls
set protocols isis interface ge-0/0/1.0 level 2 ipv4-adjacency-segment protected index 500
set protocols isis interface ge-0/0/1.0 level 2 ipv4-adjacency-segment unprotected index 502
set protocols isis interface ge-0/0/1.0 level 2 ipv6-adjacency-segment protected index 501
set protocols isis interface ge-0/0/1.0 level 2 ipv6-adjacency-segment unprotected index 503
set protocols isis interface ge-0/0/1.0 level 2 post-convergence-lfa
set protocols isis interface ge-0/0/1.0 point-to-point
set protocols isis interface ge-0/0/2.0 level 2 ipv4-adjacency-segment protected index 504
set protocols isis interface ge-0/0/2.0 level 2 ipv4-adjacency-segment unprotected index 506
set protocols isis interface ge-0/0/2.0 level 2 ipv6-adjacency-segment protected index 505
set protocols isis interface ge-0/0/2.0 level 2 ipv6-adjacency-segment unprotected index 507
set protocols isis interface ge-0/0/2.0 level 2 post-convergence-lfa
set protocols isis interface ge-0/0/2.0 point-to-point
set protocols isis interface ge-0/0/3.0 level 2 ipv4-adjacency-segment protected index 508
set protocols isis interface ge-0/0/3.0 level 2 ipv4-adjacency-segment unprotected index 510
set protocols isis interface ge-0/0/3.0 level 2 ipv6-adjacency-segment protected index 509
set protocols isis interface ge-0/0/3.0 level 2 ipv6-adjacency-segment unprotected index 511
set protocols isis interface ge-0/0/3.0 level 2 post-convergence-lfa
set protocols isis interface ge-0/0/3.0 point-to-point
set protocols isis interface lo0.0 passive
set protocols isis source-packet-routing srgb start-label 80000
set protocols isis source-packet-routing srgb index-range 50000
set protocols isis source-packet-routing node-segment ipv4-index 5005
set protocols isis source-packet-routing node-segment ipv6-index 5505
```

```
set protocols isis source-packet-routing traffic-statistics statistics-granularity per-interface
set protocols isis level 1 disable
set protocols isis backup-spf-options use-post-convergence-lfa
set protocols isis backup-spf-options use-source-packet-routing
set protocols isis traffic-engineering l3-unicast-topology
set protocols isis traffic-engineering credibility-protocol-preference
set protocols isis traffic-engineering tunnel-source-protocol spring-te
set protocols ldp interface ge-0/0/1.0
set protocols ldp interface ge-0/0/3.0
set protocols ldp interface lo0.0
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols mpls interface ge-0/0/3.0
set protocols mpls interface lo0.0
set protocols source-packet-routing segment-list seg1 inherit-label-nexthops
set protocols source-packet-routing segment-list seg1 auto-translate
set protocols source-packet-routing segment-list seg1 hop1 ip-address 10.1.6.1
set protocols source-packet-routing segment-list seg1 hop2 ip-address 10.1.5.1
set protocols source-packet-routing segment-list seg1 hop3 ip-address 10.1.4.1
set protocols source-packet-routing source-routing-path sr_static_r1 ldp-tunneling
set protocols source-packet-routing source-routing-path sr_static_r1 to 192.168.100.1
set protocols source-packet-routing source-routing-path sr_static_r1 binding-sid 1003001
set protocols source-packet-routing source-routing-path sr_static_r1 primary seg1
set routing-options router-id 192.168.100.2
```

Device R3

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/1 description R3-to-R4
set interfaces ge-0/0/1 unit 0 family inet address 10.1.5.1/24
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 description R3-to-R1
set interfaces ge-0/0/2 unit 0 family inet address 10.1.4.2/24
set interfaces ge-0/0/2 unit 0 family iso
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.100.3/32
set interfaces lo0 unit 0 family iso address 49.0004.0192.0168.0003.00
set interfaces lo0 unit 0 family mpls
set protocols isis interface ge-0/0/1.0 level 2 ipv4-adjacency-segment protected index 204
set protocols isis interface ge-0/0/1.0 level 2 ipv4-adjacency-segment unprotected index 206
```

```
set protocols isis interface ge-0/0/1.0 level 2 ipv6-adjacency-segment protected index 205
set protocols isis interface ge-0/0/1.0 level 2 ipv6-adjacency-segment unprotected index 207
set protocols isis interface ge-0/0/1.0 level 2 post-convergence-lfa
set protocols isis interface ge-0/0/1.0 point-to-point
set protocols isis interface ge-0/0/2.0 level 2 ipv4-adjacency-segment protected index 200
set protocols isis interface ge-0/0/2.0 level 2 ipv4-adjacency-segment unprotected index 202
set protocols isis interface ge-0/0/2.0 level 2 ipv6-adjacency-segment protected index 201
set protocols isis interface ge-0/0/2.0 level 2 ipv6-adjacency-segment unprotected index 203
set protocols isis interface ge-0/0/2.0 level 2 post-convergence-lfa
set protocols isis interface ge-0/0/2.0 point-to-point
set protocols isis interface lo0.0 passive
set protocols isis source-packet-routing srgb start-label 80000
set protocols isis source-packet-routing srgb index-range 50000
set protocols isis source-packet-routing node-segment ipv4-index 5003
set protocols isis source-packet-routing node-segment ipv6-index 5503
set protocols isis level 1 disable
set protocols isis backup-spf-options use-post-convergence-lfa
set protocols isis backup-spf-options use-source-packet-routing
set protocols isis traffic-engineering l3-unicast-topology
set protocols isis traffic-engineering credibility-protocol-preference
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols mpls interface lo0.0
set routing-options router-id 192.168.100.3
```

Device R4

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/1 description R4-to-R3
set interfaces ge-0/0/1 unit 0 family inet address 10.1.5.2/24
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 description R4-to-R2
set interfaces ge-0/0/2 unit 0 family inet address 10.1.6.1/24
set interfaces ge-0/0/2 unit 0 family iso
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.100.4/32
set interfaces lo0 unit 0 family iso address 49.0004.0192.0168.0004.00
set interfaces lo0 unit 0 family mpls
set protocols isis interface ge-0/0/1.0 level 2 ipv4-adjacency-segment protected index 300
set protocols isis interface ge-0/0/1.0 level 2 ipv4-adjacency-segment unprotected index 302
```

```
set protocols isis interface ge-0/0/1.0 level 2 ipv6-adjacency-segment protected index 301
set protocols isis interface ge-0/0/1.0 level 2 ipv6-adjacency-segment unprotected index 303
set protocols isis interface ge-0/0/1.0 level 2 post-convergence-lfa
set protocols isis interface ge-0/0/1.0 point-to-point
set protocols isis interface ge-0/0/2.0 level 2 ipv4-adjacency-segment protected index 304
set protocols isis interface ge-0/0/2.0 level 2 ipv4-adjacency-segment unprotected index 306
set protocols isis interface ge-0/0/2.0 level 2 ipv6-adjacency-segment protected index 305
set protocols isis interface ge-0/0/2.0 level 2 ipv6-adjacency-segment unprotected index 307
set protocols isis interface ge-0/0/2.0 level 2 post-convergence-lfa
set protocols isis interface ge-0/0/2.0 point-to-point
set protocols isis interface lo0.0 passive
set protocols isis source-packet-routing srgb start-label 80000
set protocols isis source-packet-routing srgb index-range 50000
set protocols isis source-packet-routing node-segment ipv4-index 5004
set protocols isis source-packet-routing node-segment ipv6-index 5504
set protocols isis source-packet-routing traffic-statistics statistics-granularity per-interface
set protocols isis level 1 disable
set protocols isis backup-spf-options use-post-convergence-lfa
set protocols isis backup-spf-options use-source-packet-routing
set protocols isis traffic-engineering l3-unicast-topology
set protocols isis traffic-engineering credibility-protocol-preference
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols mpls interface lo0.0
set routing-options router-id 192.168.100.4
```

Device PE2

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/2 description PE2-to-CE2
set interfaces ge-0/0/2 unit 0 family inet address 10.1.8.1/24
set interfaces ge-0/0/2 unit 0 family iso
set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/3 description PE2-to-R2
set interfaces ge-0/0/3 unit 0 family inet address 10.1.7.2/24
set interfaces ge-0/0/3 unit 0 family iso
set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
set interfaces lo0 unit 0 family inet address 192.168.100.6/32
set interfaces lo0 unit 0 family iso address 49.0004.0192.0168.0006.00
set interfaces lo0 unit 0 family mpls
set policy-options policy-statement export_bgp term a from protocol bgp
```

```
set policy-options policy-statement export_bgp term a from protocol direct
set policy-options policy-statement export_bgp term a then accept
set routing-instances CE2_vpn1 instance-type vrf
set routing-instances CE2_vpn1 protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set routing-instances CE2_vpn1 protocols ospf export export_bgp
set routing-instances CE2_vpn1 interface ge-0/0/2.0
set routing-instances CE2_vpn1 route-distinguisher 192.168.100.6:1
set routing-instances CE2_vpn1 vrf-target target:100:4
set routing-instances CE2_vpn1 vrf-table-label
set protocols bgp group ibgp1 type internal
set protocols bgp group ibgp1 local-address 192.168.100.6
set protocols bgp group ibgp1 family inet unicast
set protocols bgp group ibgp1 family inet-vpn unicast
set protocols bgp group ibgp1 neighbor 192.168.100.5
set protocols isis interface ge-0/0/3.0 point-to-point
set protocols isis interface lo0.0 passive
set protocols ldp interface ge-0/0/3.0
set protocols ldp interface lo0.0
set protocols mpls interface ge-0/0/3.0
set protocols mpls interface lo0.0
set routing-options router-id 192.168.100.6
set routing-options autonomous-system 65410
```

Device CE2

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/1 description CE2-to-PE2
set interfaces ge-0/0/2 unit 0 family inet address 10.1.8.2/24
set interfaces lo0 unit 0 family inet address 192.168.100.22/32
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set routing-options router-id 192.168.100.22
```

Configuring PE1

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the CLI User Guide.

To configure device PE1:

1. Configure the network services mode as Enhanced IP. Enhanced IP sets the router's network services to enhanced Internet Protocol and uses enhanced mode capabilities.

```
[edit chassis]
user@PE1#set network-services enhanced-ip
```

   After you configure the `enhanced-ip` statement and commit the configuration, the following warning message appears prompting you to reboot the router:

```
'chassis'
WARNING: Chassis configuration for network services has been changed. A system reboot is
mandatory. Please reboot the system NOW. Continuing without a reboot might result in
unexpected system behavior.
commit complete
```

   The reboot brings up the FPCs on the router.

2. Configure the device's interfaces.

```
[edit interfaces]
user@PE1#set ge-0/0/1 description PE1-to-CE1
user@PE1#set ge-0/0/1 unit 0 family inet address 10.1.1.2/24
user@PE1#set ge-0/0/1 unit 0 family iso

user@PE1#set ge-0/0/3 description PE1-to-R1
user@PE1#set ge-0/0/3 unit 0 family inet address 10.1.2.1/24
user@PE1#set ge-0/0/3 unit 0 family iso
user@PE1#set ge-0/0/3 unit 0 family mpls

user@PE1#set lo0 unit 0 family inet address 192.168.100.5/32
user@PE1#set lo0 unit 0 family iso address 49.0004.0192.0168.0005.00
user@PE1#set lo0 unit 0 family mpls
```

3. Configure policy options to export BGP routes to the CE router, which runs the OSPF protocol in this example.

```
[edit policy-options]
user@PE1#set policy-statement export_bgp term a from protocol bgp
```

```
user@PE1#set policy-statement export_bgp term a from protocol direct
user@PE1#set policy-statement export_bgp term a then accept
```

4. Configure a Layer 3 VPN routing instance to support the OSPF-based CE1 device.

```
[edit routing-instances]
user@PE1#set CE1_vpn1 instance-type vrf
user@PE1#set CE1_vpn1 protocols ospf area 0.0.0.0 interface ge-0/0/1.0
user@PE1#set CE1_vpn1 protocols ospf export export_bgp
user@PE1#set CE1_vpn1 interface ge-0/0/1.0
user@PE1#set CE1_vpn1 route-distinguisher 192.168.100.5:1
user@PE1#set CE1_vpn1 vrf-target target:100:4
user@PE1#set CE1_vpn1 vrf-table-label
```

5. Configure the router ID and autonomous system number for Device PE1.

```
[edit routing-options]
user@PE1#set router-id 192.168.100.5
userPE1# set autonomous-system 65410
```

6. Configure ISIS, LDP, and MPLS on the interfaces connected to the core network.

```
[edit protocols]
user@PE1#set isis interface ge-0/0/3.0 point-to-point
user@PE1#set isis interface lo0.0 passive

user@PE1#set ldp interface ge-0/0/3.0
user@PE1#set ldp interface lo0.0

user@PE1#set mpls interface ge-0/0/3.0
user@PE1#set mpls interface lo0.0
```

7. Configure BGP between the PE devices.

```
[edit protocols]
user@PE1#set bgp group ibgp1 type internal
user@PE1#set bgp group ibgp1 local-address 192.168.100.5
user@PE1#set bgp group ibgp1 family inet unicast
```

```
user@PE1#set bgp group ibgp1 family inet-vpn unicast
user@PE1#set bgp group ibgp1 neighbor 192.168.100.6
```

Results

From configuration mode, confirm your configuration by entering the `show chassis`, `show interfaces`, `show policy-options`, `show routing-instances`,`show routing-options`, and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@PE1#show chassis
network-services enhanced-ip;
```

```
user@PE1#show interfaces
ge-0/0/1 {
    description PE1-to-CE1;
    unit 0 {
        family inet {
            address 10.1.1.2/24;
        }
        family iso;
    }
}
ge-0/0/3 {
    description PE1-to-R1;
    unit 0 {
        family inet {
            address 10.1.2.1/24;
        }
        family iso;
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 192.168.100.5/32;
        }
```

```
        family iso {
            address 49.0004.0192.0168.0005.00;
        }
        family mpls;
    }
}
```

```
user@PE1#show policy-options
policy-statement export_bgp {
    term a {
        from protocol [ bgp direct ];
        then accept;
    }
}
```

```
user@PE1#show routing-instances
CE1_vpn1 {
    instance-type vrf;
    protocols {
        ospf {
            area 0.0.0.0 {
                interface ge-0/0/1.0;
            }
            export export_bgp;
        }
    }
    interface ge-0/0/1.0;
    route-distinguisher 192.168.100.5:1;
    vrf-target target:100:4;
    vrf-table-label;
}
```

```
user@PE1#show routing-options
```

```
router-id 192.168.100.5;
autonomous-system 65410;
```

```
user@PE1#show protocols
bgp {
    group ibgp1 {
        type internal;
        local-address 192.168.100.5;
        family inet {
            unicast;
        }
        family inet-vpn {
            unicast;
        }
        neighbor 192.168.100.6;
    }
}
isis {
    interface ge-0/0/3.0 {
        point-to-point;
    }
    interface lo0.0 {
        passive;
    }
}
ldp {
    interface ge-0/0/3.0;
    interface lo0.0;
}
mpls {
    interface ge-0/0/3.0;
    interface lo0.0;
}
```

Configuring R1 Device

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the CLI User Guide.

To configure device R1:

1. Configure the network services mode as Enhanced IP. Enhanced IP sets the router's network services to enhanced Internet Protocol and uses enhanced mode capabilities.

```
[edit chassis]
user@R1#set network-services enhanced-ip
```

After you configure the `enhanced-ip` statement and commit the configuration, the following warning message appears prompting you to reboot the router:

```
'chassis'
WARNING: Chassis configuration for network services has been changed. A system reboot is
mandatory. Please reboot the system NOW. Continuing without a reboot might result in
unexpected system behavior.
commit complete
```

The reboot brings up the FPCs on the router.

2. Configure the device's interfaces.

```
[edit interfaces]
user@R1#set ge-0/0/1 description R1-to-R2
user@R1#set ge-0/0/1 unit 0 family inet address 10.1.3.1/24
user@R1#set ge-0/0/1 unit 0 family iso
user@R1#set ge-0/0/1 unit 0 family mpls maximum-labels 8

user@R1#set ge-0/0/2 description R1-to-R3
user@R1#set ge-0/0/2 unit 0 family inet address 10.1.4.1/24
user@R1#set ge-0/0/2 unit 0 family iso
user@R1#set ge-0/0/2 unit 0 family mpls maximum-labels 8

user@R1#set ge-0/0/3 description R1-to-PE1
user@R1#set ge-0/0/3 unit 0 family inet address 10.1.2.2/24
```

```
user@R1#set ge-0/0/3 unit 0 family iso
user@R1#set ge-0/0/3 unit 0 family mpls maximum-labels 8

user@R1#set lo0 unit 0 family inet address 192.168.100.1/32
user@R1#set lo0 unit 0 family iso address 49.0004.0192.0168.0001.00
user@R1#set lo0 unit 0 family mpls
```

3. Configure routing options to identify the router in the domain.

```
[edit routing-options]
user@R1#set router-id 192.168.100.1
```

4. Configure ISIS adjacency SIDs on the interfaces and allocate SRGB labels to enable segment routing. The labels in the entire SRGB are available for ISIS. Prefix SIDs (and Node SIDs) are indexed from the SRGB.

```
[edit protocols]
user@R1#set isis interface ge-0/0/1.0 level 2 ipv4-adjacency-segment protected index 108
user@R1#set isis interface ge-0/0/1.0 level 2 ipv4-adjacency-segment unprotected index 110
user@R1#set isis interface ge-0/0/1.0 level 2 ipv6-adjacency-segment protected index 109
user@R1#set isis interface ge-0/0/1.0 level 2 ipv6-adjacency-segment unprotected index 111
user@R1#set isis interface ge-0/0/1.0 level 2 post-convergence-lfa
user@R1#set isis interface ge-0/0/1.0 point-to-point

user@R1#set isis interface ge-0/0/2.0 level 2 ipv4-adjacency-segment protected index 104
user@R1#set isis interface ge-0/0/2.0 level 2 ipv4-adjacency-segment unprotected index 106
user@R1#set isis interface ge-0/0/2.0 level 2 ipv6-adjacency-segment protected index 105
user@R1#set isis interface ge-0/0/2.0 level 2 ipv6-adjacency-segment unprotected index 107
user@R1#set isis interface ge-0/0/2.0 level 2 post-convergence-lfa
user@R1#set isis interface ge-0/0/2.0 point-to-point

user@R1#set isis interface ge-0/0/3.0 level 2 ipv4-adjacency-segment protected index 100
user@R1#set isis interface ge-0/0/3.0 level 2 ipv4-adjacency-segment unprotected index 102
user@R1#set isis interface ge-0/0/3.0 level 2 ipv6-adjacency-segment protected index 101
user@R1#set isis interface ge-0/0/3.0 level 2 ipv6-adjacency-segment unprotected index 103
user@R1#set isis interface ge-0/0/3.0 level 2 post-convergence-lfa
user@R1#set isis interface ge-0/0/3.0 point-to-point

user@R1#set isis interface lo0.0 passive

user@R1#set isis source-packet-routing srgb start-label 80000
```

```
user@R1#set isis source-packet-routing srgb index-range 50000
user@R1#set isis source-packet-routing node-segment ipv4-index 5001
user@R1#set isis source-packet-routing node-segment ipv6-index 5501
user@R1#set isis level 1 disable
```

5. Configure TI-LFA to enable protection against link and node failures. SR using TI-LFA provides faster restoration of network connectivity by routing the traffic instantly to a backup or an alternate path if the primary path fails or becomes unavailable.

```
[edit protocols]
user@R1#set isis backup-spf-options use-post-convergence-lfa
user@R1#set isis backup-spf-options use-source-packet-routing
```

6. Configure ISIS traffic engineering parameters.

```
[edit protocols]
user@R1#set isis traffic-engineering l3-unicast-topology
user@R1#set isis traffic-engineering credibility-protocol-preference
```

7. Enable LDP tunneling over SR-TE.

```
[edit protocols]
user@R1#set isis traffic-engineering tunnel-source-protocol spring-te
```

8. Configure MPLS and LDP protocols on the interfaces in the LDP domain to exchange labels in the LDP domain.

```
[edit protocols]
user@R1#set ldp preference 1
user@R1#set ldp interface ge-0/0/3.0
user@R1#set ldp interface lo0.0

user@R1#set mpls interface ge-0/0/1.0
user@R1#set mpls interface ge-0/0/2.0
user@R1#set mpls interface ge-0/0/3.0
user@R1#set mpls interface lo0.0
```

9. Enable targeted LDP session between the edge routers in the LDP domain.

```
[edit protocols]
user@R1#set ldp auto-targeted-session
```

10. Configure a segment list to route the traffic to a specific path.

```
[edit protocols]
user@R1#set source-packet-routing segment-list seg1 inherit-label-nexthops
user@R1#set source-packet-routing segment-list seg1 auto-translate
user@R1#set source-packet-routing segment-list seg1 hop1 ip-address 192.168.4.2
user@R1#set source-packet-routing segment-list seg1 hop2 ip-address 192.168.5.2
user@R1#set source-packet-routing segment-list seg1 hop3 ip-address 192.168.6.2
```

11. Configure SR-TE LSP to the remote edge routers to enable LDP tunneling over SR-TE.

```
[edit protocols]
user@R1#set source-packet-routing source-routing-path sr_static_r5 ldp-tunneling
user@R1#set source-packet-routing source-routing-path sr_static_r5 to 192.168.66.66
user@R1#set source-packet-routing source-routing-path sr_static_r5 binding-sid 1003001
user@R1#set source-packet-routing source-routing-path sr_static_r5 primary seg1
```

Results

From configuration mode, confirm your configuration by entering the `show chassis`, `show interfaces`, `show routing-options`, and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R1#show chassis
network-services enhanced-ip;
```

```
user@R1#show interfaces
ge-0/0/1 {
    description R1-to-R2;
    unit 0 {
        family inet {
            address 10.1.3.1/24;
```

```
        }
        family iso;
        family mpls {
            maximum-labels 8;
        }
    }
}
ge-0/0/2 {
    description R1-to-R3;
    unit 0 {
        family inet {
            address 10.1.4.1/24;
        }
        family iso;
        family mpls {
            maximum-labels 8;
        }
    }
}
ge-0/0/3 {
    description R1-to-PE1;
    unit 0 {
        family inet {
            address 10.1.2.2/24;
        }
        family iso;
        family mpls {
            maximum-labels 8;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 192.168.100.1/32;
        }
        family iso {
            address 49.0004.0192.0168.0001.00;
        }
        family mpls;
```

```
    }
}
```

```
user@R1#show protocols
isis {
    interface ge-0/0/1.0 {
        level 2 {
            ipv4-adjacency-segment {
                protected index 108;
                unprotected index 110;
            }
            ipv6-adjacency-segment {
                protected index 109;
                unprotected index 111;
            }
            post-convergence-lfa;
        }
        point-to-point;
    }
    interface ge-0/0/2.0 {
        level 2 {
            ipv4-adjacency-segment {
                protected index 104;
                unprotected index 106;
            }
            ipv6-adjacency-segment {
                protected index 105;
                unprotected index 107;
            }
            post-convergence-lfa;
        }
        point-to-point;
    }
    interface ge-0/0/3.0 {
        level 2 {
            ipv4-adjacency-segment {
                protected index 100;
                unprotected index 102;
            }
            ipv6-adjacency-segment {
                protected index 101;
```

```
                unprotected index 103;
            }
            post-convergence-lfa;
        }
        point-to-point;
    }
    interface lo0.0 {
        passive;
    }
    source-packet-routing {
        srgb start-label 80000 index-range 50000;
        node-segment {
            ipv4-index 5001;
            ipv6-index 5501;
        }
    }
    level 1 disable;
    backup-spf-options {
        use-post-convergence-lfa;
        use-source-packet-routing;
    }
    traffic-engineering {
        l3-unicast-topology;
        credibility-protocol-preference;
        tunnel-source-protocol {
            spring-te;
        }
    }
}
ldp {
    auto-targeted-session;
    preference 1;
    interface ge-0/0/3.0;
    interface lo0.0;
}
mpls {
    interface ge-0/0/1.0;
    interface ge-0/0/2.0;
    interface ge-0/0/3.0;
    interface lo0.0;
}
source-packet-routing {
    segment-list seg1 {
```

```
        inherit-label-nexthops;
        auto-translate;
        hop1 ip-address 10.1.4.2;
        hop2 ip-address 10.1.5.2;
        hop3 ip-address 10.1.6.2;
    }
    source-routing-path sr_static_r5 {
        ldp-tunneling;
        to 192.168.100.4;
        binding-sid 1003001;
        primary {
            seg1;
        }
    }
}
```

```
user@R1#show routing-options
router-id 192.168.100.1;
```

Verification

IN THIS SECTION

- Verifying LDP Tunneling over SR-TE | **1624**
- Verify LDP Forwarding to the Remote PE Device | **1626**
- Verifying the Advertised Label | **1629**

To confirm that the configuration is working properly, perform the following tasks:

Verifying LDP Tunneling over SR-TE

Purpose

Verify that the LDP over SR-TE tunnel is enabled and the LDP tunnel to the remote edge router is taking the right path.

Action

From operational mode, run the `show spring-traffic-engineering lsp detail` command.

On R1

```
user@R1>show spring-traffic-engineering lsp detail
Name: sr_static_r5
  Tunnel-source: Static configuration
  To: 192.168.100.2
  State: Up
  LDP-tunneling enabled
    Path: seg1
    Outgoing interface: NA
    Auto-translate status: Enabled Auto-translate result: Success
    Compute Status:Disabled , Compute Result:N/A , Compute-Profile Name:N/A
    BFD status: N/A BFD name: N/A
    ERO Valid: true
      SR-ERO hop count: 3
        Hop 1 (Strict):
          NAI: IPv4 Adjacency ID, 0.0.0.0 -> 10.1.4.2
          SID type: 20-bit label, Value: 80104
        Hop 2 (Strict):
          NAI: IPv4 Adjacency ID, 0.0.0.0 -> 10.1.5.2
          SID type: 20-bit label, Value: 80204
        Hop 3 (Strict):
          NAI: IPv4 Adjacency ID, 0.0.0.0 -> 10.1.6.2
          SID type: 20-bit label, Value: 80304


Total displayed LSPs: 1 (Up: 1, Down: 0)
```

On R2

```
user@R2>show spring-traffic-engineering lsp detail

Name: sr_static_r1
  Tunnel-source: Static configuration
  To: 192.168.100.1
  State: Up
  LDP-tunneling enabled
```

```
    Path: seg1
    Outgoing interface: NA
    Auto-translate status: Enabled Auto-translate result: Success
    Compute Status:Disabled , Compute Result:N/A , Compute-Profile Name:N/A
    BFD status: N/A BFD name: N/A
    ERO Valid: true
      SR-ERO hop count: 3
        Hop 1 (Strict):
          NAI: IPv4 Adjacency ID, 0.0.0.0 -> 10.1.6.1
          SID type: 20-bit label, Value: 80504
        Hop 2 (Strict):
          NAI: IPv4 Adjacency ID, 0.0.0.0 -> 10.1.5.1
          SID type: 20-bit label, Value: 80300
        Hop 3 (Strict):
          NAI: IPv4 Adjacency ID, 0.0.0.0 -> 10.1.4.1
          SID type: 20-bit label, Value: 80200


Total displayed LSPs: 1 (Up: 1, Down: 0)
```

Meaning

- On R1, the LDP tunnel is established with the remote edge router **192.168.100.2** in the SR-TE core network. You can also see the SID label values **80104, 80204, 80304** in the output.
- On R2, the LDP tunnel is established with the remote edge router **192.168.100.1** in the SR-TE core network. You can also see the SID label values **80504, 80300, 80200** in the output.

Verify LDP Forwarding to the Remote PE Device

Purpose

Verify that the route to the remote PE router uses LDP forwarding and is tunneled over SR-TE.

Action

From operational mode, run the `show route` *`destination-prefix`* command.

On R1

Verify that the route to the remote PE (**PE2**) router is through LDP over SR-TE tunnel.

```
user@R1>show route 192.168.100.6

inet.0: 24 destinations, 24 routes (24 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.100.6/32   *[IS-IS/18] 5d 13:10:09, metric 20
                    >  to 10.1.3.2 via ge-0/0/1.0

inet.3: 8 destinations, 13 routes (5 active, 0 holddown, 6 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.100.6/32   *[LDP/1] 5d 13:10:09, metric 1
                    >  to 10.1.4.2 via ge-0/0/2.0, Push 16, Push 80304, Push 80204(top)
                       to 10.1.3.2 via ge-0/0/1.0, Push 16, Push 80304, Push 80204, Push 85003,
Push 85004(top)
```

On R2

Verify that the route to the remote PE (**PE1**) router is through LDP over SR-TE tunnel.

```
user@R2>show route 192.168.100.5

inet.0: 24 destinations, 24 routes (24 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.100.5/32   *[IS-IS/18] 5d 13:20:15, metric 20
                    >  to 10.1.3.1 via ge-0/0/1.0

inet.3: 8 destinations, 13 routes (5 active, 0 holddown, 6 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.100.5/32   *[LDP/9] 5d 13:20:15, metric 1
                    >  to 10.1.6.1 via ge-0/0/2.0, Push 16, Push 80200, Push 80300(top)
                       to 10.1.3.1 via ge-0/0/1.0, Push 16, Push 80200, Push 80300, Push 85004,
Push 85003(top)
```

On PE1

Verify that the route to the remote PE (**PE2**) router is through a targeted LDP session to the remote PE.

```
user@PE1>show route 192.168.100.6

inet.0: 22 destinations, 22 routes (22 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.100.6/32   *[IS-IS/18] 1w3d 15:58:20, metric 30
                    >  to 10.1.2.2 via ge-0/0/3.0

inet.3: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.100.6/32   *[LDP/9] 1w0d 16:00:05, metric 1
                    >  to 10.1.2.2 via ge-0/0/3.0, Push 18
```

On PE2

Verify that the route to the remote PE (**PE1**) router is through a targeted LDP session to the remote PE.

```
user@PE2>show route 192.168.100.5

inet.0: 22 destinations, 22 routes (22 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.100.5/32   *[IS-IS/18] 1w3d 15:59:19, metric 30
                    >  to 10.1.7.1 via ge-0/0/3.0

inet.3: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.100.5/32   *[LDP/9] 1w0d 16:01:14, metric 1
                    >  to 10.1.7.1 via ge-0/0/3.0, Push 18
```

Meaning

- On R1, you can see the LDP label as **16** and the SR-TE label stacks as **80304, 80204, 85003, 85004**.
- On R2, you can see the LDP label as **16** and the SR-TE label stacks as **80200, 80300, 85004, 85003**.
- On PE1 and PE2, you can see the LDP label as **18** and **19**, respectively.

Verifying the Advertised Label

Purpose

Verify the labels advertised for the forwarding equivalence class (FEC).

Action

From operational mode, run the `show ldp database` command.

On R1

Verify the labels advertised towards the directly connected PE (PE1) and the labels received from remote edge router (R2).

```
user@R1>show ldp database

Input label database, 192.168.100.1:0--192.168.100.2:0
Labels received: 4
  Label     Prefix
     17      192.168.100.1/32
      3      192.168.100.2/32
     18      192.168.100.5/32
     16      192.168.100.6/32

Output label database, 192.168.100.1:0--192.168.100.2:0
Labels advertised: 4
  Label     Prefix
      3      192.168.100.1/32
     17      192.168.100.2/32
     16      192.168.100.5/32
     18      192.168.100.6/32

Input label database, 192.168.100.1:0--192.168.100.5:0
Labels received: 4
  Label     Prefix
     17      192.168.100.1/32
     18      192.168.100.2/32
      3      192.168.100.5/32
     19      192.168.100.6/32

Output label database, 192.168.100.1:0--192.168.100.5:0
```

```
Labels advertised: 4
  Label     Prefix
      3      192.168.100.1/32
     17      192.168.100.2/32
     16      192.168.100.5/32
     18      192.168.100.6/32
```

On R2

Verify the labels advertised towards the directly connected PE (PE2) and the labels received from remote edge router (R1).

```
user@R2>show ldp database

Input label database, 192.168.100.2:0--192.168.100.1:0
Labels received: 4
  Label     Prefix
      3      192.168.100.1/32
     17      192.168.100.2/32
     16      192.168.100.5/32
     18      192.168.100.6/32

Output label database, 192.168.100.2:0--192.168.100.1:0
Labels advertised: 4
  Label     Prefix
     17      192.168.100.1/32
      3      192.168.100.2/32
     18      192.168.100.5/32
     16      192.168.100.6/32

Input label database, 192.168.100.2:0--192.168.100.6:0
Labels received: 4
  Label     Prefix
     18      192.168.100.1/32
     17      192.168.100.2/32
     19      192.168.100.5/32
      3      192.168.100.6/32

Output label database, 192.168.100.2:0--192.168.100.6:0
Labels advertised: 4
  Label     Prefix
     17      192.168.100.1/32
```

```
       3      192.168.100.2/32
      18      192.168.100.5/32
      16      192.168.100.6/32
```

On PE1

Verify the label for the remote PE (PE2) device's loopback address is advertised by edge device R1 to the local PE (PE1) device.

```
user@PE1>show ldp database

Input label database, 192.168.100.5:0--192.168.100.1:0
Labels received: 4
  Label     Prefix
      3      192.168.100.1/32
     17      192.168.100.2/32
     16      192.168.100.5/32
     18      192.168.100.6/32

Output label database, 192.168.100.5:0--192.168.100.1:0
Labels advertised: 4
  Label     Prefix
     17      192.168.100.1/32
     18      192.168.100.2/32
      3      192.168.100.5/32
     19      192.168.100.6/32
```

On PE2

Verify the label for the remote PE (PE1) device's loopback address is advertised by edge device R2 to the local PE (PE2) device.

```
user@PE2>show ldp database

Input label database, 192.168.100.6:0--192.168.100.2:0
Labels received: 4
  Label     Prefix
     17      192.168.100.1/32
      3      192.168.100.2/32
     18      192.168.100.5/32
     16      192.168.100.6/32
```

```
Output label database, 192.168.100.6:0--192.168.100.2:0
Labels advertised: 4
  Label     Prefix
     18      192.168.100.1/32
     17      192.168.100.2/32
     19      192.168.100.5/32
      3      192.168.100.6/32
```

Meaning

- On R1, you can see label **18** is advertised towards the directly connected PE (PE1) and the label **19** is received from remote edge router (R2).
- On R2, you can see label **17** is advertised towards the directly connected PE (PE2) and the label **19** is received from remote edge router (R1).
- On PE1, you can see label **18** is received from the local edge router (R1).
- On PE2, you can see label **17** is received from the local edge router (R2).

RELATED DOCUMENTATION

vLab Sandbox: Segment Routing - Basic

Example: Tunneling LDP over SR-TE in OSPF Network

IN THIS SECTION

- Overview | **1633**
- Requirements | **1633**
- Configuration | **1634**
- Verification | **1652**

Overview

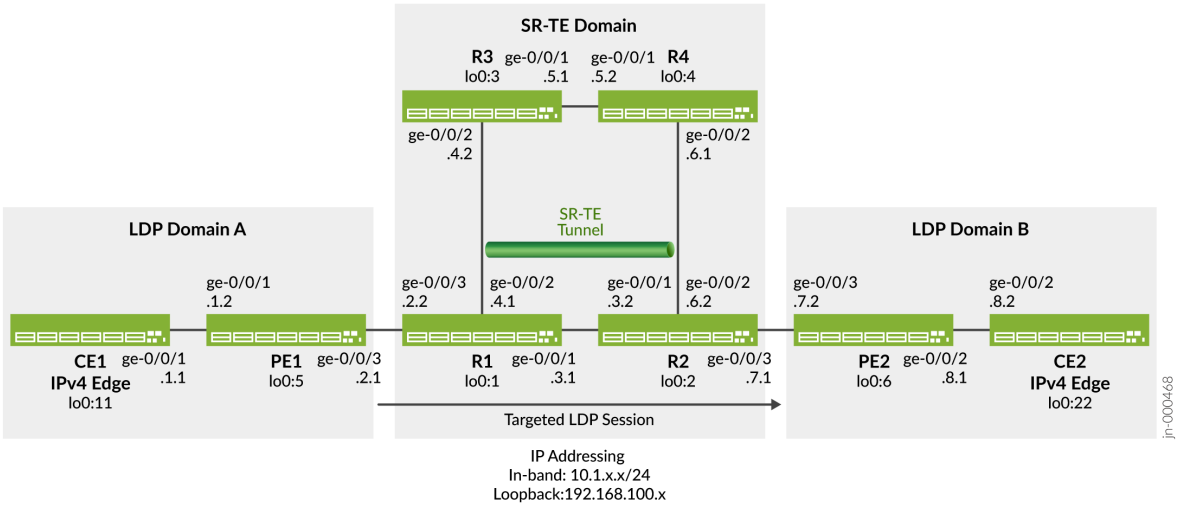
IN THIS SECTION

- Topology | 1633

This example shows how to configure LDP tunneling over SR-TE in an OSPF network. This is illustrated by verifying that the LDP over SR-TE tunnel is enabled and the LDP tunnel to the remote edge device takes the right path. It also shows that the route to the remote edge device uses LDP forwarding and is tunneled over SR-TE. In the following topology (Figure 107 on page 1633), PE1 and PE2 are ingress and egress devices that support IPv4 only devices CE1 and CE2. The devices R1, R2, R3, and R4 comprise an IPv4 only SR-TE core network. The topology shows two LDP domains: LDP domain consists of devices CE1 and PE1; LDP domain B consists of devices PE2 and CE2. The LDP domains are connected to the SR-TE core network, which extends the LSP session over the core by tunneling them over SR-TE.

Topology

Figure 107: Tunneling LDP over SR-TE in OSPF Network



Requirements

This example uses the following hardware and software components:

- MX Series routers as CE, PE, and core routers.

- Junos OS Release 22.4R1 or later running on all devices.

Configuration

IN THIS SECTION

- CLI Quick Configuration | **1634**
- Configuring PE1 | **1641**
- Configuring R1 Device | **1646**

To tunnel LDP LSP over SR-TE in your core network, perform these tasks:

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

NOTE: When you configure the `enhanced-ip` statement and commit the configuration, the following warning message appears prompting you to reboot the router:

```
'chassis'
WARNING: Chassis configuration for network services has been changed. A system reboot
is mandatory. Please reboot the system NOW. Continuing without a reboot might result
in unexpected system behavior.
commit complete
```

The reboot brings up the FPCs on the router.

Device CE1

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/1 description CE1-to-PE1
set interfaces ge-0/0/1 unit 0 family inet address 10.1.1.1/24
set interfaces lo0 unit 0 family inet address 192.168.100.11/32
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
```

```
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set routing-options router-id 192.168.100.11
```

Device PE1

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/1 description PE1-to-CE1
set interfaces ge-0/0/1 unit 0 family inet address 10.1.1.2/24
set interfaces ge-0/0/1 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/3 description PE1-to-R1
set interfaces ge-0/0/3 unit 0 family inet address 10.1.2.1/24
set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
set interfaces lo0 unit 0 family inet address 192.168.100.5/32
set interfaces lo0 unit 0 family mpls
set policy-options policy-statement export_bgp term a from protocol bgp
set policy-options policy-statement export_bgp term a from protocol direct
set policy-options policy-statement export_bgp term a then accept
set routing-instances CE1_vpn1 instance-type vrf
set routing-instances CE1_vpn1 protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set routing-instances CE1_vpn1 protocols ospf export export_bgp
set routing-instances CE1_vpn1 interface ge-0/0/1.0
set routing-instances CE1_vpn1 route-distinguisher 192.168.100.5:1
set routing-instances CE1_vpn1 vrf-target target:100:4
set routing-instances CE1_vpn1 vrf-table-label
set routing-options router-id 192.168.100.5
set routing-options autonomous-system 65410
set protocols bgp group ibgp1 type internal
set protocols bgp group ibgp1 local-address 192.168.100.5
set protocols bgp group ibgp1 family inet unicast
set protocols bgp group ibgp1 family inet-vpn unicast
set protocols bgp group ibgp1 neighbor 192.168.100.6
set protocols ldp interface ge-0/0/3.0
set protocols ldp interface lo0.0
set protocols mpls interface ge-0/0/3.0
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0 point-to-point
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

Device R1

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/1 description R1-to-R2
set interfaces ge-0/0/1 unit 0 family inet address 10.1.3.1/24
set interfaces ge-0/0/1 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/2 description R1-to-R3
set interfaces ge-0/0/2 unit 0 family inet address 10.1.4.1/24
set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/3 description R1-to-PE1
set interfaces ge-0/0/3 unit 0 family inet address 10.1.2.2/24
set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
set interfaces lo0 unit 0 family inet address 192.168.100.1/32
set interfaces lo0 unit 0 family mpls
set routing-options router-id 192.168.100.1
set protocols ldp auto-targeted-session
set protocols ldp preference 1
set protocols ldp interface ge-0/0/1.0
set protocols ldp interface ge-0/0/3.0
set protocols ldp interface lo0.0
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols mpls interface ge-0/0/3.0
set protocols ospf backup-spf-options use-post-convergence-lfa
set protocols ospf backup-spf-options use-source-packet-routing
set protocols ospf traffic-engineering l3-unicast-topology
set protocols ospf traffic-engineering credibility-protocol-preference
set protocols ospf traffic-engineering advertisement always
set protocols ospf traffic-engineering tunnel-source-protocol spring-te
set protocols ospf source-packet-routing node-segment ipv4-index 5001
set protocols ospf source-packet-routing srgb start-label 80000
set protocols ospf source-packet-routing srgb index-range 50000
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0 point-to-point
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0 post-convergence-lfa
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0 ipv4-adjacency-segment protected index 108
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0 ipv4-adjacency-segment unprotected index 110
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0 point-to-point
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0 post-convergence-lfa
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0 ipv4-adjacency-segment protected index 104
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0 ipv4-adjacency-segment unprotected index 106
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0 point-to-point
```

```
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0 post-convergence-lfa
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0 ipv4-adjacency-segment protected index 100
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0 ipv4-adjacency-segment unprotected index 102
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols source-packet-routing segment-list seg1 inherit-label-nexthops
set protocols source-packet-routing segment-list seg1 auto-translate
set protocols source-packet-routing segment-list seg1 hop1 ip-address 10.1.4.2
set protocols source-packet-routing segment-list seg1 hop2 ip-address 10.1.5.2
set protocols source-packet-routing segment-list seg1 hop3 ip-address 10.1.6.2
set protocols source-packet-routing source-routing-path sr_static_r5 ldp-tunneling
set protocols source-packet-routing source-routing-path sr_static_r5 to 192.168.100.2
set protocols source-packet-routing source-routing-path sr_static_r5 binding-sid 1003001
set protocols source-packet-routing source-routing-path sr_static_r5 primary seg1
```

Device R2

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/1 description R2-to-R1
set interfaces ge-0/0/1 unit 0 family inet address 10.1.3.2/24
set interfaces ge-0/0/1 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/2 description R2-to-R4
set interfaces ge-0/0/2 unit 0 family inet address 10.1.6.2/24
set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/3 description R2-to-PE2
set interfaces ge-0/0/3 unit 0 family inet address 10.1.7.1/24
set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
set interfaces lo0 unit 0 family inet address 192.168.100.2/32
set interfaces lo0 unit 0 family mpls
set routing-options router-id 192.168.100.2
set protocols ldp auto-targeted-session
set protocols ldp preference 1
set protocols ldp interface ge-0/0/1.0
set protocols ldp interface ge-0/0/3.0
set protocols ldp interface lo0.0
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols mpls interface ge-0/0/3.0
set protocols ospf backup-spf-options use-post-convergence-lfa
set protocols ospf backup-spf-options use-source-packet-routing
set protocols ospf traffic-engineering l3-unicast-topology
set protocols ospf traffic-engineering credibility-protocol-preference
```

```
set protocols ospf traffic-engineering advertisement always
set protocols ospf traffic-engineering tunnel-source-protocol spring-te
set protocols ospf source-packet-routing node-segment ipv4-index 5002
set protocols ospf source-packet-routing srgb start-label 80000
set protocols ospf source-packet-routing srgb index-range 50000
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0 point-to-point
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0 post-convergence-lfa
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0 ipv4-adjacency-segment protected index 500
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0 ipv4-adjacency-segment unprotected index 502
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0 point-to-point
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0 post-convergence-lfa
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0 ipv4-adjacency-segment protected index 504
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0 ipv4-adjacency-segment unprotected index 506
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0 point-to-point
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0 post-convergence-lfa
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0 ipv4-adjacency-segment protected index 508
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0 ipv4-adjacency-segment unprotected index 510
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols source-packet-routing segment-list seg1 inherit-label-nexthops
set protocols source-packet-routing segment-list seg1 auto-translate
set protocols source-packet-routing segment-list seg1 hop1 ip-address 10.1.6.1
set protocols source-packet-routing segment-list seg1 hop2 ip-address 10.1.5.1
set protocols source-packet-routing segment-list seg1 hop3 ip-address 10.1.4.1
set protocols source-packet-routing source-routing-path sr_static_r1 ldp-tunneling
set protocols source-packet-routing source-routing-path sr_static_r1 to 192.168.100.1
set protocols source-packet-routing source-routing-path sr_static_r1 binding-sid 1003001
set protocols source-packet-routing source-routing-path sr_static_r1 primary seg1
```

Device R3

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/1 description R3-to-R4
set interfaces ge-0/0/1 unit 0 family inet address 10.1.5.1/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 description R3-to-R1
set interfaces ge-0/0/2 unit 0 family inet address 10.1.4.2/24
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.100.3/32
set interfaces lo0 unit 0 family mpls
set routing-options router-id 192.168.100.3
set protocols mpls interface ge-0/0/1.0
```

```
set protocols mpls interface ge-0/0/2.0
set protocols ospf backup-spf-options use-post-convergence-lfa
set protocols ospf backup-spf-options use-source-packet-routing
set protocols ospf traffic-engineering l3-unicast-topology
set protocols ospf traffic-engineering credibility-protocol-preference
set protocols ospf traffic-engineering advertisement always
set protocols ospf source-packet-routing node-segment ipv4-index 5003
set protocols ospf source-packet-routing srgb start-label 80000
set protocols ospf source-packet-routing srgb index-range 50000
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0 point-to-point
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0 post-convergence-lfa
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0 ipv4-adjacency-segment protected index 204
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0 ipv4-adjacency-segment unprotected index 206
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0 point-to-point
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0 post-convergence-lfa
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0 ipv4-adjacency-segment protected index 200
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0 ipv4-adjacency-segment unprotected index 202
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

Device R4

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/1 description R4-to-R3
set interfaces ge-0/0/1 unit 0 family inet address 10.1.5.2/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 description R4-to-R2
set interfaces ge-0/0/2 unit 0 family inet address 10.1.6.1/24
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.100.4/32
set interfaces lo0 unit 0 family mpls
set routing-options router-id 192.168.100.4
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols ospf backup-spf-options use-post-convergence-lfa
set protocols ospf backup-spf-options use-source-packet-routing
set protocols ospf traffic-engineering l3-unicast-topology
set protocols ospf traffic-engineering credibility-protocol-preference
set protocols ospf traffic-engineering advertisement always
set protocols ospf source-packet-routing node-segment ipv4-index 5004
set protocols ospf source-packet-routing srgb start-label 80000
set protocols ospf source-packet-routing srgb index-range 50000
```

```
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0 point-to-point
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0 post-convergence-lfa
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0 ipv4-adjacency-segment protected index 300
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0 ipv4-adjacency-segment unprotected index 302
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0 point-to-point
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0 post-convergence-lfa
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0 ipv4-adjacency-segment protected index 304
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0 ipv4-adjacency-segment unprotected index 306
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

Device PE2

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/2 description PE2-to-CE2
set interfaces ge-0/0/2 unit 0 family inet address 10.1.8.1/24
set interfaces ge-0/0/2 unit 0 family mpls maximum-labels 8
set interfaces ge-0/0/3 description PE2-to-R2
set interfaces ge-0/0/3 unit 0 family inet address 10.1.7.2/24
set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 8
set interfaces lo0 unit 0 family inet address 192.168.100.6/32
set interfaces lo0 unit 0 family mpls
set policy-options policy-statement export_bgp term a from protocol bgp
set policy-options policy-statement export_bgp term a from protocol direct
set policy-options policy-statement export_bgp term a then accept
set routing-instances CE2_vpn1 instance-type vrf
set routing-instances CE2_vpn1 protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set routing-instances CE2_vpn1 protocols ospf export export_bgp
set routing-instances CE2_vpn1 interface ge-0/0/2.0
set routing-instances CE2_vpn1 route-distinguisher 192.168.100.6:1
set routing-instances CE2_vpn1 vrf-target target:100:4
set routing-instances CE2_vpn1 vrf-table-label
set routing-options router-id 192.168.100.6
set routing-options autonomous-system 65410
set protocols bgp group ibgp1 type internal
set protocols bgp group ibgp1 local-address 192.168.100.6
set protocols bgp group ibgp1 family inet unicast
set protocols bgp group ibgp1 family inet-vpn unicast
set protocols bgp group ibgp1 neighbor 192.168.100.5
set protocols ldp interface ge-0/0/3.0
set protocols ldp interface lo0.0
set protocols mpls interface ge-0/0/3.0
```

```
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0 point-to-point
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

Device CE2

```
set chassis network-services enhanced-ip
set interfaces ge-0/0/1 description CE2-to-PE2
set interfaces ge-0/0/2 unit 0 family inet address 10.1.8.2/24
set interfaces lo0 unit 0 family inet address 192.168.100.22/32
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set routing-options router-id 192.168.100.22
```

Configuring PE1

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the CLI User Guide.

To configure device PE1:

1. Configure the network services mode as Enhanced IP. Enhanced IP sets the router's network services to enhanced Internet Protocol and uses enhanced mode capabilities.

```
[edit chassis]
user@PE1#set network-services enhanced-ip
```

2. Configure the device's interfaces.

```
[edit interfaces]
user@PE1#set ge-0/0/1 description PE1-to-CE1
user@PE1#set ge-0/0/1 unit 0 family inet address 10.1.1.2/24
user@PE1#set ge-0/0/3 description PE1-to-R1
user@PE1#set ge-0/0/3 unit 0 family inet address 10.1.2.1/24
user@PE1#set ge-0/0/3 unit 0 family mpls
```

```
user@PE1#set lo0 unit 0 family inet address 192.168.100.5/32
user@PE1#set lo0 unit 0 family mpls
```

3. Configure policy options to export BGP routes to the CE router, which runs the OSPF protocol in this example.

```
[edit policy-options]
user@PE1#set policy-statement export_bgp term a from protocol bgp
user@PE1#set policy-statement export_bgp term a from protocol direct
user@PE1#set policy-statement export_bgp term a then accept
```

4. Configure a Layer 3 VPN routing instance to support the OSPF-based CE1 device.

```
[edit routing-instances]
user@PE1#set CE1_vpn1 instance-type vrf
user@PE1#set CE1_vpn1 protocols ospf area 0.0.0.0 interface ge-0/0/1.0
user@PE1#set CE1_vpn1 protocols ospf export export_bgp
user@PE1#set CE1_vpn1 interface ge-0/0/1.0
user@PE1#set CE1_vpn1 route-distinguisher 192.168.100.5:1
user@PE1#set CE1_vpn1 vrf-target target:100:4
user@PE1#set CE1_vpn1 vrf-table-label
```

5. Configure the router ID and autonomous system number for Device PE1.

```
[edit routing-options]
user@PE1#set router-id 192.168.100.5
userPE1# set autonomous-system 65410
```

6. Configure OSPF, LDP, and MPLS on the interfaces connected to the core network.

```
[edit protocols]
user@PE1#set ospf area 0.0.0.0 interface ge-0/0/3.0 point-to-point
user@PE1#set ospf area 0.0.0.0 lo0.0 passive
user@PE1#set ldp interface ge-0/0/3.0
user@PE1#set ldp interface lo0.0
user@PE1#set mpls interface ge-0/0/3.0
user@PE1#set mpls interface lo0.0
```

7. Configure BGP between the PE devices.

```
[edit protocols]
user@PE1#set bgp group ibgp1 type internal
user@PE1#set bgp group ibgp1 local-address 192.168.100.5
user@PE1#set bgp group ibgp1 family inet unicast
user@PE1#set bgp group ibgp1 family inet-vpn unicast
user@PE1#set bgp group ibgp1 neighbor 192.168.100.6
```

Results

From configuration mode, confirm your configuration by entering the `show chassis`, `show interfaces`, `show policy-options`, `show routing-instances`,`show routing-options`, and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@PE1#show chassis
network-services enhanced-ip;
```

```
user@PE1#show interfaces
ge-0/0/1 {
    description PE1-to-CE1;
    unit 0 {
        family inet {
            address 10.1.1.2/24;
        }
        family mpls {
            maximum-labels 8;
        }
    }
}
ge-0/0/3 {
    description PE1-to-R1;
    unit 0 {
        family inet {
            address 10.1.2.1/24;
        }
```

```
        family mpls {
            maximum-labels 8;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 192.168.100.5/32;
        }
        family mpls;
    }
}
```

```
user@PE1#show policy-options
policy-statement export_bgp {
    term a {
        from protocol [ bgp direct ];
        then accept;
    }
}
```

```
user@PE1#show routing-instances
CE1_vpn1 {
    instance-type vrf;
    protocols {
        ospf {
            area 0.0.0.0 {
                interface ge-0/0/1.0;
            }
            export export_bgp;
        }
    }
    interface ge-0/0/1.0;
    route-distinguisher 192.168.100.5:1;
    vrf-target target:100:4;
```

```
    vrf-table-label;
}
```

```
user@PE1#show routing-options
router-id 192.168.100.5;
autonomous-system 65410;
```

```
user@PE1#show protocols
bgp {
    group ibgp1 {
        type internal;
        local-address 192.168.100.5;
        family inet {
            unicast;
        }
        family inet-vpn {
            unicast;
        }
        neighbor 192.168.100.6;
    }
}
ldp {
    interface ge-0/0/3.0;
    interface lo0.0;
}
mpls {
    interface ge-0/0/3.0;
    interface lo0.0;
}
ospf {
    area 0.0.0.0 {
        interface ge-0/0/3.0 {
            point-to-point;
        }
        interface lo0.0 {
            passive;
        }
```

```
    }
}
```

Configuring R1 Device

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the CLI User Guide.

To configure device R1:

1. Configure the network services mode as Enhanced IP. Enhanced IP sets the router's network services to enhanced Internet Protocol and uses enhanced mode capabilities.

```
[edit chassis]
user@R1#set network-services enhanced-ip
```

After you configure the `enhanced-ip` statement and commit the configuration, the following warning message appears prompting you to reboot the router:

```
'chassis'
WARNING: Chassis configuration for network services has been changed. A system reboot is
mandatory. Please reboot the system NOW. Continuing without a reboot might result in
unexpected system behavior.
commit complete
```

The reboot brings up the FPCs on the router.

2. Configure the device's interfaces.

```
[edit interfaces]
user@R1#set ge-0/0/1 description R1-to-R2
user@R1#set ge-0/0/1 unit 0 family inet address 10.1.3.1/24
user@R1#set ge-0/0/1 unit 0 family mpls maximum-labels 8
user@R1#set ge-0/0/2 description R1-to-R3
user@R1#set ge-0/0/2 unit 0 family inet address 10.1.4.1/24
user@R1#set ge-0/0/2 unit 0 family mpls maximum-labels 8
user@R1#set ge-0/0/3 description R1-to-PE1
user@R1#set ge-0/0/3 unit 0 family inet address 10.1.2.2/24
```

```
user@R1#set ge-0/0/3 unit 0 family mpls maximum-labels 8
user@R1#set lo0 unit 0 family inet address 192.168.100.1/32
user@R1#set lo0 unit 0 family mpls
```

3. Configure routing options to identify the router in the domain.

```
[edit routing-options]
user@R1#set router-id 192.168.100.1
```

4. Configure OSPF adjacency SIDs on the interfaces and allocate SRGB labels to enable segment routing. The labels in the entire SRGB are available for OSPF. Prefix SIDs (and Node SIDs) are indexed from the SRGB.

```
[edit protocols]
user@R1#set ospf area 0.0.0.0 interface ge-0/0/1.0 ipv4-adjacency-segment protected index
108
user@R1#set ospf area 0.0.0.0 interface ge-0/0/1.0 ipv4-adjacency-segment unprotected index
110
user@R1#set ospf area 0.0.0.0 interface ge-0/0/1.0 post-convergence-lfa
user@R1#set ospf area 0.0.0.0 interface ge-0/0/1.0 point-to-point
user@R1#set ospf area 0.0.0.0 interface ge-0/0/2.0 ipv4-adjacency-segment protected index
104
user@R1#set ospf area 0.0.0.0 interface ge-0/0/2.0 ipv4-adjacency-segment unprotected index
106
user@R1#set ospf area 0.0.0.0 interface ge-0/0/2.0 post-convergence-lfa
user@R1#set ospf area 0.0.0.0 interface ge-0/0/2.0 point-to-point
user@R1#set ospf area 0.0.0.0 interface ge-0/0/3.0 ipv4-adjacency-segment protected index
100
user@R1#set ospf area 0.0.0.0 interface ge-0/0/3.0 ipv4-adjacency-segment unprotected index
102
user@R1#set ospf area 0.0.0.0 interface ge-0/0/3.0 post-convergence-lfa
user@R1#set ospf area 0.0.0.0 interface ge-0/0/3.0 point-to-point
user@R1#set ospf area 0.0.0.0 interface lo0.0 passive
user@R1#set ospf source-packet-routing srgb start-label 80000
user@R1#set ospf source-packet-routing srgb index-range 50000
user@R1#set ospf source-packet-routing node-segment ipv4-index 5001
```

5. Configure TI-LFA to enable protection against link and node failures. SR using TI-LFA provides faster restoration of network connectivity by routing the traffic instantly to a backup or an alternate path if the primary path fails or becomes unavailable.

```
[edit protocols]
user@R1#set ospf backup-spf-options use-post-convergence-lfa
user@R1#set ospf backup-spf-options use-source-packet-routing
```

6. Configure OSPF traffic engineering parameters.

```
[edit protocols]
user@R1#set ospf traffic-engineering l3-unicast-topology
user@R1#set ospf traffic-engineering credibility-protocol-preference
user@R1#set protocols ospf traffic-engineering advertisement always
```

7. Enable LDP tunneling over SR-TE.

```
[edit protocols]
user@R1#set ospf traffic-engineering tunnel-source-protocol spring-te
```

8. Configure MPLS and LDP protocols on the interfaces in the LDP domain to exchange labels in the LDP domain.

```
[edit protocols]
user@R1#set ldp preference 1
user@R1#set protocols ldp interface ge-0/0/1.0
user@R1#set ldp interface ge-0/0/3.0
user@R1#set ldp interface lo0.0
user@R1#set mpls interface ge-0/0/1.0
user@R1#set mpls interface ge-0/0/2.0
user@R1#set mpls interface ge-0/0/3.0
user@R1#set mpls interface lo0.0
```

9. Enable targeted LDP session between the edge routers in the LDP domain.

```
[edit protocols]
user@R1#set ldp auto-targeted-session
```

10. Configure a segment list to route the traffic to a specific path.

```
[edit protocols]
user@R1#set source-packet-routing segment-list seg1 inherit-label-nexthops
user@R1#set source-packet-routing segment-list seg1 auto-translate
user@R1#set source-packet-routing segment-list seg1 hop1 ip-address 192.168.4.2
user@R1#set source-packet-routing segment-list seg1 hop2 ip-address 192.168.5.2
user@R1#set source-packet-routing segment-list seg1 hop3 ip-address 192.168.6.2
```

11. Configure SR-TE LSP to the remote edge routers to enable LDP tunneling over SR-TE.

```
[edit protocols]
user@R1#set source-packet-routing source-routing-path sr_static_r5 ldp-tunneling
user@R1#set source-packet-routing source-routing-path sr_static_r5 to 192.168.66.66
user@R1#set source-packet-routing source-routing-path sr_static_r5 binding-sid 1003001
user@R1#set source-packet-routing source-routing-path sr_static_r5 primary seg1
```

Results

From configuration mode, confirm your configuration by entering the `show chassis`, `show interfaces`, `show routing-options`, and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R1#show chassis
network-services enhanced-ip;
```

```
user@R1#show interfaces
ge-0/0/1 {
    description R1-to-R2;
    unit 0 {
        family inet {
            address 10.1.3.1/24;
        }
        family mpls {
            maximum-labels 8;
        }
    }
}
```

```
ge-0/0/2 {
    description R1-to-R3;
    unit 0 {
        family inet {
            address 10.1.4.1/24;
        }
        family mpls {
            maximum-labels 8;
        }
    }
}
ge-0/0/3 {
    description R1-to-PE1;
    unit 0 {
        family inet {
            address 10.1.2.2/24;
        }
        family mpls {
            maximum-labels 8;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 192.168.100.1/32;
        }
        family mpls;
    }
}
```

```
user@R1#show protocols
ldp {
    auto-targeted-session;
    preference 1;
    interface ge-0/0/1.0;
    interface ge-0/0/3.0;
    interface lo0.0;
}
mpls {
    interface ge-0/0/1.0;
```

```
    interface ge-0/0/2.0;
    interface ge-0/0/3.0;
    interface lo0.0;
}
ospf {
    backup-spf-options {
        use-post-convergence-lfa;
        use-source-packet-routing;
    }
    traffic-engineering {
        l3-unicast-topology;
        credibility-protocol-preference;
        advertisement always;
        tunnel-source-protocol {
            spring-te;
        }
    }
    source-packet-routing {
        node-segment ipv4-index 5001;
        srgb start-label 80000 index-range 50000;
    }
    area 0.0.0.0 {
        interface ge-0/0/1.0 {
            point-to-point;
            post-convergence-lfa;
            ipv4-adjacency-segment {
                protected index 108;
                unprotected index 110;
            }
        }
        interface ge-0/0/2.0 {
            point-to-point;
            post-convergence-lfa;
            ipv4-adjacency-segment {
                protected index 104;
                unprotected index 106;
            }
        }
        interface ge-0/0/3.0 {
            point-to-point;
            post-convergence-lfa;
            ipv4-adjacency-segment {
                protected index 100;
```

```
                unprotected index 102;
            }
        }
        interface lo0.0 {
            passive;
        }
    }
}
source-packet-routing {
    segment-list seg1 {
        inherit-label-nexthops;
        auto-translate;
        hop1 ip-address 10.1.4.2;
        hop2 ip-address 10.1.5.2;
        hop3 ip-address 10.1.6.2;
    }
    source-routing-path sr_static_r5 {
        ldp-tunneling;
        to 192.168.100.2;
        binding-sid 1003001;
        primary {
            seg1;
        }
    }
}
```

```
user@R1#show routing-options
router-id 192.168.100.1;
```

Verification

IN THIS SECTION

- Verifying LDP Tunneling over SR-TE | **1653**
- Verifying the Advertised Label | **1655**
- Verify LDP Forwarding to the Remote PE Device | **1658**
- Verify End-to-End Reachability | **1660**

To confirm that the configuration is working properly, perform the following tasks:

Verifying LDP Tunneling over SR-TE

Purpose

Verify that the LDP over SR-TE tunnel is enabled and the LDP tunnel to the remote edge router is taking the right path.

Action

From operational mode, run the `show spring-traffic-engineering lsp detail` command.

On R1

```
user@R1>show spring-traffic-engineering lsp detail
Name: sr_static_r5
  Tunnel-source: Static configuration
  Tunnel Forward Type: SRMPLS
  To: 192.168.100.2
  Te-group-id: 0
  State: Up
  LDP-tunneling enabled
    Path: seg1
    Path Status: NA
    Outgoing interface: NA
    Auto-translate status: Enabled Auto-translate result: Success
    Compute Status:Disabled , Compute Result:N/A , Compute-Profile Name:N/A
    BFD status: N/A BFD name: N/A
    BFD remote-discriminator: N/A
    Segment ID : 128
    ERO Valid: true
      SR-ERO hop count: 3
        Hop 1 (Strict):
          NAI: IPv4 Adjacency ID, 0.0.0.0 -> 10.1.4.2
          SID type: 20-bit label, Value: 80104
        Hop 2 (Strict):
          NAI: IPv4 Adjacency ID, 0.0.0.0 -> 10.1.5.2
          SID type: 20-bit label, Value: 80204
        Hop 3 (Strict):
          NAI: IPv4 Adjacency ID, 0.0.0.0 -> 10.1.6.2
          SID type: 20-bit label, Value: 80304
```

```
Total displayed LSPs: 1 (Up: 1, Down: 0)
```

On R2

```
user@R2>show spring-traffic-engineering lsp detail
Name: sr_static_r1
  Tunnel-source: Static configuration
  Tunnel Forward Type: SRMPLS
  To: 192.168.100.1
  Te-group-id: 0
  State: Up
  LDP-tunneling enabled
    Path: seg1
    Path Status: NA
    Outgoing interface: NA
    Auto-translate status: Enabled Auto-translate result: Success
    Compute Status:Disabled , Compute Result:N/A , Compute-Profile Name:N/A
    BFD status: N/A BFD name: N/A
    BFD remote-discriminator: N/A
    Segment ID : 128
    ERO Valid: true
      SR-ERO hop count: 3
        Hop 1 (Strict):
          NAI: IPv4 Adjacency ID, 0.0.0.0 -> 10.1.6.1
          SID type: 20-bit label, Value: 80504
        Hop 2 (Strict):
          NAI: IPv4 Adjacency ID, 0.0.0.0 -> 10.1.5.1
          SID type: 20-bit label, Value: 80300
        Hop 3 (Strict):
          NAI: IPv4 Adjacency ID, 0.0.0.0 -> 10.1.4.1
          SID type: 20-bit label, Value: 80200


Total displayed LSPs: 1 (Up: 1, Down: 0)
```

Meaning

- On R1, the LDP tunnel is established with the remote edge router **192.168.100.2** in the SR-TE core network. You can also see the SID label values **80104, 80204, 80304** in the output.

- On R2, the LDP tunnel is established with the remote edge router **192.168.100.1** in the SR-TE core network. You can also see the SID label values **80504, 80300, 80200** in the output.

Verifying the Advertised Label

Purpose

Verify the labels advertised for the forwarding equivalence class (FEC).

Action

From operational mode, run the `show ldp database` command.

On R1

Verify the labels advertised towards the directly connected PE (PE1) and the labels received from remote edge router (R2).

```
user@R1>show ldp database

Input label database, 192.168.100.1:0--192.168.100.2:0
Labels received: 4
  Label     Prefix
     18      192.168.100.1/32
      3      192.168.100.2/32
     20      192.168.100.5/32
     16      192.168.100.6/32


Output label database, 192.168.100.1:0--192.168.100.2:0
Labels advertised: 4
  Label     Prefix
      3      192.168.100.1/32
     4115    192.168.100.2/32
     4114    192.168.100.5/32
     4117      192.168.100.6/32

Input label database, 192.168.100.1:0--192.168.100.5:0
Labels received: 4
  Label     Prefix
     17      192.168.100.1/32
     22      192.168.100.2/32
```

```
      3      192.168.100.5/32
     24      192.168.100.6/32

Output label database, 192.168.100.1:0--192.168.100.5:0
Labels advertised: 4
  Label     Prefix
      3      192.168.100.1/32
   4115      192.168.100.2/32
   4114      192.168.100.5/32
   4117        192.168.100.6/32
```

On R2

Verify the labels advertised towards the directly connected PE (PE2) and the labels received from remote edge router (R1).

```
user@R2>show ldp database

Input label database, 192.168.100.2:0--192.168.100.1:0
Labels received: 4
  Label     Prefix
      3      192.168.100.1/32
   4115      192.168.100.2/32
   4114      192.168.100.5/32
   4117      192.168.100.6/32

Output label database, 192.168.100.2:0--192.168.100.1:0
Labels advertised: 4
  Label     Prefix
     18      192.168.100.1/32
      3      192.168.100.2/32
     20      192.168.100.5/32
     16      192.168.100.6/32

Input label database, 192.168.100.2:0--192.168.100.6:0
Labels received: 4
  Label     Prefix
     24      192.168.100.1/32
     17      192.168.100.2/32
     25      192.168.100.5/32
      3      192.168.100.6/32
```

```
Output label database, 192.168.100.2:0--192.168.100.6:0
Labels advertised: 4
  Label     Prefix
     18      192.168.100.1/32
      3      192.168.100.2/32
     20      192.168.100.5/32
     16      192.168.100.6/32
```

On PE1

Verify the label for the remote PE (PE2) device's loopback address is advertised by edge device R1 to the local PE (PE1) device.

```
user@PE1>show ldp database

Input label database, 192.168.100.5:0--192.168.100.1:0
Labels received: 4
  Label     Prefix
      3      192.168.100.1/32
   4115      192.168.100.2/32
   4114      192.168.100.5/32
   4117        192.168.100.6/32

Output label database, 192.168.100.5:0--192.168.100.1:0
Labels advertised: 4
  Label     Prefix
     17      192.168.100.1/32
     22      192.168.100.2/32
      3      192.168.100.5/32
     24      192.168.100.6/32
```

On PE2

Verify the label for the remote PE (PE1) device's loopback address is advertised by edge device R2 to the local PE (PE2) device.

```
user@PE2>show ldp database

Input label database, 192.168.100.6:0--192.168.100.2:0
Labels received: 4
  Label     Prefix
     18      192.168.100.1/32
```

```
       3      192.168.100.2/32
      20      192.168.100.5/32
      16      192.168.100.6/32

Output label database, 192.168.100.6:0--192.168.100.2:0
Labels advertised: 4
  Label     Prefix
     24      192.168.100.1/32
     17      192.168.100.2/32
     25      192.168.100.5/32
      3      192.168.100.6/32
```

Meaning

- On R1, you can see label **4117** is advertised towards the directly connected PE (PE1) and the label **27** is received from remote edge router (R2).
- On R2, you can see label **18** is advertised towards the directly connected PE (PE2) and the label **25** is received from remote edge router (R1).
- On PE1, you can see label **4117** is received from the local edge router (R1).
- On PE2, you can see label **18** is received from the local edge router (R2).

Verify LDP Forwarding to the Remote PE Device

Purpose

Verify that the route to the remote PE router uses LDP forwarding and is tunneled over SR-TE.

Action

From operational mode, run the `show route` *destination-prefix* command.

On R1

Verify that the route to the remote PE (**PE2**) router is through LDP over SR-TE tunnel.

```
user@R1>show route 192.168.100.6
inet.0: 29 destinations, 29 routes (29 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.100.6/32   *[OSPF/10/10] 1d 03:28:10, metric 2
```

```
                    >  to 10.1.3.2 via ge-0/0/1.0

inet.3: 10 destinations, 15 routes (5 active, 0 holddown, 8 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.100.6/32   *[LDP/1] 21:42:29, metric 1
                    >  to 10.1.4.2 via ge-0/0/2.0, Push 16, Push 80304, Push 80204(top)
                       to 10.1.3.2 via ge-0/0/1.0, Push 16, Push 80304, Push 80204, Push 85003,
Push 85004(top)
```

On R2

Verify that the route to the remote PE (**PE1**) router is through LDP over SR-TE tunnel.

```
user@R2>show route 192.168.100.5
inet.0: 29 destinations, 29 routes (29 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.100.5/32   *[OSPF/10/10] 22:22:45, metric 2
                    >  to 10.1.3.1 via ge-0/0/1.0

inet.3: 10 destinations, 15 routes (5 active, 0 holddown, 8 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.100.5/32   *[LDP/1] 22:22:45, metric 1
                    >  to 10.1.6.1 via ge-0/0/2.0, Push 4114, Push 80200, Push 80300(top)
                       to 10.1.3.1 via ge-0/0/1.0, Push 4114, Push 80200, Push 80300, Push
85004, Push 85003(top)
```

On PE1

Verify that the route to the remote PE (**PE2**) router is through a targeted LDP session to the remote PE.

```
user@PE1>show route 192.168.100.6

inet.0: 27 destinations, 27 routes (27 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.100.6/32   *[OSPF/10] 01:15:29, metric 3
                    >  to 10.1.2.2 via ge-0/0/3.0

inet.3: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
```

```
+ = Active Route, - = Last Active, * = Both

192.168.100.6/32   *[LDP/9] 01:15:29, metric 1
                    >  to 10.1.2.2 via ge-0/0/3.0, Push 4117
```

On PE2

Verify that the route to the remote PE (**PE1**) router is through a targeted LDP session to the remote PE.

```
user@PE2>show route 192.168.100.5

inet.0: 27 destinations, 27 routes (27 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.100.5/32   *[OSPF/10] 01:27:40, metric 3
                    >  to 10.1.7.1 via ge-0/0/3.0

inet.3: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.100.5/32   *[LDP/9] 01:27:40, metric 1
                    >  to 10.1.7.1 via ge-0/0/3.0, Push 20
```

Meaning

- On R1, you can see the LDP label as **16** and the SR-TE label stacks as **80304, 80204, 85003, 85004**.
- On R2, you can see the LDP label as **22** and the SR-TE label stacks as **80200, 80300, 85004, 85003**.
- On PE1 and PE2, you can see the LDP label as **4117** and **20**, respectively.

Verify End-to-End Reachability

Purpose

Verify CE1 can ping CE2 by using the ping 192.168.100.22 source 192.168.100.11 count 2 operational mode command.

Action

```
user@CE1> ping 192.168.100.22 source 192.168.100.11 count 2
PING 192.168.100.22 (192.168.100.22): 56 data bytes
64 bytes from 192.168.100.22: icmp_seq=0 ttl=58 time=8.772 ms
64 bytes from 192.168.100.22: icmp_seq=1 ttl=58 time=9.189 ms

--- 192.168.100.22 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 8.772/8.980/9.189/0.209 ms
```

Meaning

The output from CE1 shows that CE1 can ping CE2.

RELATED DOCUMENTATION

Example: Tunneling LDP over SR-TE in IS-IS Network | 1602

MPLS TTL Propagation Flexibility for LDP-signaled LSPs

IN THIS SECTION

- Purpose | 1662
- LDP No Propagate TTL Configuration | 1662

We support disabling time-to-live (TTL) propagation at a more granular level. You can disable TTL propagation specifically for LDP-signaled label-switched paths (LSPs). When a route is very long, disable TTL propagation to ensure that the TTL doesn't expire while the packet is traversing the path. This feature also gives you more flexibility in hiding your network topology.

To disable TTL propagation for LDP-signaled LSPs, use the no-propagate-ttl statement at the `[edit protocol ldp]` hierarchy level.

NOTE: If the TTL value of the top label is less than the TTL value of the bottom label at an egress node, Junos OS copies the TTL value from the top label to the bottom label. In this case, the TTL value can still propagate down even when no-propagate-ttl is configured.

NOTE: If you don't configure the `no-propagate-ttl` statement, the device copies the MPLS TTL value to the inner IP header during the MPLS label removal process. The device copies that value only when the TTL of the inner IP header is greater than or equal to the MPLS TTL value. If the TTL of the inner IP header is less than the MPLS TTL value, the device does not copy the MPLS TTL value.

[See no-propagate-ttl.]

Purpose

The purpose is to support the configuration and functionality of the `no-propagate-ttl` option at the `[edit protocol ldp]` hierarchy.

LDP No Propagate TTL Configuration

The following command results in LDP following the no propagate TTL behavior.

```
set protocol ldp no-propagate-ttl
```

If you configure the global `no-propagate-ttl` option with this command, there is no change in behavior.

The `no-propagate-ttl` option has no impact on the independent RSVP LSP. However, when you use this option with LDPoRSVP, the TTL action of the RSVP LSP decides the TTL propagation behavior.

This means that when the router is acting as an ingress node for LDP and RSVP for the LDPoRSVP LSP, if you configure `no-decrement-ttl` for the RSVP LSP, then the LDPoRSVP does not propagate TTL. And if you don't configure the `no-decrement-ttl` option for the RSVP LSP, then it propagates TTL.

RELATED DOCUMENTATION

Example: Configuring MPLS TTL Propagation for LDP-signaled LSPs | 1663

no-propagate-ttl

show ldp overview

Example: Configuring MPLS TTL Propagation for LDP-signaled LSPs

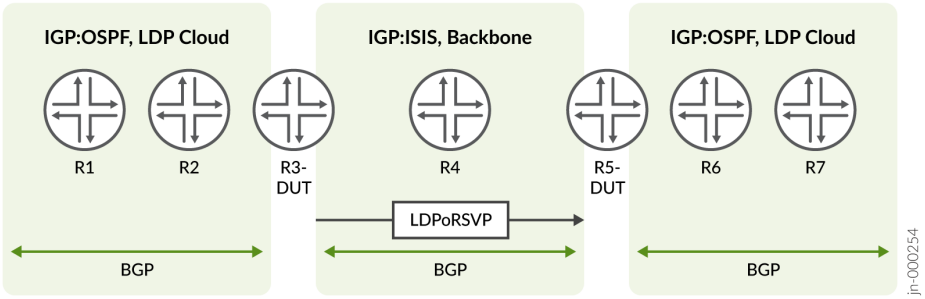
IN THIS SECTION

- Overview | 1663
- Topology | 1663
- Purpose | 1663
- Use Case for the No Propagate TTL and the Propagate TTL behavior at LDP | 1664
- Configuration | 1665

Overview

The following figure depicts a typical scenario in which the `no-propagate-ttl` statement at the `[edit protocol ldp]` hierarchy is beneficial.

Topology



In the figure, you can see two native LDP clouds connected by a backbone with LDPoRSVP.

- From Router R1, packets are encapsulated with the LDP label and sent to the destination R7.
- On Router R3, the LDP label from a packet is stripped to an IP packet. The packet is then encapsulated with LDP over RSVP (LDPoRSVP) and sent across the backbone.
- On Router R5, the LDPoRSVP labels are stripped to an IP packet, then the packet is again encapsulated in the LDP label and sent to the destination.

Purpose

The purpose is to do the following actions:

- Hide the two LDP clouds by using no TTL propagation
- Unhide the backbone (LDPoRSVP)

When a packet is sent from Router R1 to Router R7, these actions must be performed on two routers, R1 and R5. You cannot achieve this with the existing options. For example, when you use the global option (`set protocol mpls no-propagate-ttl`) on Router R5, it disables the TTL propagation LDPoRSVP backbone in the reverse direction (R7-R1). This happens because the option is applicable for both LDP and RSVP.

Use Case for the No Propagate TTL and the Propagate TTL behavior at LDP

LDP on Router R3 needs to support both no propagate TTL and the propagate TTL behavior.

From Router R3 with LDPoRSVP, in the R4 direction, the router needs to support the propagate TTL behavior. However, towards the native LDP (Router R2), the LDP needs to support the no propagate TTL behavior.

To achieve this result, we have introduced a new option, `no-propagate-ttl` under LDP that you need to configure for Router R3 and Router R5. This option disables the propagation of TTL for LDP paths.

In an LDPoRSVP scenario, propagation behavior depends on the RSVP no decrement TTL (`no-decrement-ttl`) option.

- If you configure the `no-propagate-ttl` option in the LDPoRSVP scenario, and the no decrement TTL (`no-decrement-ttl`) is not configured, then the TTL propagation takes place.
  For example:

  On Router R3, in the case of the LDPoRSVP scenario, if you set the following configuration, then TTL propagation takes place.

  ```
  user@host> set protocol ldp no-propagate-ttl
  ```

- If you configure the `no-decrement-ttl` option over the LSP between Router R3 and Router R5, then the TTL propagation is disabled.
  For example, on Router R3:

  ```
  user@host> set protocol ldp no-propagate-ttl
  user@host> set protocol mpls no-decrement-ttl
  ```

On Router R1, packets are encapsulated with the LDP label with TTL 255, as any of the `no-propagate-ttl` CLI is configured.

On Router R3:

- The LDP label from a packet is stripped to an IP header, and the TTL is not copied from the LDP label to the IP header.
- The packet is encapsulated with LDPoRSVP labels and sent across the backbone.
- The new option `ldp no-propagate-ttl` with `no-decrement-tll` decides whether the TTL should be propagated or not.
- The `no-decrement-ttl` option is not configured, so the usual TTL propagation occurs

On Router R5, the LDPoRSVP labels are stripped to the IP header. The new option is configured to support `no-propagate-ttl` for LDP protocol, and the IP packet is encapsulated with an LDP label with TTL 255 and sent across.

Configuration

IN THIS SECTION

- CLI Quick Configuration | **1665**
- Results | **1665**

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level.

```
set protocol ldp no-propagate-ttl
```

Results

We have modified the output of CLI command `show ldp overview` to display the TTL configuration.

Check the results of the configuration:

```
user@host> show ldp overview
Instance: master
Reference count: 4
Router ID: 10.1.1.1
```

```
Inet LSR ID: 10.1.1.4
Inet6 LSR ID: 10.1.1.6
LDP inet: enabled
LDP inet6: enabled
Transport preference: Single-stack
Message id: 21
Configuration sequence: 1
Deaggregate: disabled
Explicit null: disabled
IPv6 tunneling: disabled
Strict targeted hellos: disabled
Loopback if added: yes
Route preference: 9
Unicast transit LSP chaining: disabled
P2MP transit LSP chaining: disabled
Transit LSP statistics based on route statistics: disabled
LDP route acknowledgement: enabled
BGP export: enabled
No TTL propagate: enabled
LDP mtu discovery: disabled
LDP SR Mapping Client: disabled
Capabilities enabled: none
Egress FEC capabilities enabled: entropy-label-capability
Downstream unsolicited Sessions:
Operational: 2
Retention: liberal
Control: ordered
Auto targeted sessions:
Auto targeted: disabled
Dynamic tunnel session count: 0
P2MP:
Recursive route: disabled
No rsvp tunneling: disabled
Timers:
Keepalive interval: 10, Keepalive timeout: 30
Link hello interval: 5, Link hello hold time: 15
Targeted hello interval: 15, Targeted hello hold time: 45
Label withdraw delay: 60, Make before break timeout: 30
Make before break switchover delay: 3
Link protection timeout: 120
Graceful restart:
Restart: enabled, Helper: enabled, Restart in process: false
Reconnect time: 60000, Max neighbor reconnect time: 120000
```

```
Recovery time: 160000, Max neighbor recovery time: 240000
Traffic Engineering:
Bgp igp: disabled
Both ribs: disabled
Mpls forwarding: disabled
IGP:
Tracking igp metric: disabled
Sync session up delay: 10
Session protection:
Session protection: disabled
Session protection timeout: 0
Interface addresses advertising:
10.1.1.1
10.100.2.1
10.101.2.1
10.1.1.4
10.1.1.6
10.53.85.142
2001:db8:1000:1:2::1
2001:db8:1001:1:2::1
2001:db8:1111::1
2001:db8:abcd::128:53:85:142
fe80:1:2::1
fe80:1001:1002::1
LDP Job:
Read job time quantum: 1000, Write job time quantum: 1000
Read job loop quantum: 100, Write job loop quantum: 100
Backup inbound read job time quantum: 1000, Backup outbound read job time quantum: 1000
Backup inbound read job loop quantum: 100, Backup outbound read job loop quantum: 100
Label allocation:
Current number of LDP labels allocated: 4
Total number of LDP labels allocated: 7
Total number of LDP labels freed: 3
Total number of LDP label allocation failure: 0
Current number of labels allocated by all protocols: 4
```

RELATED DOCUMENTATION

- *no-propagate-ttl*
- *no-decrement-ttl*

show ldp overview

MPLS TTL Propagation Flexibility for LDP-signaled LSPs | **1661**

Understanding Multipoint LDP Recursive FEC

IN THIS SECTION

- How it works | **1668**

The Multipoint LDP (MLDP) Recursive Forwarding Equivalence Class (FEC) is useful in a deployment in which the intermediate routers do not have the route to reach the root node.

We've introduced the recursive opaque value as defined in RFC 6512 that helps to form MLDP point-to-multipoint (P2MP) tunnels between two autonomous systems (ASs) when the backbone has no direct route to the root node. This enhances flexibility and robustness of MLDP in complex network configurations, especially when spanning multiple autonomous systems.

To overcome the issue of establishing MLDP P2MP tunnels when the backbone has no direct route to the root node, RFC 6512 provides a solution to temporarily replace the actual root node address with the address known to the intermediate nodes, making it possible to establish the path through the network even in inter-AS scenarios involving BGP and Multipoint-to-Multipoint Label Switched Paths (MP LSPs). For a P2MP LSP, the root node address and its associated opaque value are the key components used to route MLDP messages through the MLDP control plane and create the P2MP LSP.

.

How it works

To form an MLDP tunnel, an application provides the root address and opaque value to LDP. It also informs LDP to be the egress for the particular P2MP LSP. The label advertised for P2MP FEC by the egress Label-switching router (LSR) can be decided either by LDP or the application. The LSR then finds the upstream router to the root by performing a route lookup. The labels assigned for the FEC are advertised as a label mapping message to the upstream router.

A transit LSR upon receiving a label advertisement from the downstream LSR advertises a label to the upstream LSR towards the root in the P2MP FEC. It also installs the necessary forwarding state to forward the packets when they arrive with the label it advertised. This repeats on each router till the label reaches the ingress router.

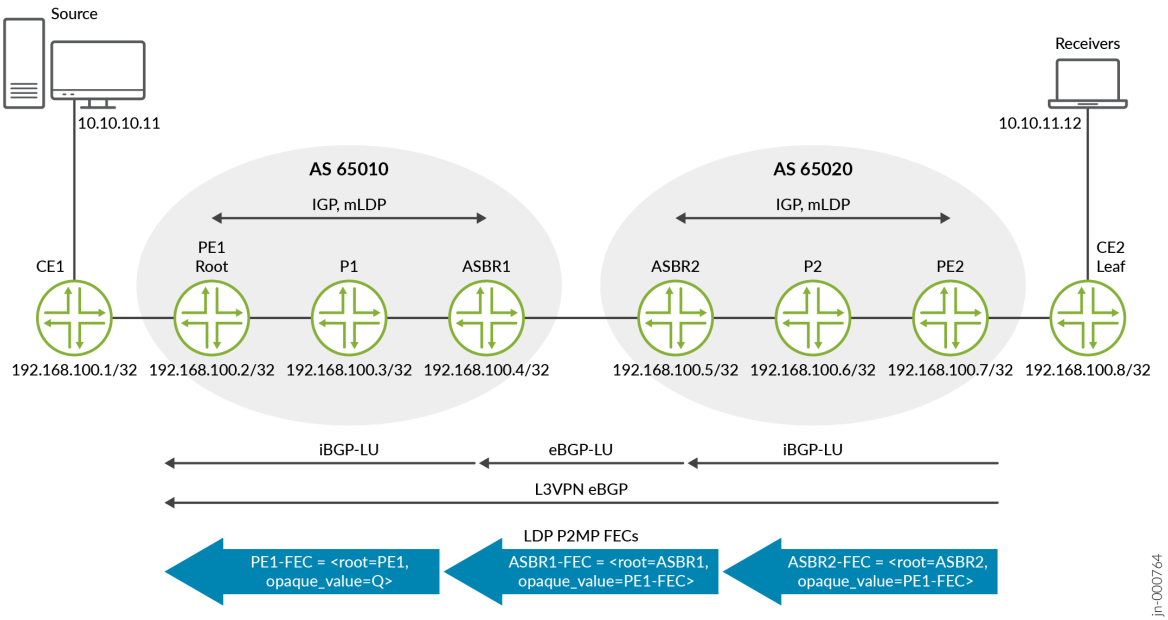
At the ingress LSR, mapping the traffic onto the P2MP LSP is determined by the application. Thus, LDP does not install any forwarding state for the P2MP LSP. However, it provides necessary information about the P2MP LSP to the application so that the application installs the forwarding state.

Figure 108 on page 1669 depicts a host system capable of sending multicast traffic and a host system capable of receiving multicast traffic.

The Provider edge (PE) and AS border router (ASBR) routers have route to the root PE1. The P (P1 and P2) routers do not have a route towards the root. The MLDP P2MP tunnel is formed from the egress to the ingress routers. When the egress router PE2 tries to send a label map message to the root PE1, it finds that the upstream router to reach the root PE1 is P2. Therefore, label map message is sent to P2. However, P2 does not have a route to the root and the upstream router for the root cannot be found. Therefore, P2 cannot send a label map message to its upstream router and hence tunnel cannot be formed.

In such topologies, recursive opaque value feature is enabled using the `set protocol ldp p2mp recursive fec` configuration statement.

Figure 108: MLDP FEC for Inter-AS with Recursive Opaque Value in MVPN Option C scenario



NOTE: The recursive P2MP comes into effect if the route to the root is an indirect BGP route.

Let's understand this in detail:

- For LDP to form a P2MP tunnel from leaf PE2 to root PE1, PE2 creates an MP FEC element with the address of PE1 as the root node address as follows: **PE1-FEC = <root=PE1, opaque_value=Q>**
- PE2 sends the FEC element as MLDP label map message to P2. However, P2 cannot use the FEC element because it does not have a route to PE1.
- With `set protocol ldp p2mp recursive fec` configuration statement configured, PE2 determines that the root node's address matches a BGP route, with ASBR2 as the protocol next-hop.
- Therefore, PE2 creates the new MP FEC element with the recursive opaque value. Within the FEC element, the root node address is the address of the protocol nexthop ASBR2 and the opaque value is a recursive opaque value that contains PE2-FEC. We refer to this FEC element as ASBR2-FEC. **ASBR2-FEC = <root=ASBR2, opaque_value=PE1-FEC>** which is **PE1-FEC = <root=PE1, opaque_value=<root=S, opaque_value=Q>>**
- LDP requests the interior routers (P1 and P2) to build an MP LSP for which the root node is ASBR2. However, the routers must not interpret the opaque value.
- When ASBR2-FEC arrives at ASBR2, ASBR2 notes that it is the identified root node and that the opaque value is a recursive opaque value. Therefore, ASBR2 reads the opaque value to find the actual root.
- Then ASBR2 does a lookup to find the route to actual root, that is PE1. As the route to PE1 is a BGP route with an indirect next hop, ASBR2 determines that it should form a LDP recursive FEC to the root PE1 with the protocol nexthop as ASBR1 and keep the received opaque value intact. **ASBR1-FEC = <root=ASBR1, opaque_value=PE1-FEC>**
- When ASBR1-FEC arrives at ASBR1, ASBR1 notes that it is the identified root node and that the opaque value is a recursive opaque value. Therefore, ASBR1 reads the opaque value to find the actual root. The ASBR1 finds the route to the root which is an IGP route. So ASBR1 replaces ASBR1-FEC with the contents of the recursive opaque value, that is, with PE1-FEC before any further processing.
- This results in PE1-FEC being sent on to P1. **PE1-FEC = <root=PE1, opaque_value=<root=S, opaque_value=Q>> P1**.
- PE1 receives the PE1-FEC and processes it like any normal LDP P2MP FEC.

NOTE: The recursive P2MP FEC is not applicable for PIM in-band signalling and static LDP P2MP tunnel.

RELATED DOCUMENTATION

p2mp (Protocols LDP)

Example: Configuring Multipoint LDP Recursive FEC

SUMMARY

IN THIS SECTION

- Example Prerequisites | **1671**
- Before You Begin | **1672**
- Functional Overview | **1672**
- Topology Overview | **1672**
- Topology Illustration | **1674**
- PE1 Configuration Steps | **1674**
- Verification | **1678**
- Appendix 1: Set Commands on All Devices | **1683**
- Appendix 2: Show Configuration Output on PE1 and ASBR2 | **1692**

NOTE: Our content testing team has validated and updated this example.

Use this configuration example to configure and verify Multipoint Label Distribution Protocol (LDP) Recursive forwarding equivalence class (FEC) in an OSPF network. This enables to form MLDP point-to-multipoint (P2MP) tunnels between two autonomous systems (ASs) when the backbone has no direct route to the root node.

TIP:

Table 34: Readability Score and Time Estimates

Reading Time	45 minutes
Configuration Time	1 hour

Example Prerequisites

Hardware requirements	MX Series routers as CE, PE, and intermediate routers.
Software requirements	Junos OS Release 23.4R1 or later running on all devices.

Before You Begin

Benefits	The Multipoint LDP (MLDP) Recursive Forwarding Equivalence Class (FEC) is useful in a deployment in which the intermediate routers do not have the route to reach the root node.
Know more	Understanding Multipoint LDP Recursive FEC.

Functional Overview

Technologies used	• Service Family:Layer 3 VPN • Protocols:BGP, IS-IS, LDP, MPLS, OSPF, PIM • Transport Tunnels: MLDP P2MP,
Primary verification tasks	• Verify LDP P2MP tunnel is formed. • Verify the formation of LDP recursive FEC • Verify the recursive opaque value functionality in the ingress and transit routers.

Topology Overview

This configuration example depicts a host system capable of sending multicast traffic and a host system capable of receiving multicast traffic. It shows two autonomous systems (ASes); AS65010 that consists of PE1, P1, and ASBR1 routers and AS65020 that consists of PE2, P2, and ASBR2 routers. All routers within an AS run OSPF as the IGP. All routers belong to area 0. The customer edge (CE) devices use EBGP peering to exchange routes with their provider edge device as part of a Layer 3 VPN service. The PE devices use IBGP to exchange IPv4 routes with the remote PE.

The Provider edge (PE2) and AS border router (ASBR2) routers have route to the root PE1. However, The P (P1 and P2) routers do not have a route towards the root. The MLDP P2MP tunnel is formed from the egress (PE2) to ingress (PE1) routers to pass the FEC element through mLDP.

Hostname	Role	Function
"CE1" on page 1684	Remote CE device.	EBGP peer to PE1 router to advertise and learn CE device loopback addresses.
"PE1 (DUT)" on page 1685	Local PE device.	The root node that intermediate routers of another AS try to reach in a scenario where there is no direct route to PE1.
"P1" on page 1686	Intermediate device P1.	Intermediate router that has no route to PE2 router. It receives PE1-FEC from ASBR1 and processes it as a normal LDP P2MP FEC.
"ASBR1" on page 1687	AS border router ASBR1	EBGP peer to ASBR2 router to advertise and learn IGP routes. The ASBR1 finds the route to the root (which is an IGP route) and replaces ASBR1-FEC with the contents of the Recursive Opaque Value (PE1-FEC) before doing any further processing.
"ASBR2" on page 1688	AS border router ASBR2	EBGP peer to ASBR1 router to advertise and learn IGP routes. Acts as root node for P2 and does a lookup to find the route to actual root (PE1). Determines the route to PE1 is a BGP route and forms a LDP Recursive FEC with ASBR1 to PE1.

(Continued)

Hostname	Role	Function
"P2" on page 1689	Intermediate device P2.	Intermediate router that has no route to PE1 router. It receives the FEC element from PE2 through mLDP. However, it cannot use the same FEC element, because it does not have route to PE1.
"PE2" on page 1690	Local PE device.	PE2 creates an MP FEC element with the address of PE1 as the root node address to form a P2MP tunnel from PE2 to root PE1.
"CE2" on page 1691	Remote CE device.	EBGP peering to PE2 router to advertise and learn CE device loopback addresses.

Topology Illustration

Figure 109: MLDP FEC for Inter-AS with Recursive Opaque Value in MVPN Option C scenario

PE1 Configuration Steps

NOTE: For complete sample configurations on PE1, see:

- "Appendix 1: Set Commands on All Devices" on page 1683
- "Appendix 2: Show Configuration Output on PE1 and ASBR2" on page 1692

This section highlights the main configuration tasks needed to configure the PE1 device for this example. The first step is common to configuring a basic Layer 3 VPN service. The following set of steps are specific to configuring recursive FEC. Both PE devices have a similar configuration, here we focus on PE1.

1. First, provision the general Layer 3 VPN:

a. Configure and number the loopback, core facing, and CE-facing interfaces for IPv4. Be sure to enable the `mpls` family on the core-facing interfaces connecting to the P devices to support MPLS switching.

b. Configure an autonomous system number.

c. Configure single area OSPF on the loopback and core-facing interfaces.

d. Configure LDP, MPLS on all the interfaces.

e. Configure the IBGP peering session to include the `inet-vpn` address family to support IPv4 layer 3 VPN.

f. Configure a VRF based routing-instance for the CE1 device. Use EBGP as the PE-CE routing protocol.

```
[edit]
set interfaces ge-0/0/0 description "CONNECTED TO CE1"
set interfaces ge-0/0/0 unit 0 family inet address 172.16.1.2/30

set interfaces ge-0/0/1 unit 0 family inet address 10.1.23.1/24
set interfaces ge-0/0/1 unit 0 family mpls

set interfaces lo0 unit 0 family inet address 192.168.100.2/32
set interfaces lo0 unit 1 family inet address 192.168.102.1/32
```

```
[edit]
set chassis network-services enhanced-ip
```

```
[edit]
set routing-instances VPN1 instance-type vrf
set routing-instances VPN1 protocols bgp group CE1-PE1 type external
set routing-instances VPN1 protocols bgp group CE1-PE1 family inet unicast
set routing-instances VPN1 protocols bgp group CE1-PE1 export EXPORT-LOCAL-ROUTES
set routing-instances VPN1 protocols bgp group CE1-PE1 peer-as 65101
set routing-instances VPN1 protocols bgp group CE1-PE1 local-as 65100
set routing-instances VPN1 protocols bgp group CE1-PE1 neighbor 172.16.1.1
set routing-instances VPN1 interface ge-0/0/0.0
set routing-instances VPN1 interface lo0.1
```

```
set routing-instances VPN1 route-distinguisher 192.168.100.2:1
set routing-instances VPN1 vrf-table-label
```

```
[edit]
set protocols bgp group INT type internal
set protocols bgp group INT local-address 192.168.100.2
set protocols bgp group INT family inet labeled-unicast resolve-vpn
set protocols bgp group INT family inet-vpn any
set protocols bgp group INT family inet-mvpn signaling
set protocols bgp group INT peer-as 65100
set protocols bgp group INT local-as 65100
set protocols bgp group INT neighbor 192.168.100.3


set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable


set protocols ldp interface all
set protocols ldp interface fxp0.0 disable

set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
```

```
[edit]

set routing-options router-id 192.168.100.2
set routing-options autonomous-system 65100
```

2. Configure the mode for C-PIM join messages to use rendezvous-point trees, and switch to the shortest-path tree after the source is known.

```
[edit]
set routing-instances VPN1 protocols mvpn mvpn-mode rpt-spt
set routing-instances VPN1 provider-tunnel ldp-p2mp
set routing-instances VPN1 vrf-target target:1:1
```

3. Configure PIM to enable PE to CE multicast routing.

```
[edit]
set routing-instances VPN1 protocols pim rp static address 192.168.100.1
```

4. Enable PIM on the interfaces connected to the CE router.

```
[edit]
set routing-instances VPN1 protocols pim interface lo0.1
set routing-instances VPN1 protocols pim interface ge-0/0/0.0
```

5. Configure dynamic selective point-to-multipoint LSP and specify the data threshold required before the new tunnel is created.

```
[edit]
set routing-instances VPN1 provider-tunnel selective group 225.1.1.1/32 source 172.16.12.1/32
ldp-p2mp
set routing-instances VPN1 provider-tunnel selective group 225.1.1.1/32 source 172.16.12.1/32
threshold-rate 1
```

6. Configure point-to-multipoint recursive LDP FEC.

```
[edit]

set protocols ldp p2mp recursive fec
```

7. Configure the routing policies.

```
[edit]

set policy-options policy-statement EXPORT-LOCAL-ROUTES term 1 from protocol direct
set policy-options policy-statement EXPORT-LOCAL-ROUTES term 1 then accept
```

Verification

IN THIS SECTION

- Verify Recursive FEC in the LDP overview information. | 1678
- Verify the Advertised Label | 1680
- Verify P2MP FEC Information | 1682
- Verify P2MP Path Information | 1683

Command	Verification Task
show ldp overview	Verify `Recursive fec` enabled in the LDP overview information.
show ldp database	Verify the labels advertised point-to-multipoint binding information in the LDP database.
show ldp p2mp path	Verify LDP P2MP FEC information.
show ldp p2mp fec	Verify LDP P2MP LSPs.

Verify Recursive FEC in the LDP overview information.

Purpose

To confirm `recursive FEC` is enabled and is displayed in PE1 LDP overview information.

Action

From operational mode, enter the `show ldp overview` command.

```
user@PE1 show ldp overview
Instance: master
  Reference count: 3
  Router ID: 192.168.100.2
  LDP inet: enabled
  Transport preference: IPv4
  Message id: 33
  Configuration sequence: 12
```

```
Deaggregate: disabled
Explicit null: disabled
IPv6 tunneling: disabled
Strict targeted hellos: disabled
Loopback if added: yes
Route preference: 9
P2MP max branches: 4096
Unicast transit LSP chaining: disabled
P2MP transit LSP chaining: disabled
Transit LSP statistics based on route statistics: disabled
LDP route acknowledgement: enabled
BGP export: enabled
No TTL propagate: disabled
LDP mtu discovery: disabled
LDP SR Mapping Client: disabled
Capabilities enabled: p2mp, make-before-break
Egress FEC capabilities enabled: entropy-label-capability
Downstream unsolicited Sessions:
  Operational: 1
  Retention: liberal
  Control: ordered
Auto targeted sessions:
  Auto targeted: disabled
  Dynamic tunnel session count: 0
P2MP:
  Recursive route: disabled
  Recursive fec: enabled
  No rsvp tunneling: disabled
Timers:
  Keepalive interval: 10, Keepalive timeout: 30
  Link hello interval: 5, Link hello hold time: 15
  Targeted hello interval: 15, Targeted hello hold time: 45
  Label withdraw delay: 60, Make before break timeout: 30
  Make before break switchover delay: 3
  Link protection timeout: 120
Graceful restart:
  Restart: disabled, Helper: enabled, Restart in process: false
  Reconnect time: 60000, Max neighbor reconnect time: 120000
  Recovery time: 160000, Max neighbor recovery time: 240000
Traffic Engineering:
  Bgp igp: disabled
  Both ribs: disabled
  Mpls forwarding: disabled
```

```
  IGP:
    Tracking igp metric: disabled
    Sync session up delay: 10
  Session protection:
    Session protection: disabled
    Session protection timeout: 0
  Interface addresses advertising:
    10.1.23.1
    192.168.100.2
  LDP Job:
    Read job time quantum: 1000, Write job time quantum: 1000
    Read job loop quantum: 100, Write job loop quantum: 100
    Backup inbound read job time quantum: 1000, Backup outbound read job time quantum: 1000
    Backup inbound read job loop quantum: 100, Backup outbound read job loop quantum: 100
  Label allocation:
    Current number of LDP labels allocated: 2
    Total number of LDP labels allocated: 51
    Total number of LDP labels freed: 49
    Total number of LDP label allocation failure: 0
    Current number of labels allocated by all protocols: 0
```

Meaning

The output confirms that recursive fec is enabled on PE1.

Verify the Advertised Label

Purpose

Verify the labels advertised from ASBR1 and received by PE1.

Action

From operational mode, enter the `show ldp database` command.

On ASBR1

```
user@ASBR1 show ldp database
Input label database, 192.168.100.4:0--192.168.100.5:0
Labels received: 5
  Label     Prefix
```

```
    145      192.168.100.4/32
      3      192.168.100.5/32
    139      192.168.100.6/32
    140      192.168.100.7/32
    144      P2MP root-addr 192.168.100.2, lsp-id 16777220 next-root:192.168.100.4

Output label database, 192.168.100.4:0--192.168.100.5:0
Labels advertised: 4
  Label     Prefix
      3      192.168.100.4/32
    164      192.168.100.5/32
    159      192.168.100.2/32
    158      192.168.100.3/32

Input label database, 192.168.100.4:0--192.168.100.3:0
Labels received: 3
  Label     Prefix
    125      192.168.100.4/32
     88      192.168.100.2/32
      3      192.168.100.3/32

Output label database, 192.168.100.4:0--192.168.100.3:0
Labels advertised: 5
  Label     Prefix
      3      192.168.100.4/32
    164      192.168.100.5/32
    159      192.168.100.2/32
    158      192.168.100.3/32
    163      P2MP root-addr 192.168.100.2, lsp-id 16777220
```

On PE1

```
user@PE1 show ldp database
Input label database, 192.168.100.2:0--192.168.100.3:0
Labels received: 4
  Label     Prefix
    125      192.168.100.4/32
     88      192.168.100.2/32
      3     192.168.100.3/32
    129      P2MP root-addr 192.168.100.2, lsp-id 16777220

Output label database, 192.168.100.2:0--192.168.100.3:0
```

```
Labels advertised: 3
  Label     Prefix
    107      192.168.100.4/32
      3      192.168.100.2/32
    104      192.168.100.3/32
```

Meaning

You can see label **3** advertised from ASBR1 and label **3** received by PE1.

Verify P2MP FEC Information

Purpose

Verify the LDP P2MP FEC information from ASBR2.

Action

From operational mode, enter the `show ldp p2mp fec extensive` command.

```
user@PE2 show ldp p2mp fec extensive
LDP P2MP FECs:
 P2MP root-addr 192.168.100.2, lsp-id 16777220 next-root: 192.168.100.5
  Fec type: Egress (Active)
  Label: 52, Reference count:    1
```

```
user@ASBR2 show ldp p2mp fec extensive
LDP P2MP FECs:
 P2MP root-addr 192.168.100.2, lsp-id 16777220
  Fec type: Transit (Active)
  Label: 144, Reference count:    1
```

Meaning

The output shows that PE2 forms a LDP recursive FEC to find the route to actual root (PE1).

Verify P2MP Path Information

Purpose

Verify the LDP P2MP path information from PE2.

Action

From operational mode, enter the `show ldp p2mp path extensive` command.

```
user@PE2 show ldp p2mp path extensive
P2MP path type: Transit/Egress
  Output Session (label): 192.168.100.6:0 (52) (Primary)
  Egress label: 52
  Attached FECs:  P2MP root-addr 192.168.100.2, lsp-id 16777220 next-root:192.168.100.5 (Active)
  Address: 0x76f69e0, Reference count: 2
```

Meaning

The output shows PE2 with the FEC element, in which the root node address is the address of protocol nexthop ASBR2.

Appendix 1: Set Commands on All Devices

IN THIS SECTION

- CE1 | 1684
- PE1 | 1685
- P1 | 1686
- ASBR1 | 1687
- ASBR2 | 1688
- P2 | 1689
- PE2 | 1690
- CE2 | 1691

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level.

NOTE: We use logical systems to represent the source and receiver for this example.

CE1

```
set system host-name CE1-source
set logical-systems source interfaces lt-0/0/0 unit 12 encapsulation ethernet
set logical-systems source interfaces lt-0/0/0 unit 12 peer-unit 11
set logical-systems source interfaces lt-0/0/0 unit 12 family inet address 172.16.12.1/30
set logical-systems source interfaces lo0 unit 12 family inet address 192.168.12.12/32
set logical-systems source protocols ospf area 0.0.0.0 interface all
set logical-systems source protocols ospf area 0.0.0.0 interface lo0.12 passive
set logical-systems source protocols pim rp static address 192.168.100.1
set logical-systems source protocols pim interface lt-0/0/0.12
set chassis fpc 0 pic 0 tunnel-services
set chassis network-services enhanced-ip
set interfaces lt-0/0/0 description "CONNECTED TO source"
set interfaces lt-0/0/0 unit 11 encapsulation ethernet
set interfaces lt-0/0/0 unit 11 peer-unit 12
set interfaces lt-0/0/0 unit 11 family inet address 172.16.12.2/30
set interfaces ge-0/0/1 description "CONNECTED TO PE1"
set interfaces ge-0/0/1 unit 0 family inet address 172.16.1.1/30
set interfaces lo0 unit 0 family inet address 192.168.100.1/32
set policy-options policy-statement EXPORT-LOCAL-ROUTES term 1 from protocol direct
set policy-options policy-statement EXPORT-LOCAL-ROUTES term 1 then accept
set routing-options router-id 192.168.100.1
set routing-options autonomous-system 65101
set protocols bgp group CE1-PE1 type external
set protocols bgp group CE1-PE1 family inet unicast
set protocols bgp group CE1-PE1 export EXPORT-LOCAL-ROUTES
set protocols bgp group CE1-PE1 peer-as 65100
set protocols bgp group CE1-PE1 local-as 65101
set protocols bgp group CE1-PE1 neighbor 172.16.1.2
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols pim rp local family inet address 192.168.100.1
set protocols pim interface lo0.0
```

```
set protocols pim interface ge-0/0/1.0
set protocols pim interface lt-0/0/0.11
```

PE1

```
set system host-name PE1
set chassis network-services enhanced-ip
set interfaces ge-0/0/0 description "CONNECTED TO CE1"
set interfaces ge-0/0/0 unit 0 family inet address 172.16.1.2/30
set interfaces ge-0/0/1 description "CONNECTED TO P1"
set interfaces ge-0/0/1 unit 0 family inet address 10.1.23.1/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.100.2/32
set interfaces lo0 unit 1 family inet address 192.168.102.1/32
set policy-options policy-statement EXPORT-LOCAL-ROUTES term 1 from protocol direct
set policy-options policy-statement EXPORT-LOCAL-ROUTES term 1 then accept
set routing-instances VPN1 instance-type vrf
set routing-instances VPN1 protocols bgp group CE1-PE1 type external
set routing-instances VPN1 protocols bgp group CE1-PE1 family inet unicast
set routing-instances VPN1 protocols bgp group CE1-PE1 export EXPORT-LOCAL-ROUTES
set routing-instances VPN1 protocols bgp group CE1-PE1 peer-as 65101
set routing-instances VPN1 protocols bgp group CE1-PE1 local-as 65100
set routing-instances VPN1 protocols bgp group CE1-PE1 neighbor 172.16.1.1
set routing-instances VPN1 protocols mvpn mvpn-mode rpt-spt
set routing-instances VPN1 protocols pim rp static address 192.168.100.1
set routing-instances VPN1 protocols pim interface lo0.1 mode sparse
set routing-instances VPN1 protocols pim interface ge-0/0/0.0 mode sparse
set routing-instances VPN1 interface ge-0/0/0.0
set routing-instances VPN1 interface lo0.1
set routing-instances VPN1 route-distinguisher 192.168.100.2:1
set routing-instances VPN1 vrf-target target:1:1
set routing-instances VPN1 vrf-table-label
set routing-instances VPN1 provider-tunnel ldp-p2mp
set routing-instances VPN1 provider-tunnel selective group 225.1.1.1/32 source 172.16.12.1/32
ldp-p2mp
set routing-instances VPN1 provider-tunnel selective group 225.1.1.1/32 source 172.16.12.1/32
threshold-rate 51
set routing-options router-id 192.168.100.2
set routing-options autonomous-system 65100
set protocols bgp group INT type internal
set protocols bgp group INT local-address 192.168.100.2
```

```
set protocols bgp group INT family inet labeled-unicast resolve-vpn
set protocols bgp group INT family inet-vpn any
set protocols bgp group INT family inet-mvpn signaling
set protocols bgp group INT peer-as 65100
set protocols bgp group INT local-as 65100
set protocols bgp group INT neighbor 192.168.100.3
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols ldp p2mp recursive fec
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

P1

```
set system host-name P1
set chassis network-services enhanced-ip
set interfaces ge-0/0/0 unit 0 family inet address 10.1.23.2/24
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 10.1.34.1/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.100.3/32
set routing-options router-id 192.168.100.3
set routing-options autonomous-system 65100
set protocols bgp group INT type internal
set protocols bgp group INT local-address 192.168.100.3
set protocols bgp group INT family inet labeled-unicast resolve-vpn
set protocols bgp group INT family inet-vpn any
set protocols bgp group INT family inet-mvpn signaling
set protocols bgp group INT cluster 192.168.100.3
set protocols bgp group INT local-as 65100
set protocols bgp group INT neighbor 192.168.100.2
set protocols bgp group INT neighbor 192.168.100.4
set protocols bgp group eBGP_PE_PE type external
set protocols bgp group eBGP_PE_PE multihop ttl 25
set protocols bgp group eBGP_PE_PE local-address 192.168.100.3
set protocols bgp group eBGP_PE_PE family inet-vpn any
set protocols bgp group eBGP_PE_PE family inet-mvpn signaling
set protocols bgp group eBGP_PE_PE neighbor 192.168.100.6 peer-as 65200
set protocols ldp interface all
```

```
set protocols ldp interface fxp0.0 disable
set protocols ldp p2mp recursive route
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

ASBR1

```
set system host-name ASBR1
set chassis network-services enhanced-ip
set interfaces ge-0/0/0 unit 0 family inet address 10.1.34.2/24
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 10.1.45.1/24
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.100.4/32
set interfaces lo0 unit 0 family iso address 49.0001.0040.0400.4004.00
set policy-options policy-statement To-ASBR2 term 1 from route-filter 192.168.100.2/32 exact
set policy-options policy-statement To-ASBR2 term 1 from route-filter 192.168.100.3/32 exact
set policy-options policy-statement To-ASBR2 term 1 then accept
set policy-options policy-statement To-ASBR2 term 2 then reject
set policy-options policy-statement from-direct term 1 from interface lo0.0
set policy-options policy-statement from-direct term 1 then accept
set policy-options policy-statement from-ospf term 1 from protocol ospf
set policy-options policy-statement from-ospf term 1 then accept
set routing-options router-id 192.168.100.4
set routing-options autonomous-system 65100
set protocols bgp group INT type internal
set protocols bgp group INT local-address 192.168.100.4
set protocols bgp group INT family inet labeled-unicast resolve-vpn
set protocols bgp group INT peer-as 65100
set protocols bgp group INT local-as 65100
set protocols bgp group INT neighbor 192.168.100.3
set protocols bgp group eBGP_ASBR type external
set protocols bgp group eBGP_ASBR multihop ttl 2
set protocols bgp group eBGP_ASBR local-address 192.168.100.4
set protocols bgp group eBGP_ASBR family inet labeled-unicast resolve-vpn
set protocols bgp group eBGP_ASBR export from-ospf
set protocols bgp group eBGP_ASBR export from-direct
set protocols bgp group eBGP_ASBR export To-ASBR2
set protocols bgp group eBGP_ASBR neighbor 192.168.100.5 peer-as 65200
```

```
set protocols isis interface ge-0/0/1.0 level 2 disable
set protocols isis interface lo0.0
set protocols isis level 2 disable
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols ldp p2mp recursive fec
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

ASBR2

```
set system host-name ASBR2
set chassis network-services enhanced-ip
set interfaces ge-0/0/0 unit 0 family inet address 10.1.45.2/24
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 10.1.56.1/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.100.5/32
set interfaces lo0 unit 0 family iso address 49.0001.0050.0500.5005.00
set policy-options policy-statement To-ASBR1 term 1 from route-filter 192.168.100.7/32 exact
set policy-options policy-statement To-ASBR1 term 1 from route-filter 192.168.100.6/32 exact
set policy-options policy-statement To-ASBR1 term 1 then accept
set policy-options policy-statement To-ASBR1 term 2 then reject
set policy-options policy-statement from-direct term 1 from interface lo0.0
set policy-options policy-statement from-direct term 1 then accept
set policy-options policy-statement from-ospf term 1 from protocol ospf
set policy-options policy-statement from-ospf term 1 then accept
set routing-options router-id 192.168.100.5
set routing-options autonomous-system 65200
set protocols bgp group INT type internal
set protocols bgp group INT local-address 192.168.100.5
set protocols bgp group INT family inet labeled-unicast resolve-vpn
set protocols bgp group INT peer-as 65200
set protocols bgp group INT local-as 65200
set protocols bgp group INT neighbor 192.168.100.6
set protocols bgp group eBGP_ASBR type external
set protocols bgp group eBGP_ASBR multihop ttl 2
set protocols bgp group eBGP_ASBR local-address 192.168.100.5
set protocols bgp group eBGP_ASBR family inet labeled-unicast resolve-vpn
set protocols bgp group eBGP_ASBR export from-ospf
```

```
set protocols bgp group eBGP_ASBR export from-direct
set protocols bgp group eBGP_ASBR export To-ASBR1
set protocols bgp group eBGP_ASBR neighbor 192.168.100.4 peer-as 65100
set protocols isis interface ge-0/0/0.0 level 2 disable
set protocols isis interface lo0.0
set protocols isis level 2 disable
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols ldp p2mp recursive fec
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

P2

```
set system host-name P2
set chassis network-services enhanced-ip
set interfaces ge-0/0/0 unit 0 family inet address 10.1.56.2/24
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 10.1.67.1/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.100.6/32
set routing-options router-id 192.168.100.6
set routing-options autonomous-system 65200
set protocols bgp group INT type internal
set protocols bgp group INT local-address 192.168.100.6
set protocols bgp group INT family inet labeled-unicast resolve-vpn
set protocols bgp group INT family inet-vpn any
set protocols bgp group INT family inet-mvpn signaling
set protocols bgp group INT cluster 192.168.100.6
set protocols bgp group INT local-as 65200
set protocols bgp group INT neighbor 192.168.100.5
set protocols bgp group INT neighbor 192.168.100.7
set protocols bgp group eBGP_PE_PE type external
set protocols bgp group eBGP_PE_PE multihop ttl 25
set protocols bgp group eBGP_PE_PE local-address 192.168.100.6
set protocols bgp group eBGP_PE_PE family inet-vpn any
set protocols bgp group eBGP_PE_PE family inet-mvpn signaling
set protocols bgp group eBGP_PE_PE neighbor 192.168.100.3 peer-as 65100
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols ldp p2mp recursive route
```

```
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

PE2

```
set system host-name PE2
set chassis network-services enhanced-ip
set interfaces ge-0/0/0 unit 0 family inet address 10.1.67.2/24
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 description "CONNECTED TO CE2"
set interfaces ge-0/0/1 unit 0 family inet address 172.16.2.2/30
set interfaces lo0 unit 0 family inet address 192.168.100.7/32
set interfaces lo0 unit 1 family inet address 192.168.100.17/32
set policy-options policy-statement EXPORT-LOCAL-ROUTES term 1 from protocol direct
set policy-options policy-statement EXPORT-LOCAL-ROUTES term 1 then accept
set routing-instances VPN2 instance-type vrf
set routing-instances VPN2 protocols bgp group PE2-CE2 type external
set routing-instances VPN2 protocols bgp group PE2-CE2 family inet unicast
set routing-instances VPN2 protocols bgp group PE2-CE2 export EXPORT-LOCAL-ROUTES
set routing-instances VPN2 protocols bgp group PE2-CE2 peer-as 65201
set routing-instances VPN2 protocols bgp group PE2-CE2 local-as 65200
set routing-instances VPN2 protocols bgp group PE2-CE2 neighbor 172.16.2.1
set routing-instances VPN2 protocols mvpn mvpn-mode rpt-spt
set routing-instances VPN2 protocols pim rp static address 192.168.100.1
set routing-instances VPN2 protocols pim interface lo0.1 mode sparse
set routing-instances VPN2 protocols pim interface ge-0/0/1.0 mode sparse
set routing-instances VPN2 protocols pim interface all
set routing-instances VPN2 interface ge-0/0/1.0
set routing-instances VPN2 interface lo0.1
set routing-instances VPN2 route-distinguisher 192.168.100.17:1
set routing-instances VPN2 vrf-target target:1:1
set routing-instances VPN2 vrf-table-label
set routing-options router-id 192.168.100.7
set routing-options autonomous-system 65200
set protocols bgp group INT type internal
set protocols bgp group INT local-address 192.168.100.7
set protocols bgp group INT family inet labeled-unicast resolve-vpn
set protocols bgp group INT family inet-vpn any
set protocols bgp group INT family inet-mvpn signaling
set protocols bgp group INT peer-as 65200
```

```
set protocols bgp group INT local-as 65200
set protocols bgp group INT neighbor 192.168.100.6
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols ldp p2mp recursive fec
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

CE2

```
set system host-name CE2-receiver
set logical-systems receiver interfaces lt-0/0/0 unit 22 encapsulation ethernet
set logical-systems receiver interfaces lt-0/0/0 unit 22 peer-unit 21
set logical-systems receiver interfaces lt-0/0/0 unit 22 family inet address 172.16.22.22/30
set logical-systems receiver interfaces lo0 unit 22 family inet address 192.168.22.22/32
set logical-systems receiver protocols ospf area 0.0.0.0 interface all
set chassis fpc 0 pic 0 tunnel-services
set chassis network-services enhanced-ip
set interfaces ge-0/0/0 description "CONNECTED TO PE2"
set interfaces ge-0/0/0 unit 0 family inet address 172.16.2.1/30
set interfaces lt-0/0/0 description "CONNECTED FROM CE2 TO receiver"
set interfaces lt-0/0/0 unit 21 encapsulation ethernet
set interfaces lt-0/0/0 unit 21 peer-unit 22
set interfaces lt-0/0/0 unit 21 family inet address 172.16.22.21/30
set interfaces lo0 unit 0 family inet address 192.168.100.8/32
set policy-options policy-statement EXPORT-LOCAL-ROUTES term 1 from protocol direct
set policy-options policy-statement EXPORT-LOCAL-ROUTES term 1 then accept
set routing-options router-id 192.168.100.8
set routing-options autonomous-system 65201
set protocols bgp group CE2-PE2 type external
set protocols bgp group CE2-PE2 family inet unicast
set protocols bgp group CE2-PE2 export EXPORT-LOCAL-ROUTES
set protocols bgp group CE2-PE2 peer-as 65200
set protocols bgp group CE2-PE2 local-as 65201
set protocols bgp group CE2-PE2 neighbor 172.16.2.2
set protocols igmp interface lt-0/0/0.21 static group 225.1.1.1
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

```
set protocols pim rp static address 192.168.100.1
set protocols pim interface all
set protocols pim interface fxp0.0 disable
set protocols sap listen 225.1.1.1 port 5000
```

Appendix 2: Show Configuration Output on PE1 and ASBR2

IN THIS SECTION

- The Complete PE1 configuration in Curly Brace Format | **1692**
- The Complete ASBR2 configuration in Curly Brace Format | **1696**

The Complete PE1 configuration in Curly Brace Format

```
user@PE1# show | no-more
system {
    host-name PE1;
}
interfaces {
    ge-0/0/0 {
        description "CONNECTED TO CE1";
        unit 0 {
            family inet {
                address 172.16.1.2/30;
            }
        }
    }
    ge-0/0/1 {
        description "CONNECTED TO P1";
        unit 0 {
            family inet {
                address 10.1.23.1/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
```

```
            family inet {
                address 192.168.100.2/32;
            }
        }
        unit 1 {
            family inet {
                address 192.168.102.1/32;
            }
        }
    }
}
policy-options {
    policy-statement EXPORT-LOCAL-ROUTES {
        term 1 {
            from protocol direct;
            then accept;
        }
    }
}
routing-instances {
    VPN1 {
        instance-type vrf;
        protocols {
            bgp {
                group CE11-PE1 {
                    type external;
                    family inet {
                        unicast;
                    }
                    export EXPORT-LOCAL-ROUTES;
                    peer-as 65101;
                    local-as 65100;
                    neighbor 172.16.1.1;
                }
            }
            mvpn {
                mvpn-mode {
                    rpt-spt;
                }
            }
            pim {
                rp {
                    static {
```

```
                        address 192.168.100.1;
                    }
                }
                interface lo0.1;
                interface ge-0/0/0.0;
            }
        }
        interface ge-0/0/0.0;
        interface lo0.1;
        route-distinguisher 192.168.100.2:1;
        vrf-target target:1:1;
        vrf-table-label;
        provider-tunnel {
            ldp-p2mp;
            selective {
                group 225.1.1.1/32 {
                    source 172.16.12.1/32 {
                        ldp-p2mp;
                        threshold-rate 1;
                    }
                }
            }
        }
    }
}
routing-options {
    router-id 192.168.100.2;
    autonomous-system 65100;
    nonstop-routing;
}
protocols {
    bgp {
        group INT {
            type internal;
            local-address 192.168.100.2;
            family inet {
                labeled-unicast {
                    resolve-vpn;
                }
            }
            family inet-vpn {
                any;
            }
```

```
            family inet-mvpn {
                signaling;
            }
            export from-direct;
            peer-as 65100;
            local-as 65100;
            neighbor 192.168.100.3;
        }
    }
    ldp {
        interface all;
        interface fxp0.0 {
            disable;
        }
        p2mp {
            recursive {
                fec;
            }
        }
    }
    mpls {
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
    ospf {
        area 0.0.0.0 {
            interface lo0.0 {
                passive;
            }
            interface all;
            interface fxp0.0 {
                disable;
            }
        }
    }
}
```

The Complete ASBR2 configuration in Curly Brace Format

```
user@ASBR2# show | no-more
system {
    host-name ASBR2;
}
interfaces {
    ge-0/0/0 {
        unit 0 {
            family inet {
                address 10.1.45.2/24;
            }
            family iso;
            family mpls;
        }
    }
    ge-0/0/1 {
        unit 0 {
            family inet {
                address 10.1.56.1/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 192.168.100.5/32;
            }
            family iso {
                address 49.0001.0060.0600.6006.00;
            }
        }
    }
}
policy-options {
    policy-statement NHP {
        term 1 {
            from protocol bgp;
            then {
                next-hop self;
                accept;
```

```
            }
        }
    }
    policy-statement To-ASBR1 {
        term 1 {
            from {
                route-filter 192.168.100.7/32 exact;
                route-filter 192.168.100.6/32 exact;
            }
            then accept;
        }
        term 2 {
            then reject;
        }
    }
    policy-statement from-direct {
        term 1 {
            from interface lo0.0;
            then accept;
        }
    }
    policy-statement from-ospf {
        term 1 {
            from protocol ospf;
            then accept;
        }
    }
}
routing-options {
    router-id 192.168.100.5;
    autonomous-system 65200;
    nonstop-routing;
}
protocols {
    bgp {
        group INT {
            type internal;
            local-address 192.168.100.5;
            family inet {
                labeled-unicast {
                    resolve-vpn;
                }
            }
```

```
            peer-as 65200;
            local-as 65200;
            neighbor 192.168.100.6 {
                export NHP;
            }
        }
        group eBGP_ASBR1-ASBR2 {
            type external;
            local-address 192.168.100.5;
            family inet {
                labeled-unicast {
                    resolve-vpn;
                }
            }
            export [ from-ospf from-direct To-ASBR1 ];
            neighbor 192.168.100.4 {
                multihop;
                peer-as 65100;
            }
        }
    }
    isis {
        interface ge-0/0/0.0 {
            level 2 disable;
        }
        interface lo0.0 {
            level 1 {
                passive;
            }
        }
    }
    ldp {
        interface all;
        interface fxp0.0 {
            disable;
        }
        p2mp {
            recursive {
                fec;
            }
        }
    }
    ospf {
```

```
        traffic-engineering;
        area 0.0.0.0 {
            interface ge-0/0/1.0;
            interface lo0.0 {
                passive;
            }
        }
    }
}
```

PART 6

MPLS Traffic Engineering

- Configuring MPLS Traffic Engineering | **1701**

CHAPTER 12

Configuring MPLS Traffic Engineering

IN THIS CHAPTER

- MPLS Traffic Engineering Configuration | **1701**
- Color-Based Traffic Engineering Configuration | **1772**
- DiffServ-Aware Traffic Engineering Configuration | **1826**

MPLS Traffic Engineering Configuration

IN THIS SECTION

- MPLS and Traffic Engineering | **1702**
- MPLS Traffic Engineering and Signaling Protocols Overview | **1702**
- Traffic Engineering Capabilities | **1703**
- Components of Traffic Engineering | **1703**
- Configuring Traffic Engineering for LSPs | **1704**
- Enabling Interarea Traffic Engineering | **1708**
- Enabling Inter-AS Traffic Engineering for LSPs | **1709**
- Packet Forwarding Component | **1712**
- Offline Path Planning and Analysis | **1715**
- Flexible LSP Calculation and Configuration | **1715**
- Link-State Distribution Using BGP Overview | **1716**
- Example: Configuring Link State Distribution Using BGP | **1731**
- Configuring Link State Distribution Using BGP | **1756**
- BGP Classful Transport Planes Overview | **1760**
- Improving Traffic Engineering Database Accuracy with RSVP PathErr Messages | **1769**

MPLS and Traffic Engineering

Traffic engineering allows you to control the path that data packets follow, bypassing the standard routing model, which uses routing tables. Traffic engineering moves flows from congested links to alternate links that would not be selected by the automatically computed destination-based shortest path. With traffic engineering, you can:

- Make more efficient use of expensive long-haul fibers.
- Control how traffic is rerouted in the face of single or multiple failures.
- Classify critical and regular traffic on a per-path basis.

The core of the traffic engineering design is based on building label-switched paths (LSPs) among routers. An LSP is connection-oriented, like a virtual circuit in Frame Relay or ATM. LSPs are not reliable: Packets entering an LSP do not have delivery guarantees, although preferential treatment is possible. LSPs also are similar to unidirectional tunnels in that packets entering a path are encapsulated in an envelope and switched across the entire path without being touched by intermediate nodes. LSPs provide fine-grained control over how packets are forwarded in a network. To provide reliability, an LSP can use a set of primary and secondary paths.

LSPs can be configured for BGP traffic only (traffic whose destination is outside of an autonomous system [AS]). In this case, traffic within the AS is not affected by the presence of LSPs. LSPs can also be configured for both BGP and interior gateway protocol (IGP) traffic; therefore, both intra-AS and inter-AS traffic is affected by the LSPs.

MPLS Traffic Engineering and Signaling Protocols Overview

Traffic engineering facilitates efficient and reliable network operations while simultaneously optimizing network resources and traffic performance. Traffic engineering provides the ability to move traffic flow away from the shortest path selected by the interior gateway protocol (IGP) to a potentially less congested physical path across a network. To support traffic engineering, besides source routing, the network must do the following:

- Compute a path at the source by taking into account all the constraints, such as bandwidth and administrative requirements.
- Distribute the information about network topology and link attributes throughout the network once the path is computed.
- Reserve network resources and modify link attributes.

When transit traffic is routed through an IP network, MPLS is often used to engineer its passage. Although the exact path through the transit network is of little importance to either the sender or the receiver of the traffic, network administrators often want to route traffic more efficiently between certain source and destination address pairs. By adding a short label with specific routing instructions to each packet, MPLS switches packets from router to router through the network rather than forwarding

packets based on next-hop lookups. The resulting routes are called *label-switched paths (LSPs)*. LSPs control the passage of traffic through the network and speed traffic forwarding.

You can create LSPs manually, or through the use of signaling protocols. Signaling protocols are used within an MPLS environment to establish LSPs for traffic across a transit network. Junos OS supports two signaling protocols—LDP and the Resource Reservation Protocol (RSVP).

MPLS traffic engineering uses the following components:

- MPLS LSPs for packet forwarding
- IGP extensions for distributing information about the network topology and link attributes
- Constrained Shortest Path First (CSPF) for path computation and path selection
- RSVP extensions to establish the forwarding state along the path and to reserve resources along the path

Junos OS also supports traffic engineering across different OSPF regions.

Traffic Engineering Capabilities

The task of mapping traffic flows onto an existing physical topology is called *traffic engineering*. Traffic engineering provides the ability to move traffic flow away from the shortest path selected by the interior gateway protocol (IGP) and onto a potentially less congested physical path across a network.

Traffic engineering provides the capabilities to do the following:

- Route primary paths around known bottlenecks or points of congestion in the network.
- Provide precise control over how traffic is rerouted when the primary path is faced with single or multiple failures.
- Provide more efficient use of available aggregate bandwidth and long-haul fiber by ensuring that subsets of the network do not become overutilized while other subsets of the network along potential alternate paths are underutilized.
- Maximize operational efficiency.
- Enhance the traffic-oriented performance characteristics of the network by minimizing packet loss, minimizing prolonged periods of congestion, and maximizing throughput.
- Enhance statistically bound performance characteristics of the network (such as loss ratio, delay variation, and transfer delay) required to support a multiservices Internet.

Components of Traffic Engineering

In the Junos® operating system (OS), traffic engineering is implemented with MPLS and RSVP. Traffic engineering is composed of four functional components:

- "Packet Forwarding Component" on page 1712
- Information Distribution Component
- Path Selection Component
- Signaling Component

Configuring Traffic Engineering for LSPs

IN THIS SECTION

- Using LSPs for Both BGP and IGP Traffic Forwarding | **1704**
- Using LSPs for Forwarding in Virtual Private Networks | **1705**
- Using RSVP and LDP Routes for Forwarding but Not Route Selection | **1706**
- Advertising the LSP Metric in Summary LSAs | **1707**

When you configure an LSP, a host route (a 32-bit mask) is installed in the ingress router toward the egress router; the address of the host route is the destination address of the LSP. The `bgp` option for the `traffic engineering` statement at the `[edit protocols mpls]` hierarchy level is enabled by default (you can also explicitly configure the `bgp` option), allowing only BGP to use LSPs in its route calculations. The other `traffic-engineering` statement options allow you to alter this behavior in the master routing instance. This functionality is not available for specific routing instances. Also, you can enable only one of the `traffic-engineering` statement options (`bgp`, `bgp-igp`, `bgp-igp-both-ribs`, or `mpls-forwarding`) at a time.

NOTE: Enabling or disabling any of the `traffic-engineering` statement options causes all the MPLS routes to be removed and then reinserted into the routing tables.

You can configure OSPF and traffic engineering to advertise the LSP metric in summary link-state advertisements (LSAs) as described in the section "Advertising the LSP Metric in Summary LSAs" on page 1707.

The following sections describe how to configure traffic engineering for LSPs:

Using LSPs for Both BGP and IGP Traffic Forwarding

You can configure BGP and the IGPs to use LSPs for forwarding traffic destined for egress routers by including the `bgp-igp` option for the `traffic-engineering` statement. The `bgp-igp` option causes all inet.3 routes to be moved to the inet.0 routing table.

On the ingress router, include bgp-igp option for the traffic-engineering statement:

```
traffic-engineering bgp-igp;
```

You can include this statement at the following hierarchy levels:

- [edit protocols mpls]
- [edit logical-systems *logical-system-name* protocols mpls]

> **NOTE**: The bgp-igp option for the traffic-engineering statement cannot be configured for VPN). VPNs require that routes be in the inet.3 routing table.

Using LSPs for Forwarding in Virtual Private Networks

VPNs require that routes remain in the inet.3 routing table to function properly. For VPNs, configure the bgp-igp-both-ribs option of the traffic-engineering statement to cause BGP and the IGPs to use LSPs for forwarding traffic destined for egress routers. The bgp-igp-both-ribs option installs the ingress routes in both the inet.0 routing table (for IPv4 unicast routes) and the inet.3 routing table (for MPLS path information).

On the ingress router, include the traffic-engineering bgp-igp-both-ribs statement:

```
traffic-engineering bgp-igp-both-ribs;
```

You can include this statement at the following hierarchy levels:

- [edit protocols mpls]
- [edit logical-systems *logical-system-name* protocols mpls]

When you use the bgp-igp-both-ribs statement, the routes from the inet.3 table get copied into the inet.0 table. The copied routes are LDP-signaled or RSVP-signaled, and are likely to have a inferior preference than other routes in inet.0. Routes with a inferior preference are more likely to be chosen as the active routes. This can be a problem because routing policies only act upon active routes. To prevent this problem, use the mpls-forwarding option instead.

> **NOTE**: LSPs with the numerically lowest preference value is chosen as the preferred route.

For example:

```
user@host# show protocols mpls
label-switched-path lsp1 {
    to 192.168.4.4;
    preference 1000;
}
label-switched-path lsp2 {
    to 192.168.4.4;
    preference 1001;
}

user@host# run show route table inet.3

inet.3: 2 destinations, 3 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

198.168.4.4/32         *[RSVP/1000/1] 00:17:23, metric 30
                     >  to 192.168.2.18 via ge-0/0/1.0, label-switched-path lsp1
                        to 192.168.5.5 via ge-0/0/2.0, label-switched-path Bypass-
>192.168.2.18->192.168.3.3
                     [RSVP/1001/1] 00:17:23, metric 30
                     >  to 192.168.2.18 via ge-0/0/1.0, label-switched-path lsp2
                        to 192.168.5.5 via ge-0/0/2.0, label-switched-path Bypass-
>192.168.2.18->192.168.3.3
```

LSP with a preference value of 1000 is superior and hence is preferred over the LSP with a preference value of 1001.

Using RSVP and LDP Routes for Forwarding but Not Route Selection

If you configure the `bgp-igp` or `bgp-igp-both-ribs` options for the `traffic-engineering` statement, high-priority LSPs can supersede IGP routes in the inet.0 routing table. IGP routes might no longer be redistributed since they are no longer the active routes.

If you configure the `mpls-forwarding` option for the `traffic-engineering` statement, LSPs are used for forwarding but are excluded from route selection. These routes are added to both the inet.0 and inet.3 routing tables. LSPs in the inet.0 routing table are given an inferior preference when the active route is selected. However, LSPs in the inet.3 routing table are given a normal preference and are therefore used for selecting forwarding next hops.

When you activate the mpls-forwarding option, routes whose state is ForwardingOnly are preferred for forwarding even if their preference is inferior than that of the currently active route. To examine the state of a route, execute a show route detail command.

To use LSPs for forwarding but exclude them from route selection, include the mpls-forwarding option for the traffic-engineering statement:

```
traffic-engineering mpls-forwarding;
```

You can include this statement at the following hierarchy levels:

- [edit protocols mpls]
- [edit logical-systems *logical-system-name* protocols mpls]

When you configure the mpls-forwarding option, IGP shortcut routes are copied to the inet.0 routing table only.

Unlike the bgp-igp-both-ribs option, the mpls-forwarding option allows you to use the LDP-signaled and RSVP-signaled routes for forwarding, and keep the BGP and IGP routes active for routing purposes so that routing policies can act upon them.

For example, suppose a router is running BGP and it has a BGP route of 10.10.10.1/32 that it needs to send to another BGP speaker. If you use the bgp-igp-both-ribs option, and your router also has a label-switched-path (LSP) to 10.10.10.1, the MPLS route for 10.10.10.1 becomes active in the inet.0 routing table. This prevents your router from advertising the 10.10.10.1 route to the other BGP router. On the other hand, if you use the mpls-forwarding option instead of the bgp-igp-both-ribs option, the 10.10.10.1/32 BGP route is advertised to the other BGP speaker, and the LSP is still used to forward traffic to the 10.10.10.1 destination.

Advertising the LSP Metric in Summary LSAs

You can configure MPLS and OSPF to treat an LSP as a link. This configuration allows other routers in the network to use this LSP. To accomplish this goal, you need to configure MPLS and OSPF traffic engineering to advertise the LSP metric in summary LSAs.

For MPLS, include the traffic-engineering bgp-igp and label-switched-path statements:

```
traffic-engineering bgp-igp;
label-switched-path lsp-name {
    to address;
}
```

You can include these statements at the following hierarchy levels:

- [edit protocols mpls]
- [edit logical-systems *logical-system-name* protocols mpls]

For OSPF, include the lsp-metric-into-summary statement:

```
lsp-metric-into-summary;
```

You can include this statement at the following hierarchy levels:

- [edit protocols ospf traffic-engineering shortcuts]
- [edit logical-systems *logical-system-name* protocols ospf traffic-engineering shortcuts]

For more information about OSPF traffic engineering, see the Junos OS Routing Protocols Library for Routing Devices.

Enabling Interarea Traffic Engineering

The Junos OS can signal a contiguous traffic-engineered LSP across multiple OSPF areas. The LSP signaling must be done using either nesting or contiguous signaling, as described in RFC 4206, *Label-Switched Paths (LSP) Hierarchy with Generalized Multi-Protocol Label Switching (GMPLS) Traffic Engineering (TE)*. However, contiguous signaling support is limited to just basic signaling. Reoptimization is not supported with contiguous signaling.

The following describes some of the interarea traffic engineering features:

- Interarea traffic engineering can be enabled when the loose-hop area border routers (ABRs) are configured on the ingress router using CSPF for the Explicit Route Object (ERO) calculation within an OSPF area. ERO expansion is completed on the ABRs.
- Interarea traffic engineering can be enabled when CSPF is enabled, but without ABRs specified in the LSP configuration on the ingress router (ABRs can be automatically designated).
- Differentiated Services (DiffServ) traffic engineering is supported as long as the class type mappings are uniform across multiple areas.

To enable interarea traffic engineering, include the expand-loose-hop statement in the configuration for each LSP transit router:

```
expand-loose-hop;
```

You can include this statement at the following hierarchy levels:

- [edit protocols mpls]

- [edit logical-systems *logical-system-name* protocols mpls]

Enabling Inter-AS Traffic Engineering for LSPs

IN THIS SECTION

- Inter-AS Traffic Engineering Requirements | **1709**
- Inter-AS Traffic Engineering Limitations | **1710**
- Configuring OSPF Passive TE Mode | **1711**

Generally, traffic engineering is possible for LSPs that meet the following conditions:

- Both ends of the LSP are in the same OSPF area or at the same IS-IS level.
- The two ends of the LSP are in different OSPF areas within the same autonomous system (AS). LSPs that end in different IS-IS levels are not supported.
- The two ends of an explicit-path LSP are in different OSPF ASs and the autonomous system border routers (ASBRs) are configured statically as the loose hops supported on the explicit-path LSP. For more information, see "Configuring Explicit-Path LSPs" on page 675.

Without statically defined ASBRs on LSPs, traffic engineering is not possible between one routing domain, or AS, and another. However, when the ASs are under the control of single service provider, it is possible in some cases to have traffic engineered LSPs span the ASs and dynamically discover the OSPF ASBRs linking them (IS-IS is not supported with this feature).

Inter-AS traffic engineered LSPs are possible as long as certain network requirements are met, none of the limiting conditions apply, and OSPF passive mode is configured with EBGP. Details are provided in the following sections:

Inter-AS Traffic Engineering Requirements

The proper establishment and functioning of inter-AS traffic engineered LSPs depend on the following network requirements, all of which must be met:

- All ASs are under control of a single service provider.
- OSPF is used as the routing protocol within each AS, and EBGP is used as the routing protocol between the ASs.
- ASBR information is available inside each AS.

- EBGP routing information is distributed by OSPF, and an IBGP full mesh is in place within each AS.
- Transit LSPs are *not* configured on the inter-AS links, but *are* configured between entry and exit point ASBRs on each AS.
- The EBGP link between ASBRs in different ASs is a direct link and must be configured as a passive traffic engineering link under OSPF. The remote link address itself, not the loopback or any other link address, is used as the remote node identifier for this passive link. For more information about OSPF passive traffic engineering mode configuration, see "Configuring OSPF Passive TE Mode" on page 1711.

In addition, the address used for the remote node of the OSPF passive traffic engineering link must be the same as the address used for the EBGP link. For more information about OSPF and BGP in general, see the Junos OS Routing Protocols Library for Routing Devices.

Inter-AS Traffic Engineering Limitations

Only LSP hierarchical, or nested, signaling is supported for inter-AS traffic engineered LSPs. Only point-to-point LSPs are supported (there is no point-to-multipoint support).

In addition, the following limitations apply. Any one of these conditions is sufficient to render inter-AS traffic engineered LSPs impossible, even if the above requirements are met.

- The use of multihop BGP is not supported.
- The use of policers or topologies that prevent BGP routes from being known inside the AS is not supported.
- Multiple ASBRs on a LAN between EBGP peers are not supported. Only one ASBR on a LAN between EBGP peers is supported (others ASBRs can exist on the LAN, but cannot be advertised).
- Route reflectors or policies that hide ASBR information or prevent ASBR information from being distributed inside the ASs are not supported.
- Bidirectional LSPs are not supported (LSPs are unidirectional from the traffic engineering perspective).
- Topologies with both inter-AS and intra-AS paths to the same destination are not supported.

In addition, several features that are routine with all LSPs are not supported with inter-AS traffic engineering:

- Admin group link colors are not supported.
- Secondary standby is not supported.
- Reoptimization is not supported.

- Crankback on transit routers is not supported.
- Diverse path calculation is not supported.
- Graceful restart is not supported.

These lists of limitations or unsupported features with inter-AS traffic engineered LSPs are not exhaustive.

Configuring OSPF Passive TE Mode

Ordinarily, interior routing protocols such as OSPF are not run on links between ASs. However, for inter-AS traffic engineering to function properly, information about the inter-AS link, in particular, the address on the remote interface, must be made available inside the AS. This information is not normally included either in EBGP reachability messages or in OSPF routing advertisements.

To flood this link address information within the AS and make it available for traffic engineering calculations, you must configure OSPF passive mode for traffic engineering on each inter-AS interface. You must also supply the remote address for OSPF to distribute and include in the traffic engineering database.

To configure OSPF passive mode for traffic engineering on an inter-AS interface, include the `passive` statement for the link at the `[edit protocols ospf area` *`area-id`* `interface` *`interface-name`*`]` hierarchy level:

```
passive {
    traffic-engineering {
        remote-node-id ip-address;     /* IP address at far end of inter-AS link */
    }
}
```

OSPF must be properly configured on the router. The following example configures the inter-AS link `so-1/1/0` to distribute traffic engineering information with OSPF within the AS. The remote IP address is `192.168.207.2`.

```
[edit protocols ospf area 0.0.0.0]
interface so-1/1/0 {
    unit 0 {
        passive {
            traffic-engineering {
                remote-node-id 192.168.207.2;
            }
        }
```

```
    }
}
```

Packet Forwarding Component

IN THIS SECTION

- Packet Forwarding Based on Label Swapping | **1712**
- How a Packet Traverses an MPLS Backbone | **1713**
- Information Distribution Component | **1713**
- Path Selection Component | **1713**
- Signaling Component | **1714**

The packet forwarding component of the Junos traffic engineering architecture is MPLS, which is responsible for directing a flow of IP packets along a predetermined path across a network. This path is called a *label-switched path (LSP)*. LSPs are simplex; that is, the traffic flows in one direction from the head-end (ingress) router to a tail-end (egress) router. Duplex traffic requires two LSPs: one LSP to carry traffic in each direction. An LSP is created by the concatenation of one or more label-switched hops, allowing a packet to be forwarded from one router to another across the MPLS domain.

When an ingress router receives an IP packet, it adds an MPLS header to the packet and forwards it to the next router in the LSP. The labeled packet is forwarded along the LSP by each router until it reaches the tail end of the LSP, the egress router. At this point the MPLS header is removed, and the packet is forwarded based on Layer 3 information such as the IP destination address. The value of this scheme is that the physical path of the LSP is not limited to what the IGP would choose as the shortest path to reach the destination IP address.

Packet Forwarding Based on Label Swapping

The packet forwarding process at each router is based on the concept of label swapping. This concept is similar to what occurs at each Asynchronous Transfer Mode (ATM) switch in a permanent virtual circuit (PVC). Each MPLS packet carries a 4-byte encapsulation header that contains a 20-bit, fixed-length label field. When a packet containing a label arrives at a router, the router examines the label and copies it as an index to its MPLS forwarding table. Each entry in the forwarding table contains an interface-inbound label pair mapped to a set of forwarding information that is applied to all packets arriving on the specific interface with the same inbound label.

How a Packet Traverses an MPLS Backbone

This section describes how an IP packet is processed as it traverses an MPLS backbone network.

At the entry edge of the MPLS backbone, the IP header is examined by the ingress router. Based on this analysis, the packet is classified, assigned a label, encapsulated in an MPLS header, and forwarded toward the next hop in the LSP. MPLS provides a high degree of flexibility in the way that an IP packet can be assigned to an LSP. For example, in the Junos traffic engineering implementation, all packets arriving at the ingress router that are destined to exit the MPLS domain at the same egress router are forwarded along the same LSP.

Once the packet begins to traverse the LSP, each router uses the label to make the forwarding decision. The MPLS forwarding decision is made independently of the original IP header: the incoming interface and label are used as lookup keys into the MPLS forwarding table. The old label is replaced with a new label, and the packet is forwarded to the next hop along the LSP. This process is repeated at each router in the LSP until the packet reaches the egress router.

When the packet arrives at the egress router, the label is removed and the packet exits the MPLS domain. The packet is then forwarded based on the destination IP address contained in the packet's original IP header according to the traditional shortest path calculated by the IP routing protocol.

Information Distribution Component

Traffic engineering requires detailed knowledge about the network topology as well as dynamic information about network loading. To implement the information distribution component, simple extensions to the IGPs are defined. Link attributes are included as part of each router's link-state advertisement. IS-IS extensions include the definition of new type length values (TLVs), whereas OSPF extensions are implemented with opaque link-state advertisements (LSAs). The standard flooding algorithm used by the link-state IGPs ensures that link attributes are distributed to all routers in the routing domain. Some of the traffic engineering extensions to be added to the IGP link-state advertisement include maximum link bandwidth, maximum reserved link bandwidth, current bandwidth reservation, and link coloring.

Each router maintains network link attributes and topology information in a specialized traffic engineering database. The traffic engineering database is used exclusively for calculating explicit paths for the placement of LSPs across the physical topology. A separate database is maintained so that the subsequent traffic engineering computation is independent of the IGP and the IGP's link-state database. Meanwhile, the IGP continues its operation without modification, performing the traditional shortest-path calculation based on information contained in the router's link-state database.

Path Selection Component

After network link attributes and topology information are flooded by the IGP and placed in the traffic engineering database, each ingress router uses the traffic engineering database to calculate the paths for

its own set of LSPs across the routing domain. The path for each LSP can be represented by either a strict or loose explicit route. An explicit route is a preconfigured sequence of routers that should be part of the physical path of the LSP. If the ingress router specifies all the routers in the LSP, the LSP is said to be identified by a strict explicit route. If the ingress router specifies only some of the routers in the LSP, the LSP is described as a loose explicit route. Support for strict and loose explicit routes allows the path selection process to be given broad latitude whenever possible, but to be constrained when necessary.

The ingress router determines the physical path for each LSP by applying a Constrained Shortest Path First (CSPF) algorithm to the information in the traffic engineering database. CSPF is a shortest-path-first algorithm that has been modified to take into account specific restrictions when the shortest path across the network is calculated. Input into the CSPF algorithm includes:

- Topology link-state information learned from the IGP and maintained in the traffic engineering database
- Attributes associated with the state of network resources (such as total link bandwidth, reserved link bandwidth, available link bandwidth, and link color) that are carried by IGP extensions and stored in the traffic engineering database
- Administrative attributes required to support traffic traversing the proposed LSP (such as bandwidth requirements, maximum hop count, and administrative policy requirements) that are obtained from user configuration

As CSPF considers each candidate node and link for a new LSP, it either accepts or rejects a specific path component based on resource availability or whether selecting the component violates user policy constraints. The output of the CSPF calculation is an explicit route consisting of a sequence of router addresses that provides the shortest path through the network that meets the constraints. This explicit route is then passed to the signaling component, which establishes the forwarding state in the routers along the LSP.

Signaling Component

An LSP is not known to be workable until it is actually established by the signaling component. The signaling component, which is responsible for establishing LSP state and distributing labels, relies on a number of extensions to RSVP:

- The Explicit Route object allows an RSVP path message to traverse an explicit sequence of routers that is independent of conventional shortest-path IP routing. The explicit route can be either strict or loose.
- The Label Request object permits the RSVP path message to request that intermediate routers provide a label binding for the LSP that it is establishing.
- The Label object allows RSVP to support the distribution of labels without changing its existing mechanisms. Because the RSVP Resv message follows the reverse path of the RSVP path message, the Label object supports the distribution of labels from downstream nodes to upstream nodes.

Offline Path Planning and Analysis

Despite the reduced management effort resulting from online path calculation, an offline planning and analysis tool is still required to optimize traffic engineering globally. Online calculation takes resource constraints into account and calculates one LSP at a time. The challenge with this approach is that it is not deterministic. The order in which LSPs are calculated plays a critical role in determining each LSP's physical path across the network. LSPs that are calculated early in the process have more resources available to them than LSPs calculated later in the process because previously calculated LSPs consume network resources. If the order in which the LSPs are calculated is changed, the resulting set of physical paths for the LSPs also can change.

An offline planning and analysis tool simultaneously examines each link's resource constraints and the requirements of each LSP. Although the offline approach can take several hours to complete, it performs global calculations, compares the results of each calculation, and then selects the best solution for the network as a whole. The output of the offline calculation is a set of LSPs that optimizes utilization of network resources. After the offline calculation is completed, the LSPs can be established in any order because each is installed according to the rules for the globally optimized solution.

Flexible LSP Calculation and Configuration

Traffic engineering involves mapping traffic flow onto a physical topology. You can determine the paths online using constraint-based routing. Regardless of how the physical path is calculated, the forwarding state is installed across the network through RSVP.

The Junos OS supports the following ways to route and configure an LSP:

- You can calculate the full path for the LSP offline and individually configure each router in the LSP with the necessary static forwarding state. This is analogous to the way some Internet service providers (ISPs) configure their IP-over-ATM cores.

- You can calculate the full path for the LSP offline and statically configure the ingress router with the full path. The ingress router then uses RSVP as a dynamic signaling protocol to install a forwarding state in each router along the LSP.

- You can rely on constraint-based routing to perform dynamic online LSP calculation. You configure the constraints for each LSP; then the network itself determines the path that best meets those constraints. Specifically, the ingress router calculates the entire LSP based on the constraints and then initiates signaling across the network.

- You can calculate a partial path for an LSP offline and statically configure the ingress router with a subset of the routers in the path; then you can permit online calculation to determine the complete path.

  For example, consider a topology that includes two east-west paths across the United States: one in the north through Chicago and one in the south through Dallas. If you want to establish an LSP between a router in New York and one in San Francisco, you can configure the partial path for the

LSP to include a single loose-routed hop of a router in Dallas. The result is an LSP routed along the southern path. The ingress router uses CSPF to compute the complete path and RSVP to install the forwarding state along the LSP.

- You can configure the ingress router with no constraints whatsoever. In this case, normal IGP shortest-path routing is used to determine the path of the LSP. This configuration does not provide any value in terms of traffic engineering. However, it is easy and might be useful in situations when services such as virtual private networks (VPNs) are needed.

In all these cases, you can specify any number of LSPs as backups for the primary LSP, thus allowing you to combine more than one configuration approach. For example, you might explicitly compute the primary path offline, set the secondary path to be constraint-based, and have the tertiary path be unconstrained. If a circuit on which the primary LSP is routed fails, the ingress router notices the outage from error notifications received from a downstream router or by the expiration of RSVP soft-state information. Then the router dynamically forwards traffic to a hot-standby LSP or calls on RSVP to create a forwarding state for a new backup LSP.

Link-State Distribution Using BGP Overview

IN THIS SECTION

- Role of an Interior Gateway Protocol | **1716**
- Limitations of an Interior Gateway Protocol | **1717**
- Need for Spanning Link-State Distribution | **1718**
- Using BGP as a Solution | **1718**
- Supported and Unsupported Features | **1725**
- BGP Link-State Extensions for Source Packet Routing in Networking (SPRING) | **1726**
- Verifying NLRI Node Learned Through BGP with OSPF as IGP | **1729**
- Verifying the Prefix NLRI Learned Through BGP with OSPF as IGP | **1730**

Role of an Interior Gateway Protocol

An interior gateway protocol (IGP) is a type of protocol used for exchanging routing information between devices within an autonomous system (AS). Based on the method of computing the best path to a destination, the IGPs are divided into two categories:

- Link-state protocols—Advertise information about the network topology (directly connected links and the state of those links) to all routers using multicast addresses and triggered routing updates until all the routers running the link-state protocol have identical information about the internetwork.

The best path to a destination is calculated based on constraints such as maximum delay, minimum available bandwidth, and resource class affinity.

OSPF and IS-IS are examples of link-state protocols.

- Distance vector protocols—Advertise complete routing table information to directly connected neighbors using a broadcast address. The best path is calculated based on the number of hops to the destination network.

  RIP is an example of a distance vector protocol.

As the name implies, the role of an IGP is to provide routing connectivity within or internal to a given routing domain. A routing domain is a set of routers under common administrative control that share a common routing protocol. An AS can consist of multiple routing domains, where IGP functions to advertise and learn network prefixes (routes) from neighboring routers to build a route table that ultimately contains entries for all sources advertising reachability for a given prefix. IGP executes a route selection algorithm to select the best path between the local router and each destination, and provides full connectivity among the routers making up a routing domain.

In addition to advertising internal network reachability, IGPs are often used to advertise routing information that is external to that IGP's routing domain through a process known as route redistribution. Route redistribution is the process of exchanging routing information among distinct routing protocols to tie multiple routing domains together when intra-AS connectivity is desired.

Limitations of an Interior Gateway Protocol

While each individual IGP has its own advantages and limitations, the biggest limitations of IGP in general are performance and scalability.

IGPs are designed to handle the task of acquiring and distributing network topology information for traffic engineering purposes. While this model has served well, IGPs have inherent scaling limitations when it comes to distributing large databases. IGPs can autodetect neighbors, with which they acquire intra-area network topology information. However, the link-state database or a traffic engineering database has the scope of a single area or AS, thereby limiting applications, such as end-to-end traffic engineering, the benefit of having external visibility to make better decisions.

For label-switched networks, such as MPLS and Generalized MPLS (GMPLS), most existing traffic engineering solutions work in a single routing domain. These solutions do not work when a route from the ingress node to the egress node leaves the routing area or AS of the ingress node. In such cases, the path computation problem becomes complicated because of the unavailability of the complete routing information throughout the network. This is because service providers usually choose not to leak routing information beyond the routing area or AS for scalability constraints and confidentiality concerns.

Need for Spanning Link-State Distribution

One of the limitations of IGP is its inability to span link-state distribution outside a single area or AS. However, spanning link-state information acquired by an IGP across multiple areas or ASs has the following needs:

- LSP path computation—This information is used to compute the path for MPLS LSPs across multiple routing domains, for example an inter-area TE LSP.
- External path computing entities—External path computing entities, such as Application Layer Traffic Optimization (ALTO) and Path Computation Elements (PCE), perform path computations based on the network topology and current state of connections within the network, including traffic engineering information. This information is typically distributed by IGPs within the network.

  However, because the external path computing entities cannot extract this information from the IGPs, they perform network monitoring to optimize network services.

Using BGP as a Solution

Overview

To meet the needs for spanning link-state distribution across multiple domains, an exterior gateway protocol (EGP) is required to collect link-state and traffic engineering information from an IGP area, share it with external component, and use it for computing paths for interdomain MPLS LSPs.

BGP is a standardized EGP designed to exchange routing and reachability information between autonomous systems (ASs). BGP is a proven protocol that has better scaling properties because it can distribute millions of entries (for example, VPN prefixes) in a scalable fashion. BGP is the only routing protocol in use today that is suited to carry all of the routes in the Internet. This is largely because BGP runs on top of TCP and can make use of TCP flow control. In contrast, the internal gateway protocols (IGPs) do not have flow control. When IGPs have too much route information, they begin to churn. When BGP has a neighboring speaker that is sending information too quickly, BGP can throttle down the neighbor by delaying TCP acknowledgments.

Another benefit of BGP is that it uses type, length, value (TLV) tuples and network layer reachability information (NLRI) that provide seemingly endless extensibility without the need for the underlying protocol to be altered.

The distribution of link-state information across domains is regulated using policies to protect the interests of the service provider. This requires a control over the topology distribution using policies. BGP with its implemented policy framework serves well in the interdomain route distribution. In Junos OS, BGP is completely policy driven. The operator must explicitly configure neighbors to peer with and explicitly accept routes into BGP. Furthermore, routing policy is used to filter and modify routing information. Thus, routing policies provide complete administrative control over the routing tables.

Although, within an AS, both IGP-TE and BGP-TE provide the same set of information, BGP-TE has better scaling characteristics that are inherited from the standard BGP protocol. This makes BGP-TE a more scalable choice for acquiring multi-area/multi-AS topology information.

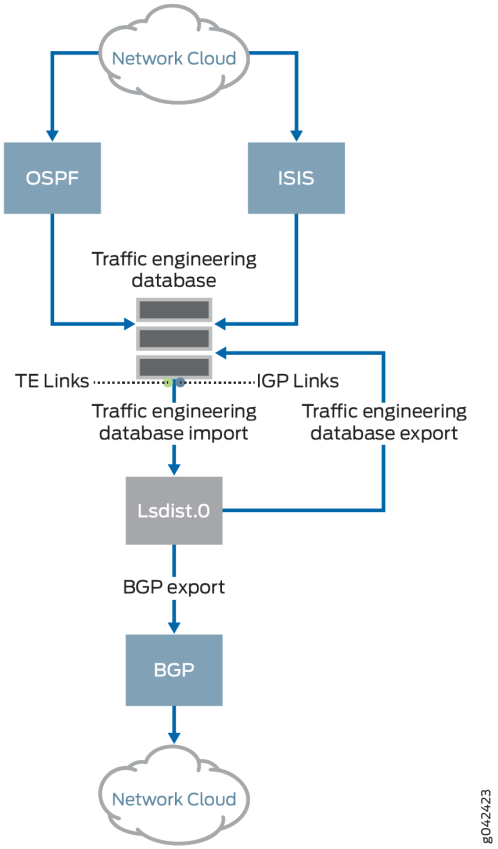
By using BGP as a solution, the IGP-acquired information is used for distribution into BGP. The ISPs can selectively expose this information with other ISPs, service providers, and content distribution networks (CDNs) through normal BGP peering. This allows for aggregation of the IGP-acquired information across multiple areas and ASs, such that an external path computing entity can access the information by passively listening to a route reflector.

Implementation

In Junos OS, the IGPs install topology information into a database called the traffic engineering database. The traffic engineering database contains the aggregated topology information. To install IGP topology information into traffic engineering database, use the `set igp-topology` configuration statement at the `[edit protocols isis traffic-engineering]` and `[edit protocols ospf traffic-engineering]` hierarchy levels. The mechanism to distribute link-state information using BGP includes the process of advertising the traffic engineering database into BGP-TE (import), and installing entries from BGP-TE into the traffic engineering database (export).

You can configure IS-IS traffic engineering to store IPv6 information in the traffic engineering database (TED) in addition to IPv4 addresses. BGP-LS distributes this information as routes from the traffic engineering database to the lsdist.0 routing table using the traffic engineering database import policies. These routes are advertised to BGP-TE peers as network layer reachability information (NLRI) with IPv6 router ID type, length, and value (TLV). With addition of IPv6 information, you can benefit from obtaining the complete network topology into the traffic engineering database.

Figure 110: Junos OS Implementation of BGP Link-State Distribution



BGP-LS NLRI and Confederation ID

Junos OS enables BGP Link State (BGP-LS) network layer reachability information (NLRI) to carry the confederation ID in TLV 512 when BGP confederation is enabled. The NLRI carries the confederation ID along with the member autonomous system number (AS number) in TLV 517 as defined in RFC 9086. The Junos OS traffic engineering database module makes necessary changes to encode confederation ID and member AS number in TLV 512 and TLV 517 respectively, while originating the BGP-LS NLRI (which is injected into lsdist.0 routing table). In releases before Junos OS Release 23.1R1, BGP-LS NLRI carries only the member AS number in TLV 512 and the confederation ID is not encoded in the lsdist.0 routing table.

Traffic Engineering Database Import

To advertise the traffic engineering database into BGP-TE, the link and node entries in the traffic engineering database are converted in the form of routes. These converted routes are then installed by the traffic engineering database on behalf of the corresponding IGP, into a user-visible routing table

called lsdist.0, on conditions subject to route policies. The procedure of leaking entries from the traffic engineering database into lsdist.0 is called traffic engineering database import as illustrated in Figure 110 on page 1720.

There are polices to govern the traffic engineering database import process. By default, no entries are leaked from the traffic engineering database into the lsdist.0 table.

The traffic engineering database installs interior gateway protocol (IGP) topology information in addition to RSVP-TE topology information in the lsdist.0 routing table as illustrated in Figure 110 on page 1720. You can monitor both IGP and traffic engineering topology information. The BGP-LS reads IGP entries from lsdist.0 and advertises these entries to the BGP peers. To import IGP topology information into BGP-LS from lsdist.0, use the `set bgp-ls` configuration statement at the `[edit protocols mpls traffic-engineering database import igp-topology]` hierarchy level.

Traffic Engineering Database Export

BGP can be configured to export or advertise routes from the lsdist.0 table, subject to policy. This is common for any kind of route origination in BGP. In order to advertise BGP-TE into the traffic engineering database, BGP needs to be configured with the BGP-TE address family, and an export policy that selects routes for redistribution into BGP.

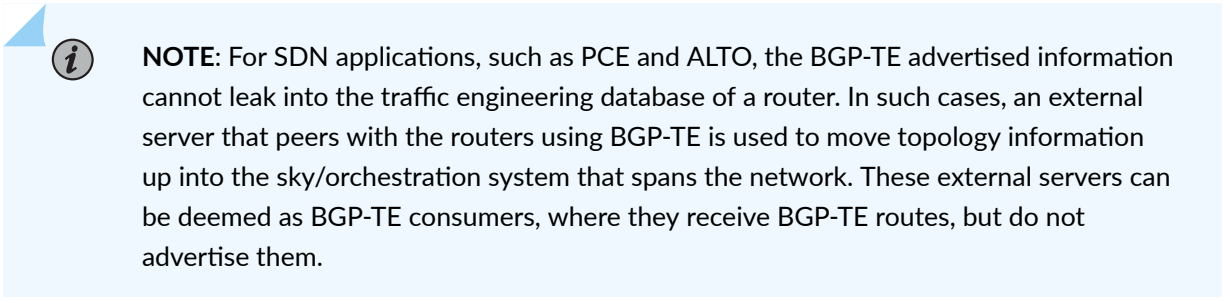
BGP then propagates these routes like any other NLRI. BGP peers that have the BGP-TE family configured and negotiated receive BGP-TE NLRIs. BGP stores the received BGP-TE NLRIs in the form of routes in the lsdist.0 table, which is the same table that stores locally originated BGP-TE routes. The BGP-installed routes in lsdist.0 are then distributed to other peers like any other route. Thus, the standard route selection procedure applies to BGP-TE NLRIs received from multiple speakers.

To achieve interdomain TE, the routes in lsdist.0 are leaked into the traffic engineering database through a policy. This process is called traffic engineering database export as illustrated in Figure 110 on page 1720.

There are polices to govern the traffic engineering database export process. By default, no entries are leaked from the lsdist.0 table into the traffic engineering database.

You can distribute the traffic engineering (TE) policies that originate from the segment routing protocol to the traffic engineering database (TED) and into the BGP link-state as routes. BGP link-state collects the information related to the TE policies, so that the external controllers can perform actions such as path-computation, re-optimization, and network visualization within and across domains.

Configure `set protocols source-packet-routing traffic-engineering database` to allow the segment routing (SR) policies to be stored in TED.



NOTE: For SDN applications, such as PCE and ALTO, the BGP-TE advertised information cannot leak into the traffic engineering database of a router. In such cases, an external server that peers with the routers using BGP-TE is used to move topology information up into the sky/orchestration system that spans the network. These external servers can be deemed as BGP-TE consumers, where they receive BGP-TE routes, but do not advertise them.

Assigning Credibility Values

Once the entries are installed in the traffic engineering database, the BGP-TE learned information is made available for CSPF path computation. The traffic engineering database uses a protocol preference scheme that is based on credibility values. A protocol with a higher credibility value is preferred over a protocol with a lower credibility value. BGP-TE has the capability to advertise information learned from multiple protocols at the same time, and so in addition to the IGP-installed entries in the traffic engineering database, there can be BGP-TE installed entries that correspond to more than one protocol. The traffic engineering database export component creates a traffic engineering database protocol and credibility level for each protocol that BGP-TE supports. These credibility values are configurable in the CLI.

The credibility order for the BGP-TE protocols is as follows:

- Unknown—80
- OSPF—81
- ISIS Level 1—82
- ISIS Level 2—83
- Static—84
- Direct—85

Cross-Credibility Path Computation

After you assign credibility values, each credibility level is treated as an individual plane. The Constrained Shorted Path First algorithm starts with the highest assigned credibility to the lowest, finding a path within that credibility level.

With BGP-TE, it is essential to compute paths across credibility levels to compute inter-AS paths. For example, different credibility settings are seen on a device from area 0 that computes the path through area 1, because area 0 entries are installed by OSPF, and area 1 entries are installed by BGP-TE.

To enable path computation across credibility levels, include the `cross-credibility-cspf` statement at the `edit protocols mpls`, `[edit protocols mpls label-switched-path` *`lsp-name`*`]`, and `[edit protocols rsvp]` hierarchy levels. At the `[edit protocols rsvp]` hierarchy level, enabling `cross-credibility-cspf` impacts bypass LSPs and loose hop expansion in transit.

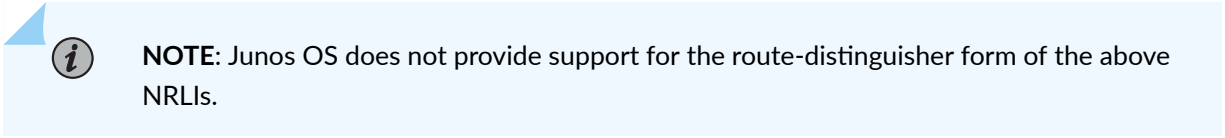
Configuring `cross-credibility-cspf` enables path computation across credibility levels using the Constrained Shortest Path First algorithm, wherein the constraint is not performed on a credibility-by-credibility basis, but as a single constraint ignoring the assigned credibility values.

BGP-TE NLRIs and TLVs

Like other BGP routes, BGP-TE NLRIs can also be distributed through a route reflector that speaks BGP-TE NLRI. Junos OS implements the route reflection support for the BGP-TE family.

The following is a list of supported NLRIs:

- Link NLRI
- Node NLRI
- IPv4 Prefix NLRI (receive and propagate)
- IPv6 Prefix NLRI (receive and propagate)
- TE policy NLRI



NOTE: Junos OS does not provide support for the route-distinguisher form of the above NRLIs.

The following is a list of supported fields in link and node NLRIs:

- Protocol-ID—NLRI originates with the following protocol values:
  - ISIS-L1
  - ISIS-L2
  - OSPF
  - SPRING-TE
- Identifier—This value is configurable. By default, the identifier value is set to `0`.
- Local/Remote node descriptor—These include:
  - Autonomous system

    - BGP-LS Identifier—This value is configurable. By default, the BGP-LS identifier value is set to `0`
    - Area-ID
    - IGP router-ID
- Link descriptors (Only for link NLRI)—This includes:
    - Link Local/Remote Identifiers
    - IPv4 interface address
    - IPv4 neighbor address
    - IPv6 neighbor/interface address—The IPv6 neighbor and interface addresses are not originated, but only stored and propagated when received.
    - Multi-topology ID—This value is not originated, but stored and propagated when received.

The following is a list of supported LINK_STATE attribute TLVs:

- Link attributes:
    - Administrative group
    - Max link bandwidth
    - Max reservable bandwidth
    - Unreserved bandwidth
    - TE default metric
    - SRLG
    - The following TLVs, which are not originated, but only stored and propagated when received:
        - Opaque link attributes
        - MPLS protocol mask
        - Metric
        - Link protection type
        - Link name attribute
- Node attributes:
    - IPv4 Router-ID

- Node flag bits—Only the overload bit is set.
- The following TLVs, which are not originated, but only stored and propagated when received:
  - Multi-topology
  - OSPF-specific node properties
  - Opaque node properties
  - Node name
  - IS-IS area identifier
  - IPv6 Router-ID
- Prefix attributes—These TLVs are stored and propagated like any other unknown TLVs.

Supported and Unsupported Features

Junos OS supports the following features with link-state distribution using BGP:

- Advertisement of multiprotocol assured forwarding capability
- Transmission and reception of node and link-state BGP and BGP-TE NLRIs
- Nonstop active routing for BGP-TE NLRIs
- Policies

Junos OS does **not** support the following functionality for link-state distribution using BGP:

- Aggregated topologies, links, or nodes
- Route distinguisher support for BGP-TE NLRIs
- Multi-topology identifiers
- Multi-instance identifiers (excluding the default instance ID 0)
- Advertisement of the link and node area TLV
- Advertisement of MPLS signaling protocols
- Importing node and link information with overlapping address

BGP Link-State Extensions for Source Packet Routing in Networking (SPRING)

The BGP link-state address family is extended to distribute the source packet routing in networking (SPRING) topology information to software-defined networking (SDN) controllers. BGP typically learns the link-state information from IGP and distributes it to BGP peers. Besides BGP, the SDN controller can get link-state information directly from IGP if the controller is a part of an IGP domain. However, BGP link-state distribution provides a scalable mechanism to export the topology information. BGP link-state extensions for SPRING is supported on interdomain networks.

Source Packet Routing in Networking (SPRING)

SPRING is a control-plane architecture that enables an ingress router to steer a packet through a specific set of nodes and links in the network without relying on the intermediate nodes in the network to decide the actual path it must take. SPRING engages IGPs, such as IS-IS and OSPF, for advertising network segments. Network segments can represent any instruction, topological or service-based. Within IGP topologies, IGP segments are advertised by the link-state routing protocols. There are two types of IGP segments:

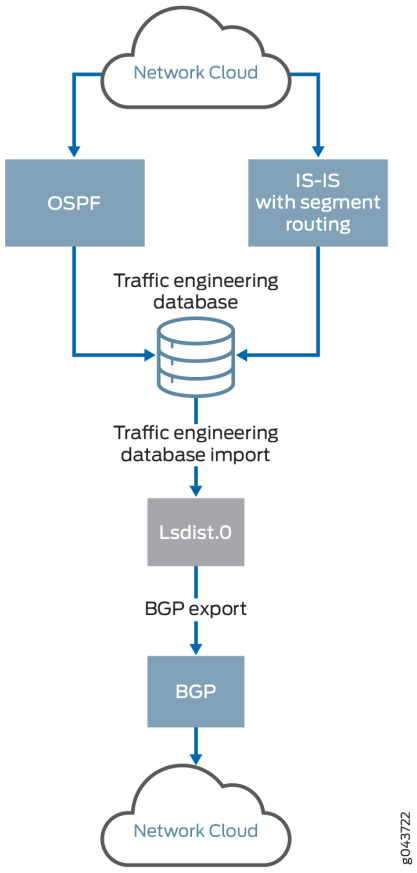
Adjacency segment	A one-hop path over a specific adjacency between two nodes in the IGP
Prefix segment	A multi-hop, equal-cost, multipath-aware shortest-path to a prefix, as per the state of the IGP topology

When SPRING in enabled in a BGP network, BGP link-state address family learns the SPRING information from the IGP link-state routing protocols and advertises segments in the form of segment identifiers (SIDs). BGP link-state address family has been extended to carry SIDs and other SPRING-related information to BGP peers. The route reflector can steer a packet through a desired set of nodes and links by prepending the packet with an appropriate combination of tunnels. This feature allows BGP link-state address family to also advertise the SPRING information to BGP peers.

Flow of BGP Link-State SPRING Data

Figure 111 on page 1727 depicts the data flow of BGP link-state SPRING data that IS-IS pushes to the traffic engineering database.

Figure 111: BGP Link-State Source Packet Routing in Networking (SPRING)



- IGP pushes the SPRING attributes to the traffic engineering database.
- SPRING capabilities and algorithm information are carried forward as node attributes into the traffic engineering database.
- Adjacent SID and LAN adjacent SID information are carried as link attributes.
- Prefix SID or node-SID information is carried as prefix attributes.
- A new set or a change to existing attributes triggers IGP updates to the traffic engineering database with new data.

> ⚠ **CAUTION**: If traffic engineering is disabled at the IGP level, none of the attributes are pushed to the traffic engineering database.

- All parameters in the BGP traffic engineering NLRI, including the link, node, and prefix descriptors are derived from entries in the traffic engineering database.

- The traffic engineering database imports route entries into the `lsdist.0` routing table from IGP subject to policy.
- The default policy of BGP is to export routes, which are known to BGP only. You configure an export policy for non-BGP routes in the `lsdis.0` routing table. This policy advertises an entry learned from the traffic engineering database.

Supported BGP Link-State Attributes and TLVs, and Unsupported Features for BGP Link-State with SPRING

BGP link-state with SPRING supports the following attributes and type, length, and values (TLVs) that are originated, received, and propagated in the network:

Node attributes

- Segment routing Capabilities
- Segment routing Algorithm

Link attributes

- Adjacent-SID
- LAN Adjacent-SID

Prefix descriptors

- IP reachability information

Prefix attributes

- Prefix SID

The following list supports TLVs that are not originated, but only received and propagated in the network:

Prefix descriptors

- Multitopology ID
- OSPF route type

Prefix attributes

- Range
- Binding SID

Junos OS does not support the following features with BGP link-state with SPRING extensions:

- IPv6 prefix origination
- Multitopology identifiers
- Traffic engineering database export for SPRING parameters
- New TLVs with tcpdump (existing TLVs are also not supported).
- SPRING over IPv6

Verifying NLRI Node Learned Through BGP with OSPF as IGP

The following is a sample output to verify the NLRI node learned through BGP with OSPF as the IGP:

Purpose

Verify the lsdist.0 routing table entries.

Action

From operational mode, run the `show route table lsdist.0` command.

```
user@host> show route table lsdist.0 te-node-ip 10.7.7.7 extensive
lsdist.0: 216 destinations, 216 routes (216 active, 0 holddown, 0 hidden)
NODE { AS:65100 Area:0.0.0.1 IPv4:10.7.7.7 OSPF:0 }/1536 (1 entry, 1 announced)
TSI:
LINK-STATE attribute handle 0x61d5da0
        *BGP    Preference: 170/-101
                Next hop type: Indirect, Next hop index: 0
                Address: 0x61b07cc
                Next-hop reference count: 216
                Source: 10.2.2.2
                Protocol next hop: 10.2.2.2
                Indirect next hop: 0x2 no-forward INH Session ID: 0x0
                State:<Active Int Ext>
                Local AS:   65100 Peer AS:   65100
                Age: 30:22      Metric2: 2
                Validation State: unverified
                Task: BGP_65100.10.2.2.2
                Announcement bits (1): 0-TED Export
                AS path: I
                Accepted
                Area border router: No
```

```
                External router: No
                Attached: No
                Overload: No
                SPRING-Capabilities:
                 - SRGB block [Start: 900000, Range: 90000, Flags: 0x00]
                SPRING-Algorithms:
                 - Algo: 0
                Localpref: 100
                Router ID: 10.2.2.2
                Indirect next hops: 1
                        Protocol next hop: 10.2.2.2 Metric: 2
                        Indirect next hop: 0x2 no-forward INH Session ID: 0x0
                        Indirect path forwarding next hops: 1
                                Next hop type: Router
                                Next hop: 10.11.1.2 via et-0/0/0.1 weight 0x1
                                Session Id: 0x143
                                10.2.2.2/32 Originating RIB: inet.0
                                  Metric: 2     Node path count: 1
                                  Forwarding nexthops: 1
                                        Nexthop: 10.11.1.2 via et-0/0/0.1
                                        Session Id: 143
```

Meaning

The routes are appearing in the lsdist.0 routing table.

Verifying the Prefix NLRI Learned Through BGP with OSPF as IGP

The following is a sample output to verify the prefix NLRI learned through BGP with OSPF as the IGP:

Purpose

Verify the lsdist.0 routing table entries.

Action

From operational mode, run the `show route table lsdist.0` command.

```
user@host> show route table lsdist.0 te-ipv4-prefix-node-ip 10.7.7.7 extensive
lsdist.0: 216 destinations, 216 routes (216 active, 0 holddown, 0 hidden)
PREFIX { Node { AS:65100 Area:0.0.0.1 IPv4:10.7.7.7 } { IPv4:10.7.7.7/32 } OSPF:0 }/1536 (1
```

```
entry, 0 announced)
        *BGP    Preference: 170/-101
                Next hop type: Indirect, Next hop index: 0
                Address: 0x61b07cc
                Next-hop reference count: 216
                Source: 10.2.2.2
                Protocol next hop: 10.2.2.2
                Indirect next hop: 0x2 no-forward INH Session ID: 0x0
                State: <Active Int Ext>
                Local AS:   65100 Peer AS:   65100
                Age: 30:51      Metric2: 2
                Validation State: unverified
                Task: BGP_65100.10.2.2.2
                AS path: I
                Accepted
                Prefix Flags: 0x00, Prefix SID: 1007, Flags: 0x50, Algo: 0
                Localpref: 65100
                Router ID: 10.2.2.2
                Indirect next hops: 1
                        Protocol next hop: 10.2.2.2 Metric: 2
                        Indirect next hop: 0x2 no-forward INH Session ID: 0x0
                        Indirect path forwarding next hops: 1
                                Next hop type: Router
                                Next hop: 10.11.1.2 via et-0/0/0.1 weight 0x1
                                Session Id: 0x143
                                10.2.2.2/32 Originating RIB: inet.0
                                  Metric: 2     Node path count: 1
                                  Forwarding nexthops: 1
                                        Nexthop: 10.11.1.2 via et-0/0/0.1
                                        Session Id: 143
```

Meaning

The routes are appearing in the lsdist.0 routing table.

Example: Configuring Link State Distribution Using BGP

IN THIS SECTION

Requirements | 1732

- Overview | **1732**
- Configuration | **1733**
- Verification | **1749**

This example shows how to configure BGP to carry link-state information across multiple domains, which is used for computing paths for MPLS LSPs spanning multiple domains, such as inter-area TE LSP, and providing a scalable and policy-controlled means for external path computing entities, such as ALTO and PCE, to acquire network topology.

Requirements

This example uses the following hardware and software components:

- Four MX Series routers
- Junos OS Release 14.2 or later running on all the routers

Before you begin:

1. Configure the device interfaces.
2. Configure the autonomous system numbers and router IDs for the devices.
3. Configure the following protocols:
   - RSVP
   - MPLS
   - BGP
   - IS-IS
   - OSPF

Overview

IN THIS SECTION

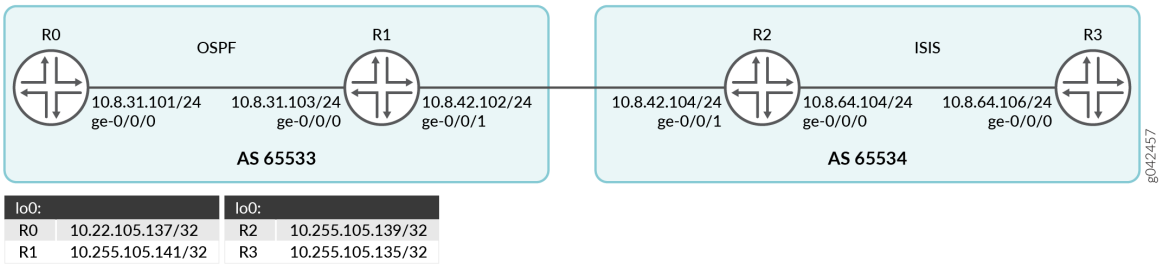
- Topology | **1733**

A new mechanism to distribute topology information across multiple areas and autonomous systems (ASs) is introduced by extending the BGP protocol to carry link -state information, which was initially acquired using IGP. The IGP protocols have scaling limitations when it comes to distributing large databases. BGP is not only a more scalable vehicle for carrying multi-area and multi-AS topology information, but also provides the policy controls that can be useful for multi-AS topology distribution. The BGP link-state topology information is used for computing paths for MPLS label-switched paths (LSPs) spanning multiple domains, such as inter-area TE LSP, and providing a scalable and policy-controlled means for external path computing entities, such as ALTO and PCE, to acquire network topology.

Link state distribution using BGP is supported on QFX10000 switches.

Topology

Figure 112: Link-State Distribution Using BGP



lo0:	
R0	10.22.105.137/32
R1	10.255.105.141/32

lo0:	
R2	10.255.105.139/32
R3	10.255.105.135/32

In Figure 112 on page 1733, Routers R0 and R1 and Routers R2 and R3 belong to different autonomous systems. Routers R0 and R1 run OSPF, and Routers R2 and R3 run IS-IS.

Configuration

IN THIS SECTION

- CLI Quick Configuration | **1734**
- Procedure | **1737**
- Procedure | **1743**

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

R0

```
set interfaces ge-0/0/0 unit 0 family inet address 10.8.31.101/24
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.105.137/32
set routing-options router-id 10.255.105.137
set routing-options autonomous-system 65533
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols mpls traffic-engineering database export policy accept-all
set protocols mpls cross-credibility-cspf
set protocols mpls label-switched-path to-R3-inter-as to 10.255.105.135
set protocols mpls label-switched-path to-R3-inter-as bandwidth 40m
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.255.105.137
set protocols bgp group ibgp family traffic-engineering unicast
set protocols bgp group ibgp neighbor 10.255.105.141
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set policy-options policy-statement accept-all from family traffic-engineering
set policy-options policy-statement accept-all then accept
```

R1

```
set interfaces ge-0/0/0 unit 0 family inet address 10.8.31.103/24
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 10.8.42.102/24
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.105.141/32
set interfaces lo0 unit 0 family iso address 47.0005.0102.5501.8181
```

```
set routing-options router-id 10.255.105.141
set routing-options autonomous-system 65533
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.255.105.141
set protocols bgp group ibgp family traffic-engineering unicast
set protocols bgp group ibgp export nlri2bgp
set protocols bgp group ibgp neighbor 10.255.105.137
set protocols bgp group ebgp type external
set protocols bgp group ebgp family traffic-engineering unicast
set protocols bgp group ebgp neighbor 10.8.42.104 local-address 10.8.42.102
set protocols bgp group ebgp neighbor 10.8.42.104 peer-as 65534
set protocols isis interface ge-0/0/1.0 passive remote-node-iso 0102.5502.4211
set protocols isis interface ge-0/0/1.0 passive remote-node-id 10.8.42.104
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0 passive traffic-engineering remote-node-id
10.8.42.104
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0 passive traffic-engineering remote-node-
router-id 10.255.105.139
set policy-options policy-statement accept-all from family traffic-engineering
set policy-options policy-statement accept-all then accept
set policy-options policy-statement nlri2bgp term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp term 1 then accept
```

R2

```
set interfaces ge-0/0/0 unit 0 family inet address 10.8.64.104/24
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 10.8.42.104/24
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.105.139/32
set interfaces lo0 unit 0 family iso address 47.0005.0102.5502.4211.00
set routing-options router-id 10.255.105.139
set routing-options autonomous-system 65534
set protocols rsvp interface all
```

```
set protocols rsvp interface fxp0.0 disable
set protocols mpls traffic-engineering database import policy ted2nlri
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.255.105.139
set protocols bgp group ibgp family traffic-engineering unicast
set protocols bgp group ibgp export nlri2bgp
set protocols bgp group ibgp neighbor 10.255.105.135
set protocols bgp group ebgp type external
set protocols bgp group ebgp family traffic-engineering unicast
set protocols bgp group ebgp export nlri2bgp
set protocols bgp group ebgp peer-as 65533
set protocols bgp group ebgp neighbor 10.8.42.102
set protocols isis level 1 disable
set protocols isis interface ge-0/0/0.0
set protocols isis interface ge-0/0/1.0 passive remote-node-iso 0102.5501.8181
set protocols isis interface ge-0/0/1.0 passive remote-node-id 10.8.42.102
set protocols isis interface lo0.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0 passive traffic-engineering remote-node-id
10.8.42.102
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0 passive traffic-engineering remote-node-
router-id 10.255.105.141
set policy-options policy-statement accept-all from family traffic-engineering
set policy-options policy-statement accept-all then accept
set policy-options policy-statement nlri2bgp term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp term 1 then accept
set policy-options policy-statement ted2nlri term 1 from protocol isis
set policy-options policy-statement ted2nlri term 1 from protocol ospf
set policy-options policy-statement ted2nlri term 1 then accept
set policy-options policy-statement ted2nlri term 2 then reject
```

R3

```
set interfaces ge-0/0/0 unit 0 family inet address 10.8.64.106/24
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.105.135/32
set interfaces lo0 unit 0 family iso address 47.0005.0102.5502.4250
set routing-options router-id 10.255.105.135
set routing-options autonomous-system 65534
```

```
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols mpls traffic-engineering database export policy accept-all
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.255.105.135
set protocols bgp group ibgp family traffic-engineering unicast
set protocols bgp group ibgp neighbor 10.255.105.139
set protocols isis interface ge-0/0/0.0 level 1 disable
set protocols isis interface lo0.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set policy-options policy-statement accept-all from family traffic-engineering
set policy-options policy-statement accept-all then accept
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode*.

To configure Router R1:

1. Configure the Router R1 interfaces.

```
[edit interfaces]
user@R1# set ge-0/0/0 unit 0 family inet address 10.8.31.103/24
user@R1# set ge-0/0/0 unit 0 family iso
user@R1# set ge-0/0/0 unit 0 family mpls
user@R1# set ge-0/0/1 unit 0 family inet address 10.8.42.102/24
user@R1# set ge-0/0/1 unit 0 family iso
user@R1# set ge-0/0/1 unit 0 family mpls
user@R1# set lo0 unit 0 family inet address 10.255.105.141/32
user@R1# set lo0 unit 0 family iso address 47.0005.0102.5501.8181
```

2. Configure the router ID and autonomous system of Router R1.

```
[edit routing-options]
user@R1# set router-id 10.255.105.141
user@R1# set autonomous-system 65533
```

3. Enable RSVP on all the interfaces of Router R1 (excluding the management interface).

```
[edit protocols]
user@R1# set rsvp interface all
user@R1# set rsvp interface fxp0.0 disable
```

4. Enable MPLS on all the interfaces of Router R1 (excluding the management interface).

```
[edit protocols]
user@R1# set mpls interface all
user@R1# set mpls interface fxp0.0 disable
```

5. Configure the BGP group for Router R1 to peer with Router R0, and assign the local address and neighbor address.

```
[edit protocols]
user@R1# set bgp group ibgp type internal
user@R1# set bgp group ibgp local-address 10.255.105.141
user@R1# set bgp group ibgp neighbor 10.255.105.137
```

6. Include the BGP-TE signaling network layer reachability information (NLRI) to the ibgp BGP group.

```
[edit protocols]
user@R1# set bgp group ibgp family traffic-engineering unicast
```

7. Enable export of policy nlri2bgp on Router R1.

```
[edit protocols]
user@R1# set bgp group ibgp export nlri2bgp
```

8. Configure the BGP group for Router R1 to peer with Router R2, and assign the local address and neighbor autonomous system to the ebgp BGP group.

```
[edit protocols]
user@R1# set bgp group ebgp type external
user@R1# set bgp group ebgp neighbor 10.8.42.104 local-address 10.8.42.102
user@R1# set bgp group ebgp neighbor 10.8.42.104 peer-as 65534
```

9. Include the BGP-TE signaling NLRI to the ebgp BGP group.

```
[edit protocols]
user@R1# set bgp group ebgp family traffic-engineering unicast
```

10. Enable passive traffic-engineering on the inter-AS link.

```
[edit protocols]
user@R1# set isis interface ge-0/0/1.0 passive remote-node-iso 0102.5502.4211
user@R1# set isis interface ge-0/0/1.0 passive remote-node-id 10.8.42.104
```

11. Enable OSPF on the interface connecting Router R1 to Router R0 and on the loopback interface of Router R1, and enable traffic engineering capabilities.

```
[edit protocols]
user@R1# set ospf traffic-engineering
user@R1# set ospf area 0.0.0.0 interface lo0.0
user@R1# set ospf area 0.0.0.0 interface ge-0/0/0.0
```

12. Enable passive traffic-engineering on the inter-AS link.

```
[edit protocols]
user@R1# set ospf area 0.0.0.0 interface ge-0/0/1.0 passive traffic-engineering remote-node-
id 10.8.42.104
user@R1# set ospf area 0.0.0.0 interface ge-0/0/1.0 passive traffic-engineering remote-node-
router-id 10.255.105.139
```

13. Configure policies to accept traffic from BGP-TE NLRI.

```
[edit policy-options]
user@R1# set policy-statement accept-all from family traffic-engineering
user@R1# set policy-statement accept-all then accept
user@R1# set policy-statement nlri2bgp term 1 from family traffic-engineering
user@R1# set policy-statement nlri2bgp term 1 then accept
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, `show protocols`, and `show policy-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R1# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 10.8.31.103/24;
        }
        family iso;
        family mpls;
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.8.42.102/24;
        }
        family iso;
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.255.105.141/32;
        family iso {
            address 47.0005.0102.5501.8181:00;
        }
```

```
    }
}
```

```
user@R1# show routing-options
router-id 10.255.105.141;
autonomous-system 65533;
```

```
user@R1# show protocols
rsvp {
    interface all;
    interface fxp0.0 {
        disable;
    }
}
mpls {
    interface all;
    interface fxp0.0 {
        disable;
    }
}
bgp {
    group ibgp {
        type internal;
        local-address 10.255.105.141;
        family traffic-engineering {
            unicast;
        }
        export nlri2bgp;
        neighbor 10.255.105.137;
    }
    group ebgp {
        type external;
        family traffic-engineering {
            unicast;
        }
        neighbor 10.8.42.104 {
            local-address 10.8.42.102;
            peer-as 65534;
        }
    }
```

```
}
isis {
    interface ge-0/0/1.0 {
        passive {
            remote-node-iso 0102.5502.4211;
            remote-node-id 10.8.42.104;
        }
    }
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface lo0.0;
        interface ge-0/0/0.0;
        interface ge-0/0/1.0 {
            passive {
                traffic-engineering {
                    remote-node-id 10.8.42.104;
                    remote-node-router-id 10.255.105.139;
                }
            }
        }
    }
}
```

```
user@R1# show policy-options
policy-statement accept-all {
    from family traffic-engineering;
    then accept;
}
policy-statement nlri2bgp {
    term 1 {
        from family traffic-engineering;
        then {
            accept;
        }
    }
}
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode*.

To configure Router R2:

1. Configure the Router R2 interfaces.

```
[edit interfaces]
user@R2# set ge-0/0/0 unit 0 family inet address 10.8.64.104/24
user@R2# set ge-0/0/0 unit 0 family iso
user@R2# set ge-0/0/0 unit 0 family mpls
user@R2# set ge-0/0/1 unit 0 family inet address 10.8.42.104/24
user@R2# set ge-0/0/1 unit 0 family iso
user@R2# set ge-0/0/1 unit 0 family mpls
user@R2# set lo0 unit 0 family inet address 10.255.105.139/32
user@R2# set lo0 unit 0 family iso address 47.0005.0102.5502.4211.00
```

2. Configure the router ID and autonomous system of Router R2.

```
[edit routing-options]
user@R2# set router-id 10.255.105.139
user@R2# set autonomous-system 65534
```

3. Enable RSVP on all the interfaces of Router R2 (excluding the management interface).

```
[edit routing-options]
user@R2# set rsvp interface all
user@R2# set rsvp interface fxp0.0 disable
```

4. Enable MPLS on all the interfaces of Router R2 (excluding the management interface).

```
[edit routing-options]
user@R2# set mpls interface all
user@R2# set mpls interface fxp0.0 disable
```

5. Enable import of traffic engineering database parameters using the ted2nlri policy.

```
[edit protocols]
user@R2# set mpls traffic-engineering database import policy ted2nlri
```

6. Configure the BGP group for Router R2 to peer with Router R3, and assign the local address and neighbor address.

```
[edit protocols]
user@R2# set bgp group ibgp type internal
user@R2# set bgp group ibgp local-address 10.255.105.139
user@R2# set bgp group ibgp neighbor 10.255.105.135
```

7. Include the BGP-TE signaling network layer reachability information (NLRI) to the ibgp BGP group.

```
[edit protocols]
user@R2# set bgp group ibgp family traffic-engineering unicast
```

8. Enable export of policy nlri2bgp on Router R2.

```
[edit protocols]
user@R2# set bgp group ibgp export nlri2bgp
```

9. Configure the BGP group for Router R2 to peer with Router R1.

```
[edit protocols]
user@R2# set bgp group ebgp type external
```

10. Include the BGP-TE signaling NLRI to the ebgp BGP group.

```
[edit protocols]
user@R2# set bgp group ebgp family traffic-engineering unicast
```

11. Assign the local address and neighbor autonomous system to the ebgp BGP group.

```
[edit protocols]
user@R2# set bgp group ebgp peer-as 65533
user@R2# set bgp group ebgp neighbor 10.8.42.102
```

12. Enable export of policy nlri2bgp on Router R2.

```
[edit protocols]
user@R2# set bgp group ebgp export nlri2bgp
```

13. Enable IS-IS on the interface connecting Router R2 with Router R3 and the loopback interface of Router R2.

```
[edit protocols]
user@R2# set isis level 1 disable
user@R2# set isis interface ge-0/0/0.0
user@R2# set isis interface lo0.0
```

14. Enable only IS-IS advertising on the interface connecting Router R2 with Router R1.

```
[edit protocols]
user@R2# set isis interface ge-0/0/1.0 passive remote-node-iso 0102.5501.8181
user@R2# set isis interface ge-0/0/1.0 passive remote-node-id 10.8.42.102
```

15. Configure traffic engineering capability on Router R2.

```
[edit protocols]
user@R2# set ospf traffic-engineering
```

16. Enable only OSPF advertisements on the interface connecting Router R2 with Router R1.

```
[edit protocols]
user@R2# set ospf area 0.0.0.0 interface ge-0/0/1.0 passive traffic-engineering remote-node-
id 10.8.42.102
```

```
user@R2# set ospf area 0.0.0.0 interface ge-0/0/1.0 passive traffic-engineering remote-node-
router-id 10.255.105.141
```

17. Configure policies to accept traffic from the BGP-TE NLRI.

```
[edit policy-options]
user@R2# set policy-statement accept-all from family traffic-engineering
user@R2# set policy-statement accept-all then accept
user@R2# set policy-statement nlri2bgp term 1 from family traffic-engineering
user@R2# set policy-statement nlri2bgp term 1 then accept
user@R2# set policy-statement ted2nlri term 1 from protocol isis
user@R2# set policy-statement ted2nlri term 1 from protocol ospf
user@R2# set policy-statement ted2nlri term 1 then accept
user@R2# set policy-statement ted2nlri term 2 then reject
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, `show protocols`, and `show policy-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R2# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 10.8.64.104/24;
        }
        family iso;
        family mpls;
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.8.42.104/24;
        }
        family iso;
        family mpls;
    }
}
```

```
lo0 {
    unit 0 {
        family inet {
            address 10.255.105.139/32;
        family iso {
            address 47.0005.0102.5502.4211.00;
        }
        family iso;
    }
}
```

```
user@R2# show routing-options
router-id 10.255.105.139;
 autonomous-system 65534;
```

```
user@R2# show protocols
rsvp {
    interface all;
    interface fxp0.0 {
        disable;
    }
}
mpls {
    traffic-engineering {
        database {
            import {
                policy ted2nlri;
            }
        }
    }
    interface all;
    interface fxp0.0 {
        disable;
    }
}
bgp {

    group ibgp {
        type internal;
        local-address 10.255.105.139;
```

```
        family traffic-engineering {
            unicast;
        }
        export nlri2bgp;
        neighbor 10.255.105.135;
    }
    group ebgp {
        type external;
        family traffic-engineering {
            unicast;
        }
        export nlri2bgp;
        peer-as 65533;
        neighbor 10.8.42.102;
    }
}
isis {
    level 1 disable;
    interface ge-0/0/0.0;
    interface ge-0/0/1.0 {
        passive {
            remote-node-iso 0102.5501.8181;
            remote-node-id 10.8.42.102;
        }
    }
    interface lo0.0;
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface ge-0/0/1.0 {
            passive {
                traffic-engineering {
                    remote-node-id 10.8.42.102;
                    remote-node-router-id 10.255.105.141;
                }
            }
        }
```

```
    }
}
```

```
user@R2# show policy-options
policy-statement accept-all {
    from family traffic-engineering;
    then accept;
}
policy-statement nlri2bgp {
    term 1 {
        from family traffic-engineering;
        then {
            accept;
        }
    }
}
policy-statement ted2nlri {
    term 1 {
        from protocol [ isis ospf ];
        then accept;
    }
    term 2 {
        then reject;
    }
}
```

Verification

IN THIS SECTION

- Verifying the BGP Summary Status | **1750**
- Verifying the MPLS LSP Status | **1751**
- Verifying the Isdist.0 Routing Table Entries | **1751**
- Verifying the Traffic Engineering Database Entries | **1755**

Verify that the configuration is working properly.

Verifying the BGP Summary Status

Purpose

Verify that BGP is up and running on Routers R0 and R1.

Action

From operational mode, run the `show bgp summary` command.

```
user@R0> show bgp summary
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
lsdist.0
                      10         10          0          0          0          0
Peer                    AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
10.255.105.141        65533         20         14       0      79        5:18 Establ
  lsdist.0: 10/10/10/0
```

From operational mode, run the `show bgp summary` command.

```
user@R1> show bgp summary
Groups: 2 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
lsdist.0
                      10         10          0          0          0          0
Peer                    AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
10.8.42.104           65534         24         17       0      70        6:43 Establ
  lsdist.0: 10/10/10/0
10.255.105.137        65533         15         23       0      79        6:19 Establ
  lsdist.0: 0/0/0/0
```

Meaning

Router R0 is peered with Router R1.

Verifying the MPLS LSP Status

Purpose

Verify the status of the MPLS LSP on Router R0.

Action

From operational mode, run the `show mpls lsp` command.

```
user@R0> show mpls lsp
Ingress LSP: 1 sessions
To              From            State Rt P     ActivePath       LSPname
10.255.105.135  10.255.105.137  Up     0 *                      to-R3-inter-as
Total 1 displayed, Up 1, Down 0

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Meaning

The MPLS LSP from Router R0 to Router R3 is established.

Verifying the lsdist.0 Routing Table Entries

Purpose

Verify the lsdist.0 routing table entries on Routers R0, R1, and R2.

Action

From operational mode, run the `show route table lsdist.0` command.

```
user@R0> show route table lsdist.0
lsdist.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
NODE { AS:65534 ISO:0102.5502.4211.00 ISIS-L2:0 }/1152
                    *[BGP/170] 00:17:32, localpref 100, from 10.255.105.141
                       AS path: 65534 I, validation-state: unverified
                     > to 10.8.31.103 via ge-0/0/0.0
NODE { AS:65534 ISO:0102.5502.4250.00 ISIS-L2:0 }/1152
                    *[BGP/170] 00:17:32, localpref 100, from 10.255.105.141
                       AS path: 65534 I, validation-state: unverified
                     > to 10.8.31.103 via ge-0/0/0.0
NODE { AS:65534 ISO:0102.5502.4250.02 ISIS-L2:0 }/1152
                    *[BGP/170] 00:17:32, localpref 100, from 10.255.105.141
                       AS path: 65534 I, validation-state: unverified
                     > to 10.8.31.103 via ge-0/0/0.0
NODE { AS:65534 Area:0.0.0.0 IPv4:10.255.105.139 OSPF:0 }/1152
                    *[BGP/170] 00:17:32, localpref 100, from 10.255.105.141
                       AS path: 65534 I, validation-state: unverified
                     > to 10.8.31.103 via ge-0/0/0.0
LINK { Local { AS:65534 ISO:0102.5502.4211.00 }.{ IPv4:8.42.1.104 } Remote { AS:65534
ISO:0102.5501.8181.00 }.{ IPv4:10.8.42.102 } ISIS-L2:0 }/1152
                    *[BGP/170] 00:17:32, localpref 100, from 10.255.105.141
                       AS path: 65534 I, validation-state: unverified
                     > to 10.8.31.103 via ge-0/0/0.0
LINK { Local { AS:65534 ISO:0102.5502.4211.00 }.{ IPv4:10.8.64.104 } Remote { AS:65534
ISO:0102.5502.4250.02 }.{ } ISIS-L2:0 }/1152
                    *[BGP/170] 00:02:03, localpref 100, from 10.255.105.141
                       AS path: 65534 I, validation-state: unverified
                     > to 10.8.31.103 via ge-0/0/0.0
LINK { Local { AS:65534 ISO:0102.5502.4250.00 }.{ IPv4:10.8.64.106 } Remote { AS:65534
ISO:0102.5502.4250.02 }.{ } ISIS-L2:0 }/1152
                    *[BGP/170] 00:17:32, localpref 100, from 10.255.105.141
                       AS path: 65534 I, validation-state: unverified
                     > to 10.8.31.103 via ge-0/0/0.0
LINK { Local { AS:65534 ISO:0102.5502.4250.02 }.{ } Remote { AS:65534 ISO:0102.5502.4211.00 }.
{ } ISIS-L2:0 }/1152
                    *[BGP/170] 00:17:32, localpref 100, from 10.255.105.141
                       AS path: 65534 I, validation-state: unverified
                     > to 10.8.31.103 via ge-0/0/0.0
LINK { Local { AS:65534 ISO:0102.5502.4250.02 }.{ } Remote { AS:65534 ISO:0102.5502.4250.00 }.
{ } ISIS-L2:0 }/1152
                    *[BGP/170] 00:17:32, localpref 100, from 10.255.105.141
                       AS path: 65534 I, validation-state: unverified
                     > to 10.8.31.103 via ge-0/0/0.0
LINK { Local { AS:65534 Area:0.0.0.0 IPv4:10.255.105.139 }.{ IPv4:10. 8.42.104 } Remote
{ AS:65534 Area:0.0.0.0 IPv4:10.255.105.141 }.{ IPv4:10.8.42.102 } OSPF:0 }/1152
```

```
                    *[BGP/170] 00:17:32, localpref 100, from 10.255.105.141
                       AS path: 65534 I, validation-state: unverified
                     > to 10.8.31.103 via ge-0/0/0.0
```

From operational mode, run the `show route table lsdist.0` command.

```
user@R1> show route table lsdist.0
lsdist.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

NODE { AS:65534 ISO:0102.5502.4211.00 ISIS-L2:0 }/1152
                    *[BGP/170] 00:18:00, localpref 100
                       AS path: 65534 I, validation-state: unverified
                     > to 10.8.42.104 via ge-0/0/1.0
NODE { AS:65534 ISO:0102.5502.4250.00 ISIS-L2:0 }/1152
                    *[BGP/170] 00:18:00, localpref 100
                       AS path: 65534 I, validation-state: unverified
                     > to 10.8.42.104 via ge-0/0/1.0
NODE { AS:65534 ISO:0102.5502.4250.02 ISIS-L2:0 }/1152
                    *[BGP/170] 00:18:00, localpref 100
                       AS path: 65534 I, validation-state: unverified
                     > to 10.8.42.104 via ge-0/0/1.0
NODE { AS:65534 Area:0.0.0.0 IPv4:10.255.105.139 OSPF:0 }/1152
                    *[BGP/170] 00:18:00, localpref 100
                       AS path: 65534 I, validation-state: unverified
                     > to 10.8.42.104 via ge-0/0/1.0
LINK { Local { AS:65534 ISO:0102.5502.4211.00 }.{ IPv4:10.8.42.104 } Remote { AS:65534
ISO:0102.5501.8181.00 }.{ IPv4:10.8.42.102 } ISIS-L2:0 }/1152
                    *[BGP/170] 00:18:00, localpref 100
                       AS path: 65534 I, validation-state: unverified
                     > to 10.8.42.104 via ge-0/0/1.0
LINK { Local { AS:65534 ISO:0102.5502.4211.00 }.{ IPv4:10.8.64.104 } Remote { AS:65534
ISO:0102.5502.4250.02 }.{ } ISIS-L2:0 }/1152
                    *[BGP/170] 00:02:19, localpref 100
                       AS path: 65534 I, validation-state: unverified
                     > to 10.8.42.104 via ge-0/0/1.0
LINK { Local { AS:65534 ISO:0102.5502.4250.00 }.{ IPv4:10.8.64.106 } Remote { AS:65534
ISO:0102.5502.4250.02 }.{ } ISIS-L2:0 }/1152
                    *[BGP/170] 00:18:00, localpref 100
                       AS path: 65534 I, validation-state: unverified
                     > to 10.8.42.104 via ge-0/0/1.0
LINK { Local { AS:65534 ISO:0102.5502.4250.02 }.{ } Remote { AS:65534 ISO:0102.5502.4211.00 }.
```

```
{ } ISIS-L2:0 }/1152
                    *[BGP/170] 00:18:00, localpref 100
                       AS path: 65534 I, validation-state: unverified
                     > to 10.8.42.104 via ge-0/0/1.0
LINK { Local { AS:65534 ISO:0102.5502.4250.02 }.{ } Remote { AS:65534 ISO:0102.5502.4250.00 }.
{ } ISIS-L2:0 }/1152
                    *[BGP/170] 00:18:00, localpref 100
                       AS path: 65534 I, validation-state: unverified
                     > to 10.8.42.104 via ge-0/0/1.0
LINK { Local { AS:65534 Area:0.0.0.0 IPv4:10.255.105.139 }.{ IPv4:10.8.42.104 } Remote
{ AS:65534 Area:0.0.0.0 IPv4:10.255.105.141 }.{ IPv4:10.8.42.102 } OSPF:0 }/1152
                    *[BGP/170] 00:18:00, localpref 100
                       AS path: 65534 I, validation-state: unverified
                     > to 10.8.42.104 via ge-0/0/1.0
```

From operational mode, run the `show route table lsdist.0` command.

```
user@R2> show route table lsdist.0
lsdist.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

NODE { AS:65534 ISO:0102.5502.4211.00 ISIS-L2:0 }/1152
                    *[IS-IS/18] 1d 00:24:39
                       Fictitious
NODE { AS:65534 ISO:0102.5502.4250.00 ISIS-L2:0 }/1152
                    *[IS-IS/18] 00:20:45
                       Fictitious
NODE { AS:65534 ISO:0102.5502.4250.02 ISIS-L2:0 }/1152
                    *[IS-IS/18] 00:20:45
                       Fictitious
NODE { AS:65534 Area:0.0.0.0 IPv4:10.255.105.139 OSPF:0 }/1152
                    *[OSPF/10] 1d 00:24:39
                       Fictitious
LINK { Local { AS:65534 ISO:0102.5502.4211.00 }.{ IPv4:10.8.42.104 } Remote { AS:65534
ISO:0102.5501.8181.00 }.{ IPv4:10.8.42.102 } ISIS-L2:0 }/1152
                    *[IS-IS/18] 00:20:58
                       Fictitious
LINK { Local { AS:65534 ISO:0102.5502.4211.00 }.{ IPv4:10.8.64.104 } Remote { AS:65534
ISO:0102.5502.4250.02 }.{ } ISIS-L2:0 }/1152
                    *[IS-IS/18] 00:02:34
                       Fictitious
LINK { Local { AS:65534 ISO:0102.5502.4250.00 }.{ IPv4:10.8.64.106 } Remote { AS:65534
```

```
ISO:0102.5502.4250.02 }.{ } ISIS-L2:0 }/1152
                  *[IS-IS/18] 00:20:45
                     Fictitious
LINK { Local { AS:65534 ISO:0102.5502.4250.02 }.{ } Remote { AS:65534 ISO:0102.5502.4211.00 }.
{ } ISIS-L2:0 }/1152
                  *[IS-IS/18] 00:20:45
                     Fictitious
LINK { Local { AS:65534 ISO:0102.5502.4250.02 }.{ } Remote { AS:65534 ISO:0102.5502.4250.00 }.
{ } ISIS-L2:0 }/1152
                  *[IS-IS/18] 00:20:45
                     Fictitious
LINK { Local { AS:65534 Area:0.0.0.0 IPv4:10.255.105.139 }.{ IPv4:10.8.42.104 } Remote
{ AS:65534 Area:0.0.0.0 IPv4:10.255.105.141 }.{ IPv4:10.8.42.102 } OSPF:0 }/1152
                  *[OSPF/10] 00:20:57
                     Fictitious
```

Meaning

The routes are appearing in the lsdist.0 routing table.

Verifying the Traffic Engineering Database Entries

Purpose

Verify the traffic engineering database entries on Router R0.

Action

From operational mode, run the `show ted database` command.

```
user@R0> show ted database
TED database: 5 ISIS nodes 5 INET nodes
ID                           Type Age(s) LnkIn LnkOut Protocol
0102.5501.8168.00(10.255.105.137) Rtr   1046     1      1 OSPF(0.0.0.0)
    To: 10.8.31.101-1, Local: 10.8.31.101, Remote: 0.0.0.0
      Local interface index: 0, Remote interface index: 0
ID                           Type Age(s) LnkIn LnkOut Protocol
0102.5501.8181.00            ---    1033     1      0
0102.5502.4211.00(10.255.105.139) Rtr   3519     2      3 Exported ISIS-L2(1)
    To: 0102.5502.4250.02, Local: 10.8.64.104, Remote: 0.0.0.0
      Local interface index: 0, Remote interface index: 0
```

```
    To: 0102.5501.8181.00, Local: 10.8.42.104, Remote: 10.8.42.102
      Local interface index: 0, Remote interface index: 0
ID                           Type Age(s) LnkIn LnkOut Protocol
                                                      Exported OSPF(2)
    To: 10.255.105.141, Local: 10.8.42.104, Remote: 10.8.42.102
      Local interface index: 0, Remote interface index: 0
ID                           Type Age(s) LnkIn LnkOut Protocol
0102.5502.4250.00(10.255.105.135) Rtr   1033     1      1 Exported ISIS-L2(1)
    To: 0102.5502.4250.02, Local: 10.8.64.106, Remote: 0.0.0.0
      Local interface index: 0, Remote interface index: 0
ID                           Type Age(s) LnkIn LnkOut Protocol
0102.5502.4250.02            Net    1033     2      2 Exported ISIS-L2(1)
    To: 0102.5502.4211.00(10.255.105.139), Local: 0.0.0.0, Remote: 0.0.0.0
      Local interface index: 0, Remote interface index: 0
    To: 0102.5502.4250.00(10.255.105.135), Local: 0.0.0.0, Remote: 0.0.0.0
      Local interface index: 0, Remote interface index: 0
ID                           Type Age(s) LnkIn LnkOut Protocol
10.8.31.101-1                 Net    1046     2      2 OSPF(0.0.0.0)
    To: 0102.5501.8168.00(10.255.105.137), Local: 0.0.0.0, Remote: 0.0.0.0
      Local interface index: 0, Remote interface index: 0
    To: 10.255.105.141, Local: 0.0.0.0, Remote: 0.0.0.0
      Local interface index: 0, Remote interface index: 0
ID                           Type Age(s) LnkIn LnkOut Protocol
10.255.105.141               Rtr    1045     2      2 OSPF(0.0.0.0)
    To: 0102.5502.4211.00(10.255.105.139), Local: 10.8.42.102, Remote: 10.8.42.104
      Local interface index: 0, Remote interface index: 0
    To: 10.8.31.101-1, Local: 10.8.31.103, Remote: 0.0.0.0
      Local interface index: 0, Remote interface index: 0
```

Meaning

The routes are appearing in the traffic engineering database.

Configuring Link State Distribution Using BGP

You can enable distribution of topology information across multiple areas and autonomous systems (ASs) by extending the BGP protocol to carry link-state information, which was initially acquired using IGP. The IGP protocols have scaling limitations when it comes to distributing large databases. BGP is not only a more scalable vehicle for carrying multi-area and multi-AS topology information, but also provides the policy controls that can be useful for multi-AS topology distribution. The BGP link-state topology information is used for computing paths for MPLS LSPs spanning multiple domains, such as inter-area

TE LSP, and providing a scalable and policy-controlled means for external path computing entities, such as ALTO and PCE, to acquire network topology.

Before you begin:

1. Configure the device interfaces.
2. Configure the router ID and autonomous system number for the device.
3. Configure the following protocols:
   - RSVP
   - MPLS
   - IS-IS
   - OSPF

To enable link-state distribution using BGP:

1. Configure an internal BGP group, and assign the local address and neighbor address for the group.

```
[edit protocols]
user@R1# set bgp group internal-group-name type internal
user@R1# set bgp group internal-group-name local-address ip-address
user@R1# set bgp group internal-group-name neighbor ip-address
```

2. Include the BGP-TE signaling network layer reachability information (NLRI) to the internal BGP group.

```
[edit protocols]
user@R1# set bgp group internal-group-name family traffic-engineering unicast
```

3. Enable export of policy on the device.

```
[edit protocols]
user@R1# set bgp group internal-group-name export second-policy-name
```

4. Configure an external BGP group, and assign the local address and neighbor autonomous system to the group.

```
[edit protocols]
user@R1# set bgp group external-group-name type external
```

```
user@R1# set bgp group external-group-name neighbor ip-address local-address ip-address
user@R1# set bgp group external-group-name neighbor ip-address peer-as as-number
```

5. Include the BGP-TE signaling NLRI to the external BGP group.

```
[edit protocols]
user@R1# set bgp group external-group-name family traffic-engineering unicast
```

6. In configuration mode, go to the following hierarchy level:

```
[edit]
user@R1# edit policy-options
```

7. Configure policies to accept traffic from the BGP-TE NLRI.

```
[edit policy-options]
user@R1# set policy-statement policy-name from family traffic-engineering
user@R1# set policy-statement policy-name then accept
user@R1# set policy-statement bgp-import-policy term 1 from family traffic-engineering
user@R1# set policy-statement bgp-import-policy term 1 then next-hop self
user@R1# set policy-statement bgp-import-policy term 1 then accept
```

8. On the remote connecting device, configure policy to accept the OSPF and IS-IS traffic.

```
[edit policy-options]
user@R2# set policy-statement bgp-export-policy term 1 from protocol isis
user@R2# set policy-statement bgp-export-policy term 1 from protocol ospf
user@R2# set policy-statement bgp-export-policy term 1 then accept
user@R2# set policy-statement bgp-export-policy term 2 then reject
```

9. Verify and commit the configuration.

   For example:

```
R1
[edit protocols]
user@R1# set rsvp interface all
user@R1# set rsvp interface fxp0.0 disable
user@R1# set mpls interface all
user@R1# set mpls interface fxp0.0 disable
user@R1# set bgp group ibgp type internal
```

```
user@R1# set bgp group ibgp local-address 10.255.105.141
user@R1# set bgp group ibgp family traffic-engineering unicast
user@R1# set bgp group ibgp export nlri2bgp
user@R1# set bgp group ibgp neighbor 10.255.105.137
user@R1# set bgp group ebgp type external
user@R1# set bgp group ebgp family traffic-engineering unicast
user@R1# set bgp group ebgp neighbor 8.42.1.104 local-address 8.42.1.102
user@R1# set bgp group ebgp neighbor 8.42.1.104 peer-as 65534
user@R1# set isis interface ge-0/0/1.0 passive remote-node-iso 0102.5502.4211
user@R1# set isis interface ge-0/0/1.0 passive remote-node-id 8.42.1.104
user@R1# set ospf traffic-engineering
user@R1# set ospf area 0.0.0.0 interface lo0.0
user@R1# set ospf area 0.0.0.0 interface ge-0/0/0.0
user@R1# set ospf area 0.0.0.0 interface ge-0/0/1.0 passive traffic-engineering remote-node-
id 8.42.1.104
user@R1# set ospf area 0.0.0.0 interface ge-0/0/1.0 passive traffic-engineering remote-node-
router-id 10.255.105.139
```

```
[edit policy-options]
user@R1# set policy-statement accept-all from family traffic-engineering
user@R1# set policy-statement accept-all then accept
user@R1# set policy-statement nlri2bgp term 1 from family traffic-engineering
user@R1# set policy-statement nlri2bgp term 1 then next-hop self
user@R1# set policy-statement nlri2bgp term 1 then accept
```

```
[edit]
user@R1# commit
commit complete
```

R2

```
[edit policy-options]
user@R2# set policy-statement accept-all from family traffic-engineering
user@R2# set policy-statement accept-all then accept
user@R2# set policy-statement nlri2bgp term 1 from family traffic-engineering
user@R2# set policy-statement nlri2bgp term 1 then next-hop self
user@R2# set policy-statement nlri2bgp term 1 then accept
user@R2# set policy-statement ted2nlri term 1 from protocol isis
user@R2# set policy-statement ted2nlri term 1 from protocol ospf
```

```
user@R2# set policy-statement ted2nlri term 1 then accept
user@R2# set policy-statement ted2nlri term 2 then reject
```

```
[edit]
user@R2# commit
commit complete
```

BGP Classful Transport Planes Overview

IN THIS SECTION

- Benefits of BGP Classful Transport Planes | **1760**
- Terminology of BGP Classful Transport Planes | **1761**
- Understanding BGP Classful Transport Planes | **1762**
- Intra-AS Implementation of BGP Classful Transport Planes | **1763**
- Inter-AS Implementation of BGP Classful Transport Planes | **1766**

Benefits of BGP Classful Transport Planes

- **Network-slicing**–Service and transport layers are decoupled from each other, laying the foundation for network-slicing and virtualization with the end-to-end slicing across multiple domains, thereby significantly reducing the CAPEX.
- **Inter-domain interoperability**–Extends transport class deployment across co-operating domains so the different transport signaling protocols in each domain interoperate. Reconciles any differences between extended community namespaces that may be in use in each domain.
- **Colored resolution with fallback**–Enables resolution over colored tunnels (RSVP, IS-IS flexible algorithm) with flexible fallback options over best-effort tunnels or any other color tunnel.
- **Quality-of-service**–Customizes and optimizes the network to achieve the end-to-end SLA requirements.
- **Leveraging existing deployments**–Supports well deployed tunneling protocols like RSVP along with new protocols, such as IS-IS flexible algorithm, preserving ROI and reducing OPEX.

Terminology of BGP Classful Transport Planes

This section provides a summary of commonly used terms for understanding BGP classful transport plane.

- **Service node**–Ingress Provider Edge (PE) devices that send and receive service routes (Internet and Layer 3 VPN).
- **Border node**–Device at the connection point of different domains (IGP areas or ASs).
- **Transport node**–Device that sends and receives BGP-Labeled Unicast (LU) routes.
- **BGP-VPN**–VPNs built using RFC4364 mechanisms.
- **Route Target (RT)**–Type of extended community used to define VPN membership.
- **Route Distinguisher (RD)**–Identifier used to distinguish to which VPN or virtual private LAN service (VPLS) a route belongs. Each routing instance must have a unique route distinguisher associated with it.
- **Resolution scheme**–Used to resolve protocol next-hop address (PNH) in resolution RIBs providing fallback.

  They map the routes to the different transport RIBs in the system based on mapping community.
- **Service family**–BGP address family used for advertising routes for data traffic, as opposed to tunnels.
- **Transport family** –BGP address family used for advertising tunnels, which are in turn used by service routes for resolution.
- **Transport tunnel**–A tunnel over which a service may place traffic, for example, GRE, UDP, LDP, RSVP, SR-TE, BGP-LU.
- **Tunnel domain**–A domain of the network containing service nodes and border nodes under a single administrative control that has a tunnel between them. An end-to-end tunnel spanning several adjacent tunnel domains can be created by stitching the nodes together using labels.
- **Transport class**–A group of transport tunnels offering the same type of service.
- **Transport class RT**–A new format of route target used to identify a specific transport class.

  A new format of Route Target used to identify a specific transport class.
- **Transport RIB**–At the service node and border node, a transport class has an associated transport RIB that holds its tunnel routes.
- **Transport RTI**–A routing instance; container of transport RIB, and associated transport class Route Target and Route Distinguisher.

- **Transport plane**–Set of transport RTIs importing same transport class RT. These are in turn stitched together to span across tunnel domain boundaries using a mechanism similar to Inter-AS option-b to swap labels at border nodes (nexthop-self), forming an end-to-end transport plane.
- **Mapping community**–Community on a service route that maps to resolve over a transport class.

Understanding BGP Classful Transport Planes

You can use BGP classful transport planes to configure transport classes for classifying a set of transport tunnels in an intra-AS network based on the traffic engineering characteristics and use these transport tunnels to map service routes with the desired SLA and intended fallback.

BGP classful transport planes can extend these tunnels to inter-domain networks that span across multiple domains (ASs or IGP areas) while preserving the transport class. To do this, you must configure the BGP classful transport transport layer BGP family between the border and service nodes.

In both inter-AS and intra-AS implementations, there can be many transport tunnels (MPLS LSPs, IS-IS flexible algorithm, SR-TE) created from the service and border nodes. The LSPs may be signaled using different signaling protocols in different domains, and can be configured with different traffic engineering characteristics (class or color). The transport tunnel endpoint also acts as the service endpoint and can be present in the same tunnel domain as the service ingress node, or in an adjacent or non-adjacent domain. You can use BGP classful transport planes to resolve services over LSPs with certain traffic engineering characteristics either inside a single domain or across multiple domains.

BGP classful transport planes reuse the BGP-VPN technology, keeping the tunneling-domains loosely coupled and coordinated.

- The network layer reachability information (NLRI) is **RD:TunnelEndpoint** used for path-hiding.
- The route target indicates the transport class of the LSPs, and leaks routes to the corresponding transport RIB on the destination device.
- Every transport tunneling protocol installs an ingress route into the transport-class.inet.3 routing table, models the tunnel transport class as a VPN route target, and collects the LSPs of the same transport class in the transport-class.inet.3 transport-rib routing table.
- Routes in this routing instance are advertised in the BGP classful transport plane (inet transport) AFI-SAFI following procedures similar to RFC-4364.
- When crossing inter-AS link boundary, you must follow Option-b procedures to stitch the transport tunnels in these adjacent domains.

  Similarly, when crossing intra-AS regions you must follow Option-b procedures to stitch the transport tunnels in the different TE-domains.
- You can define resolution schemes to specify the intent on the variety of transport classes in a fallback order.

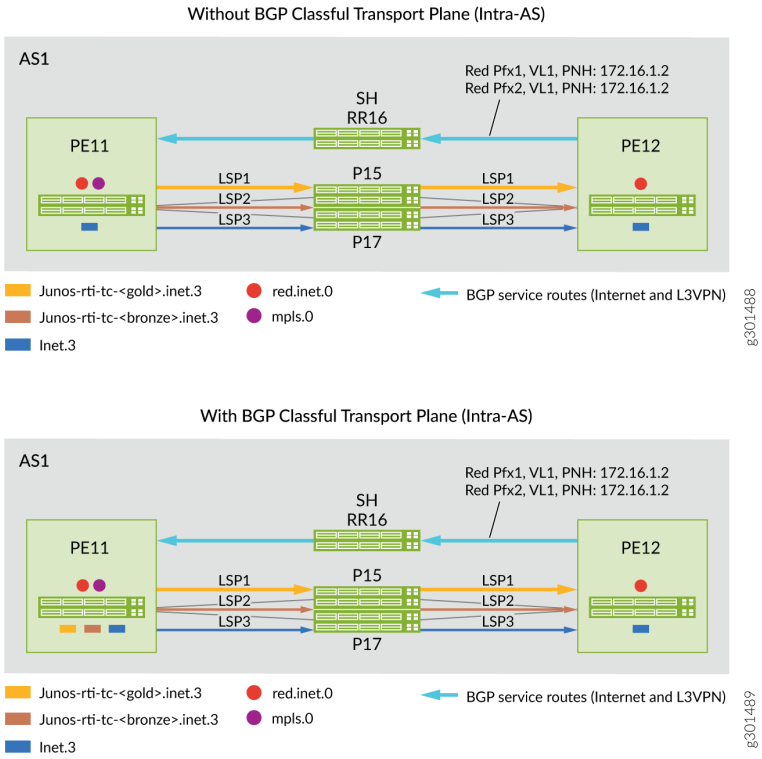
- You can resolve service routes and BGP classful transport routes over these transport classes, by carrrying the mapping community on them.

The BGP classful transport family runs along side the BGP-LU transport layer family. In a seamless MPLS network running BGP-LU, meeting stringent SLA requirements of 5G is a challenge as the traffic engineering characteristics of the tunnels are not known or preserved across domain boundaries. BGP classful transport planes provide operationally easy and scalable means to advertise multiple paths for remote loopbacks along with the transport class information in the seamless MPLS architecture. In BGP classful transport family routes, different SLA paths are represented using Transport Route-Target extended community, which carries the transport class color. This Transport Route-Target is used by the receiving BGP routers to associate the BGP classful transport route with the appropriate transport class. When re-advertising the BGP classful transport routes, MPLS swaps routes, inter connect the intra-AS tunnels of the same transport class, thereby forming an end-to-end tunnel that preserves the transport class.

Intra-AS Implementation of BGP Classful Transport Planes

Figure 113 on page 1764 illustrates a network topology with before-and-after scenarios of implementing BGP classful transport planes in an intra-AS domain. Devices PE11 and PE12 use RSVP LSPs as the transport tunnel and all transport tunnel routes are installed in inet.3 RIB. Implementing BGP classful transort planes enables RSVP transport tunnels to be color-aware similar to segment routing tunnels.

Figure 113: Intra-AS Domain: Before-and-After Scenarios For BGP Classful Transport Planes Implementation



To classify transport tunnels into BGP transport class in an intra-AS setup:

1. Define the transport class at the service node (ingress PE devices), for example, gold and bronze, and assign color community values to the defined transport class.

   Sample configuration:

```
pe11# show routing-options
route-distinguisher-id 172.16.1.1;
transport-class {
    name gold {
        color 100;
    }
    name bronze {
        color 200;
```

2. Associate the transport tunnel to a specific transport class at the ingress node of the tunnels.

Sample configuration:

```
pe11# show protocols mpls
label-switched-path toPE12-bronze {
    transport-class bronze;
}
label-switched-path toPE12-gold {
    transport-class gold;
}
```

Intra-AS BGP classful transport plane functionality:

- BGP classful transport creates predefined transport RIBs per named transport class (gold and bronze) and auto derives mapping community from its color value (100 and 200).
- Intra-AS transport routes are populated in transport RIBs by the tunneling protocol when it is associated with a transport class.

  In this example, RSVP LSP routes associated with transport class gold (color 100) and transport class bronze (color 200) are installed in the transport RIBs **junos-rti-tc-<100>.inet.3** and **junos-rti-tc-<200>.inet.3**, respectively.
- Service node (ingress PEs) match extended color community (color:0:100 and color:0:200) of service route against the mapping community in predefined resolution RIBs and resolve the protocol next hop (PNH) in corresponding transport RIBs (either junos-rti-tc-<100>.inet.3, or junos-rti-tc-<200>.inet.3).
- BGP routes bind to a resolution scheme by carrying the assiocaited mapping community.
- Each transport class automatically creates two predefined resolution schemes and automatically derives the mapping community.

  One resolution scheme is for resolving service routes that use **Color:0:<val>** as the mapping community.

  The other resolution scheme is for resolving transport routes that use **Transport-Target:0:<val>** as the mapping community.
- If service route PNH cannot be resolved using RIBs listed in the predefined resolution scheme, then it can fall back to the inet.3 routing table.
- You can also configure fallback between different colored transport RIBs by using user-defined resolution schemes under the **[edit routing-options resolution scheme]** configuration hierarchy.

Inter-AS Implementation of BGP Classful Transport Planes

In an inter-AS network, BGP-LU is converted to BGP classful transport network after configuring a minimum of two transport classes (gold and bronze) on all service nodes or PE devices and border nodes (ABRs and ASBRs).

To convert the transport tunnels into BGP classful transport:

1. Define transport class at the service nodes (ingress PE devices) and the border nodes (ABRs and ASBRs), for example, gold and broze.

   Sample configuration:

```
pe11# show routing-options
route-distinguisher-id 172.16.1.1;
transport-class {
    name gold {
        color 100;
    }
    name bronze {
        color 200;
```

2. Associate the transport tunnels to a specific transport class at the ingress node of the tunnels (ingress PEs, ABRs,and ASBRs).

   Sample configuration:

   For RSVP LSPs

```
abr23# show protocols mpls
label-switched-path toASBR21-bronze {
    transport-class bronze;
}
label-switched-path toASBR22-gold {
    transport-class gold;
```

   For IS-IS flxible algorithm

```
asbr13# show routing-options
flex-algorithm 128 {
…
color 100;
use-transport-class;
```

```
}
flex-algorithm 129 {
…
color 200;
use-transport-class;
}
```

3. Enable new family for the BGP classful transport (inet transport) and BGP-LU (inet labeled-unicast) in the network.

   Sample configuration:

```
abr23# show protocols bgp
group toAs2-RR27 {
    family inet {
        labeled-unicast {
…
        }
        transport {
…
    }
    cluster 172.16.2.3;
    neighbor 172.16.2.7;
}
```

4. Advertise service routes from the egress PE device with appropriate extended color community.

   Sample configuration:

```
pe11# show policy-options policy-statement red
term 1 {
        from {
            route-filter 192.168.3.3/32 exact;
        }
        then {
            community add map2gold;
            next-hop self;
            accept;
        }
    }
    term 2 {
        from {
```

```
            route-filter 192.168.33.33/32 exact;
        }
        then {
            community add map2bronze;
            next-hop self;
            accept;
        }
    }
community map2bronze members color:0:200;
community map2gold members color:0:100;
```

Inter-AS BGP classful transport plane functionality:

1. BGP classful transport planes create predefined transport RIBs per named transport class (gold and bronze) and automatically derives mapping community from its color value.
2. Intra-AS transport routes are populated in transport RIBs by tunneling protocols when associated with a transport class.

   For example, transport tunnel routes associated with the transport class gold and bronze are installed in the transport RIBs **junos-rti-tc-<100>.inet.3** and **junos-rti-tc-<200>.inet.3**, respectively.
3. BGP classful transport planes use unique Route Distinguisher and Route Target when it copies the transport tunnel routes from each transport RIB to the bgp.transport.3 routing table.
4. Border nodes advertise routes from bgp.transport.3 routing table to its peers in other domains if family inet transport is negotiated in the BGP session.
5. Receiving border node installs these bgp-ct routes in the bgp.transport.3 routing table and copies these routes based on the transport Route Target to the appropriate transport RIBs.
6. Service node matches the color community in the service route against a mapping community in the resolution schemes and resolves PNH in the corresponding transport RIB (either **junos-rti-tc-<100>.inet.3**, or **junos-rti-tc-<200>.inet.3**).
7. Border nodes use predefined resolution schemes for transport route PNH resolution.
8. Predefined or user defined, both resolution schemes support service route PNH resolution. Predefined uses inet.3 as fallback, and user-defined resolution scheme allows list of transport RIBs to be used in the order specified while resolving PNH.
9. If service route PNH cannot be resolved using RIBs listed in the user-defined resolution scheme, then route is discarded.

Improving Traffic Engineering Database Accuracy with RSVP PathErr Messages

IN THIS SECTION

- PathErr Messages | **1769**
- Identifying the Problem Link | **1770**
- Configuring the Router to Improve Traffic Engineering Database Accuracy | **1770**

An essential element of RSVP-based traffic engineering is the traffic engineering database. The traffic engineering database contains a complete list of all network nodes and links participating in traffic engineering, and a set of attributes each of those links can hold. (For more information about the traffic engineering database, see "Constrained-Path LSP Computation" on page 585.) One of the most important link attributes is bandwidth.

Bandwidth availability on links changes quickly as RSVP LSPs are established and terminated. It is likely that the traffic engineering database will develop inconsistencies relative to the real network. These inconsistencies cannot be fixed by increasing the rate of IGP updates.

Link availability can share the same inconsistency problem. A link that becomes unavailable can break all existing RSVP LSPs. However, its unavailability might not readily be known by the network.

When you configure the `rsvp-error-hold-time` statement, a source node (ingress of an RSVP LSP) learns from the failures of its LSP by monitoring PathErr messages transmitted from downstream nodes. Information from the PathErr messages is incorporated into subsequent LSP computations, which can improve the accuracy and speed of LSP setup. Some PathErr messages are also used to update traffic engineering database bandwidth information, reducing inconsistencies between the traffic engineering database and the network.

You can control the frequency of IGP updates by using the `update-threshold` statement. See "Configuring the RSVP Update Threshold on an Interface" on page 1244.

This section discusses the following topics:

PathErr Messages

PathErr messages report a wide variety of problems by means of different code and subcode numbers. You can find a complete list of these PathErr messages in RFC 2205, *Resource Reservation Protocol (RSVP), Version 1, Functional Specification* and RFC 3209, *RSVP-TE: Extensions to RSVP for LSP Tunnels*.

When you configure the `rsvp-error-hold-time` statement, two categories of PathErr messages, which specifically represent link failures, are examined:

- Link bandwidth is low for this LSP: Requested bandwidth unavailable—code 1, subcode 2

  This type of PathErr message represents a global problem that affects all LSPs transiting the link. They indicate that the actual link bandwidth is lower than that required by the LSP, and that it is likely that the bandwidth information in the traffic engineering database is an overestimate.

  When this type of error is received, the available link bandwidth is reduced in the local traffic engineering database, affecting all future LSP computations.

- Link unavailable for this LSP:

  - Admission Control failure—code 1, any subcode except 2
  - Policy Control failures—code 2
  - Service Preempted—code 12
  - Routing problem—no route available toward destination—code 24, subcode 5

  These types of PathErr messages are generally pertinent to the specified LSP. The failure of this LSP does not necessarily imply that other LSPs could also fail. These errors can indicate maximum transfer unit (MTU) problems, service preemption (either manually initiated by the operator or by another LSP with a higher priority), that a next-hop link is down, that a next-hop neighbor is down, or service rejection because of policy considerations. It is best to route this particular LSP away from the link.

Identifying the Problem Link

Each PathErr message includes the sender's IP address. This information is propagated unchanged toward the ingress router. A lookup in the traffic engineering database can identify the node that originated the PathErr message.

Each PathErr message carries enough information to identify the RSVP session that triggered the message. If this is a transit router, it simply forwards the message. If this router is the ingress router (for this RSVP session), it has the complete list of all nodes and links the session should traverse. Coupled with the originating node information, the link can be uniquely identified.

Configuring the Router to Improve Traffic Engineering Database Accuracy

To improve the accuracy of the traffic engineering database, configure the `rsvp-error-hold-time` statement. When this statement is configured, a source node (ingress of an RSVP LSP) learns from the failures of its LSP by monitoring PathErr messages transmitted from downstream nodes. Information from the PathErr messages is incorporated into subsequent LSP computations, which can improve the accuracy and speed

of LSP setup. Some PathErr messages also are used to update traffic engineering database bandwidth information, reducing inconsistencies between the traffic engineering database and the network.

To configure how long MPLS should remember RSVP PathErr messages and consider them in CSPF computation, include the `rsvp-error-hold-time` statement:

```
rsvp-error-hold-time seconds;
```

You can include this statement at the following hierarchy levels:

- `[edit protocols mpls]`
- `[edit logical-systems logical-system-name protocols mpls]`

The time can be a value from 1 to 240 seconds. The default is 25 seconds. Configuring a value of 0 disables the monitoring of PathErr messages.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
23.1R1	Starting in Junos OS Release 23.1R1, Junos OS enables BGP Link State BGP-LS NLRI to carry the confederation ID in TLV 512 when BGP confederation is enabled. The NLRI carries the confederation ID along with the member AS number in TLV 517 as defined in RFC 9086.
22.1R1	Starting in Junos OS Release 22.1 R1, we have added IPv6 prefixes and IPv6 adjacency SID MPLS support in the traffic engineering database (TED) and BGP Link-State (LS).
17.1R1	Link state distribution using BGP is supported on QFX10000 switches.

RELATED DOCUMENTATION

Basic MPLS Configuration | 46

Color-Based Traffic Engineering Configuration

SUMMARY

IN THIS SECTION

- BGP Classful Transport Planes Overview | **1772**
- Example: Configuring Classful Transport Planes (Intra-Domain) | **1781**
- Color-Based Mapping of VPN Services Overview | **1819**

BGP Classful Transport Planes Overview

IN THIS SECTION

- Benefits of BGP Classful Transport Planes | **1772**
- Terminology of BGP Classful Transport Planes | **1773**
- Understanding BGP Classful Transport Planes | **1774**
- Intra-AS Implementation of BGP Classful Transport Planes | **1775**
- Inter-AS Implementation of BGP Classful Transport Planes | **1778**

Benefits of BGP Classful Transport Planes

- **Network-slicing**–Service and transport layers are decoupled from each other, laying the foundation for network-slicing and virtualization with the end-to-end slicing across multiple domains, thereby significantly reducing the CAPEX.
- **Inter-domain interoperability**–Extends transport class deployment across co-operating domains so the different transport signaling protocols in each domain interoperate. Reconciles any differences between extended community namespaces that may be in use in each domain.
- **Colored resolution with fallback**–Enables resolution over colored tunnels (RSVP, IS-IS flexible algorithm) with flexible fallback options over best-effort tunnels or any other color tunnel.
- **Quality-of-service**–Customizes and optimizes the network to achieve the end-to-end SLA requirements.

- **Leveraging existing deployments**–Supports well deployed tunneling protocols like RSVP along with new protocols, such as IS-IS flexible algorithm, preserving ROI and reducing OPEX.

Terminology of BGP Classful Transport Planes

This section provides a summary of commonly used terms for understanding BGP classful transport plane.

- **Service node**–Ingress Provider Edge (PE) devices that send and receive service routes (Internet and Layer 3 VPN).
- **Border node**–Device at the connection point of different domains (IGP areas or ASs).
- **Transport node**–Device that sends and receives BGP-Labeled Unicast (LU) routes.
- **BGP-VPN**–VPNs built using RFC4364 mechanisms.
- **Route Target (RT)**–Type of extended community used to define VPN membership.
- **Route Distinguisher (RD)**–Identifier used to distinguish to which VPN or virtual private LAN service (VPLS) a route belongs. Each routing instance must have a unique route distinguisher associated with it.
- **Resolution scheme**–Used to resolve protocol next-hop address (PNH) in resolution RIBs providing fallback.

  They map the routes to the different transport RIBs in the system based on mapping community.
- **Service family**–BGP address family used for advertising routes for data traffic, as opposed to tunnels.
- **Transport family** –BGP address family used for advertising tunnels, which are in turn used by service routes for resolution.
- **Transport tunnel**–A tunnel over which a service may place traffic, for example, GRE, UDP, LDP, RSVP, SR-TE, BGP-LU.
- **Tunnel domain**–A domain of the network containing service nodes and border nodes under a single administrative control that has a tunnel between them. An end-to-end tunnel spanning several adjacent tunnel domains can be created by stitching the nodes together using labels.
- **Transport class**–A group of transport tunnels offering the same type of service.
- **Transport class RT**–A new format of route target used to identify a specific transport class.

  A new format of Route Target used to identify a specific transport class.
- **Transport RIB**–At the service node and border node, a transport class has an associated transport RIB that holds its tunnel routes.

- **Transport RTI**–A routing instance; container of transport RIB, and associated transport class Route Target and Route Distinguisher.
- **Transport plane**–Set of transport RTIs importing same transport class RT. These are in turn stitched together to span across tunnel domain boundaries using a mechanism similar to Inter-AS option-b to swap labels at border nodes (nexthop-self), forming an end-to-end transport plane.
- **Mapping community**–Community on a service route that maps to resolve over a transport class.

Understanding BGP Classful Transport Planes

You can use BGP classful transport planes to configure transport classes for classifying a set of transport tunnels in an intra-AS network based on the traffic engineering characteristics and use these transport tunnels to map service routes with the desired SLA and intended fallback.

BGP classful transport planes can extend these tunnels to inter-domain networks that span across multiple domains (ASs or IGP areas) while preserving the transport class. To do this, you must configure the BGP classful transport transport layer BGP family between the border and service nodes.

In both inter-AS and intra-AS implementations, there can be many transport tunnels (MPLS LSPs, IS-IS flexible algorithm, SR-TE) created from the service and border nodes. The LSPs may be signaled using different signaling protocols in different domains, and can be configured with different traffic engineering characteristics (class or color). The transport tunnel endpoint also acts as the service endpoint and can be present in the same tunnel domain as the service ingress node, or in an adjacent or non-adjacent domain. You can use BGP classful transport planes to resolve services over LSPs with certain traffic engineering characteristics either inside a single domain or across multiple domains.

BGP classful transport planes reuse the BGP-VPN technology, keeping the tunneling-domains loosely coupled and coordinated.

- The network layer reachability information (NLRI) is **RD:TunnelEndpoint** used for path-hiding.
- The route target indicates the transport class of the LSPs, and leaks routes to the corresponding transport RIB on the destination device.
- Every transport tunneling protocol installs an ingress route into the transport-class.inet.3 routing table, models the tunnel transport class as a VPN route target, and collects the LSPs of the same transport class in the transport-class.inet.3 transport-rib routing table.
- Routes in this routing instance are advertised in the BGP classful transport plane (inet transport) AFI-SAFI following procedures similar to RFC-4364.
- When crossing inter-AS link boundary, you must follow Option-b procedures to stitch the transport tunnels in these adjacent domains.

  Similarly, when crossing intra-AS regions you must follow Option-b procedures to stitch the transport tunnels in the different TE-domains.

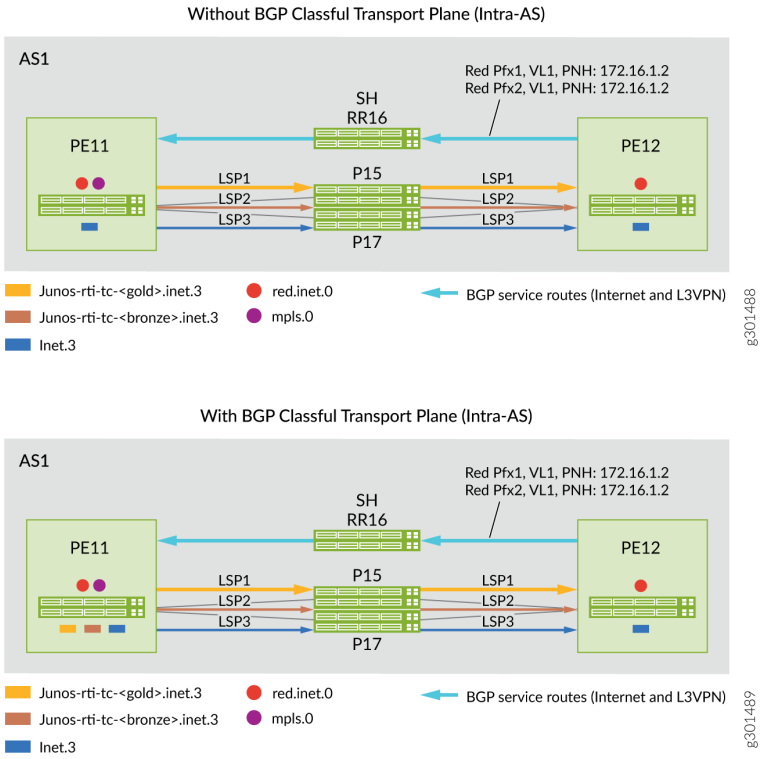
- You can define resolution schemes to specify the intent on the variety of transport classes in a fallback order.
- You can resolve service routes and BGP classful transport routes over these transport classes, by carrrying the mapping community on them.

The BGP classful transport family runs along side the BGP-LU transport layer family. In a seamless MPLS network running BGP-LU, meeting stringent SLA requirements of 5G is a challenge as the traffic engineering characteristics of the tunnels are not known or preserved across domain boundaries. BGP classful transport planes provide operationally easy and scalable means to advertise multiple paths for remote loopbacks along with the transport class information in the seamless MPLS architecture. In BGP classful transport family routes, different SLA paths are represented using Transport Route-Target extended community, which carries the transport class color. This Transport Route-Target is used by the receiving BGP routers to associate the BGP classful transport route with the appropriate transport class. When re-advertising the BGP classful transport routes, MPLS swaps routes, inter connect the intra-AS tunnels of the same transport class, thereby forming an end-to-end tunnel that preserves the transport class.

Intra-AS Implementation of BGP Classful Transport Planes

Figure 114 on page 1776 illustrates a network topology with before-and-after scenarios of implementing BGP classful transport planes in an intra-AS domain. Devices PE11 and PE12 use RSVP LSPs as the transport tunnel and all transport tunnel routes are installed in inet.3 RIB. Implementing BGP classful transort planes enables RSVP transport tunnels to be color-aware similar to segment routing tunnels.

Figure 114: Intra-AS Domain: Before-and-After Scenarios For BGP Classful Transport Planes Implementation



To classify transport tunnels into BGP transport class in an intra-AS setup:

1. Define the transport class at the service node (ingress PE devices), for example, gold and bronze, and assign color community values to the defined transport class.

   Sample configuration:

```
pe11# show routing-options
route-distinguisher-id 172.16.1.1;
transport-class {
    name gold {
        color 100;
    }
    name bronze {
        color 200;
```

2. Associate the transport tunnel to a specific transport class at the ingress node of the tunnels.

Sample configuration:

```
pe11# show protocols mpls
label-switched-path toPE12-bronze {
    transport-class bronze;
}
label-switched-path toPE12-gold {
    transport-class gold;
}
```

Intra-AS BGP classful transport plane functionality:

- BGP classful transport creates predefined transport RIBs per named transport class (gold and bronze) and auto derives mapping community from its color value (100 and 200).
- Intra-AS transport routes are populated in transport RIBs by the tunneling protocol when it is associated with a transport class.

  In this example, RSVP LSP routes associated with transport class gold (color 100) and transport class bronze (color 200) are installed in the transport RIBs **junos-rti-tc-<100>.inet.3** and **junos-rti-tc-<200>.inet.3**, respectively.
- Service node (ingress PEs) match extended color community (color:0:100 and color:0:200) of service route against the mapping community in predefined resolution RIBs and resolve the protocol next hop (PNH) in corresponding transport RIBs (either junos-rti-tc-<100>.inet.3, or junos-rti-tc-<200>.inet.3).
- BGP routes bind to a resolution scheme by carrying the assiocaited mapping community.
- Each transport class automatically creates two predefined resolution schemes and automatically derives the mapping community.

  One resolution scheme is for resolving service routes that use **Color:0:<val>** as the mapping community.

  The other resolution scheme is for resolving transport routes that use **Transport-Target:0:<val>** as the mapping community.
- If service route PNH cannot be resolved using RIBs listed in the predefined resolution scheme, then it can fall back to the inet.3 routing table.
- You can also configure fallback between different colored transport RIBs by using user-defined resolution schemes under the **[edit routing-options resolution scheme]** configuration hierarchy.

Inter-AS Implementation of BGP Classful Transport Planes

In an inter-AS network, BGP-LU is converted to BGP classful transport network after configuring a minimum of two transport classes (gold and bronze) on all service nodes or PE devices and border nodes (ABRs and ASBRs).

To convert the transport tunnels into BGP classful transport:

1. Define transport class at the service nodes (ingress PE devices) and the border nodes (ABRs and ASBRs), for example, gold and broze.

   Sample configuration:

```
pe11# show routing-options
route-distinguisher-id 172.16.1.1;
transport-class {
    name gold {
        color 100;
    }
    name bronze {
        color 200;
```

2. Associate the transport tunnels to a specific transport class at the ingress node of the tunnels (ingress PEs, ABRs,and ASBRs).

   Sample configuration:

   For RSVP LSPs

```
abr23# show protocols mpls
label-switched-path toASBR21-bronze {
    transport-class bronze;
}
label-switched-path toASBR22-gold {
    transport-class gold;
```

   For IS-IS flxible algorithm

```
asbr13# show routing-options
flex-algorithm 128 {
…
color 100;
use-transport-class;
```

```
}
flex-algorithm 129 {
…
color 200;
use-transport-class;
}
```

3. Enable new family for the BGP classful transport (inet transport) and BGP-LU (inet labeled-unicast) in the network.

   Sample configuration:

```
abr23# show protocols bgp
group toAs2-RR27 {
    family inet {
        labeled-unicast {
…
        }
        transport {
…
    }
    cluster 172.16.2.3;
    neighbor 172.16.2.7;
}
```

4. Advertise service routes from the egress PE device with appropriate extended color community.

   Sample configuration:

```
pe11# show policy-options policy-statement red
term 1 {
        from {
            route-filter 192.168.3.3/32 exact;
        }
        then {
            community add map2gold;
            next-hop self;
            accept;
        }
    }
    term 2 {
        from {
```

```
            route-filter 192.168.33.33/32 exact;
        }
        then {
            community add map2bronze;
            next-hop self;
            accept;
        }
    }
community map2bronze members color:0:200;
community map2gold members color:0:100;
```

Inter-AS BGP classful transport plane functionality:

1. BGP classful transport planes create predefined transport RIBs per named transport class (gold and bronze) and automatically derives mapping community from its color value.
2. Intra-AS transport routes are populated in transport RIBs by tunneling protocols when associated with a transport class.

   For example, transport tunnel routes associated with the transport class gold and bronze are installed in the transport RIBs **junos-rti-tc-<100>.inet.3** and **junos-rti-tc-<200>.inet.3**, respectively.
3. BGP classful transport planes use unique Route Distinguisher and Route Target when it copies the transport tunnel routes from each transport RIB to the bgp.transport.3 routing table.
4. Border nodes advertise routes from bgp.transport.3 routing table to its peers in other domains if family inet transport is negotiated in the BGP session.
5. Receiving border node installs these bgp-ct routes in the bgp.transport.3 routing table and copies these routes based on the transport Route Target to the appropriate transport RIBs.
6. Service node matches the color community in the service route against a mapping community in the resolution schemes and resolves PNH in the corresponding transport RIB (either **junos-rti-tc-<100>.inet.3**, or **junos-rti-tc-<200>.inet.3**).
7. Border nodes use predefined resolution schemes for transport route PNH resolution.
8. Predefined or user defined, both resolution schemes support service route PNH resolution. Predefined uses inet.3 as fallback, and user-defined resolution scheme allows list of transport RIBs to be used in the order specified while resolving PNH.
9. If service route PNH cannot be resolved using RIBs listed in the user-defined resolution scheme, then route is discarded.

Example: Configuring Classful Transport Planes (Intra-Domain)

IN THIS SECTION

- Before You Begin | **1781**
- Functional Overview | **1782**
- Topology Overview | **1784**
- Topology Illustrations | **1786**
- PE1 Configuration Steps | **1786**
- Verify Classful Transport Planes | **1790**
- Appendix 1: Troubleshooting | **1800**
- Appendix 2: Set Commands on All Devices | **1809**
- Appendix 3: Show Configuration Output on PE1 | **1815**

Before You Begin

Hardware and Software requirements	Junos OS Release 21.1R1 or later. **NOTE**: Only the provider edge routers (PE1 and PE2) require Junos OS Release support for the BGP-CT feature.
Estimated reading time	45 minutes
Estimated configuration time	1 hour

What to expect?	A working BGP-CT network with three service levels that map to diversely routed LSP paths. A Junos configuration that maps specific traffic (VPN customer routes) to the desired transport class using BGP color attribute extended communities. Basic LSP traffic engineering to force traffic classes on to diverse paths in the provider network

Business impact	Use this configuration example to configure and verify the BGP Classful Transport (BGP-CT) feature within a single autonomous network (intra-domain). BGP-CT maps customer routes to network paths that can be engineered to provide varying levels of performance. A typical use case for intra-domain BGP-CT is for a service provider to deploy BGP-CT to offer tiered VPN service levels to their customers.
Useful resources:	
Know more	To better understand BGP-CT, see BGP Classful Transport Planes Overview
Juniper vLabs	Visit the Juniper virtual labs (vLabs) to reserve a pre-configured sandbox. Use the sandbox to interact with and understand the BGP-CT feature. You'll find the "Classful Transport Planes (Intra-Domain Scenario)" demonstration in the routing section .
Learn more	Junos Class of Service (JCOS) On-Demand

Functional Overview

Table 35 on page 1782 provides a quick summary of the configuration components deployed in this example.

Table 35: Classful Transport Planes Functional Overview

Routing and Signaling protocols	
OSPF	All routers run OSPF as the IGP. All routers belong to area 0 (also called the backbone area). The single OSPF routing domain provides loopback connectivity in the provider network.

Internal and External BGP	The customer edge (CE) devices use EBGP peering to exchange routes with their provider edge device as part of a Layer 3 VPN service. The PE devices use IBGP to exchange IPv4 Layer 3 VPN routes with the remote PE. These routes also carry a color community used to map traffic to the correct data plane tunnel. Our example does not use a route reflector, instead opting for direct PE-PE peering. **NOTE**: The provider router (P router) does not run BGP. Its part of a BGP-free core to promote scaling. The P device uses MPLS label based switching to transport the customer VPN traffic between the PE devices.
RSVP	Each PE devices signals three LSPs to the remote PE. These LSPs map to the corresponding service classes of gold, bronze, and Best-Effort (BE). RSVP supports rich traffic engineering to force traffic onto desired paths in the provider network. These paths can in turn be engineered to provide varying Class of Service (CoS) handling need to enforce the SLA for each transport class. Our basic topology provides three paths between the PE devices. We use a named path with an ERO to ensure diverse routing of the LSPs over the core. Junos supports a rich set of capabilities for traffic engineering. **NOTE**: The classful transport feature is also supported with LSPs established through segment routing–traffic engineering (SR-TE) and IS-IS flex-algorithm tunnels.
MPLS	The provider network uses a MPLS based label switching data plane. The use of MPLS with TE paths ensures that each service class can be routed over disjoint paths with the desired performance levels. As noted above, MPLS also provides support for a BGP-free core.
Transport tunnels	

Three MPLS tunnels (LSPs) are established between the PE devices:	Each tunnel is assigned to the following transport classes: • Gold • Bronze • Best-effort This is the default transport class. This class provides best-effort (BE) level service. Customers that are not mapped to any specific transport class, or those that are mapped to a transport class that is down, default to the BE service class and the associated LSP path.
Service family	
Layer 3 VPN (`family inet-vpn unicast`	BGP-CT also works with other service families, such as BGP labeled Unicast, Flowspec, or Layer 2 VPN.
Primary verification tasks	
• Confirm overall network operation.	Verify working of IGP, RSVP, MPLS, BGP, and L3VPN.
• Verify mapping of Layer 3 VPN customer traffic to a transport class.	Modify the network to effect traffic steering between transport class tunnels to simulate the failure of a service tunnel and subsequent fail over to the BE path/class.

Topology Overview

This configuration example is based on a simple MPLS-Based Layer 3 VPN with two customer edge (CE) devices that communicate over the service provider network. The network core has three provider (P) routers that transport the VPN customer traffic using labeled-based switching. The two provider edge (PE) devices provide a Layer 3 VPN service to their attached CEs. The PEs use RSVP signaled MPLS LSPs to transport VPN traffic over the core. See Example: Configure a Basic MPLS-Based Layer 3 VPN for background information on the operation and configuration of a MPLS-based L3VPN.

We focus on the left to right flow of traffic from CE1 to CE2, and how PE1 uses a BGP color community attached to routes learned from PE2 to map traffic sent to the remote CE over the desired LSP forwarding next hops. In our example, PE1 uses explicit route objects (ERO) to force the routing of these LSPs over diverse paths. We skip this step at PE2, allowing the LSPs to be routed based on IGP load balancing. In order to have traffic flow from CE1 to CE2, CE1 must have a route to reach CE2. The

routes for CE2 travel in the opposite direction of the traffic it attracts from CE1. That is, the route to CE2's loopback travels from right to left.

In our example, the gold service class LSP is constrained to the PE1-P1-PE2 path. The bronze service class uses the PE1-P2-PE2 path. The best-effort LSP is routed along the PE1-P3-PE2 path. The topology diagram uses colored links to represent the three paths.

In our example, we add the `protocols mpls icmp-tunneling` statement. This is to allow the CE devices to trace the path through the provider network, even when that path involves MPLS switching as is the case for the Layer 3 VPN traffic. This option helps you confirm the expected forwarding path as a function of transport class is used.

Table 36 on page 1785 describes the role and functionality of each device in the context of this topology. Click on any device name to view its quick configuration.

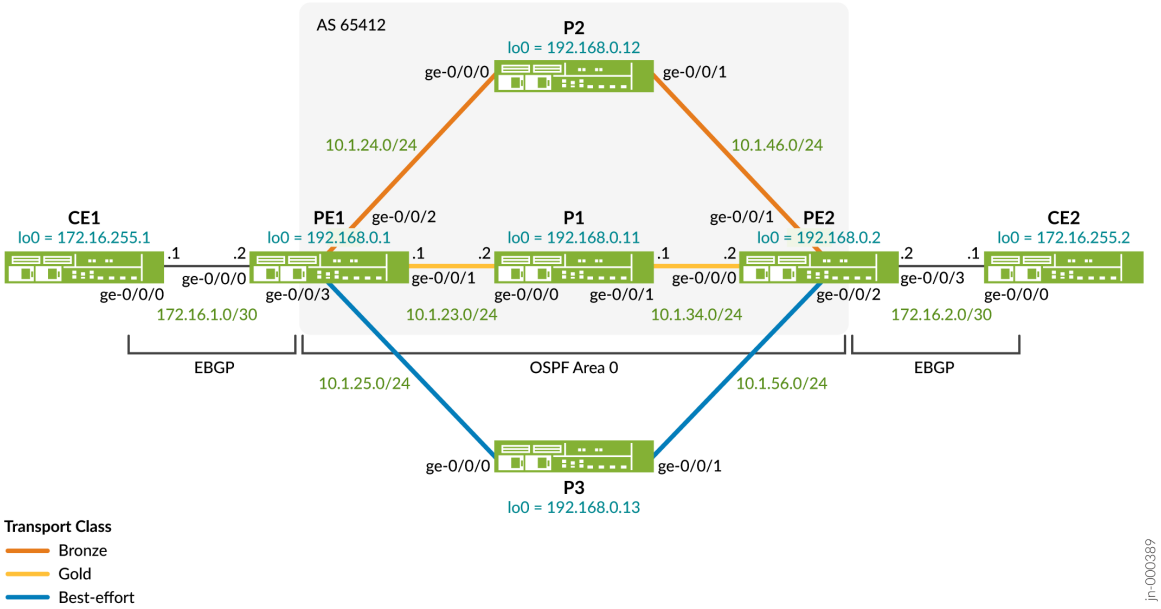
Table 36: Intra-Domain Classful Transport Planes Topology Overview

Device Name	Role	Function
"CE1" on page 1809	Local CE device (R1) .	EBGP peer to PE1 router to advertise and learn CE device loopback addresses. Test service connectivity with pings to the loopback address of CE2.
"CE2" on page 1810	Remote CE device (R7)	EBGP peering to PE2 router to advertise and learn CE device loopback addresses. Configures and attaches the color mapping community.
"PE1 (DUT)" on page 1810	local PE device (R2).	PE1 maps the color tagged service routes that originate at CE2 to a cosponsoring transport class (TC). PE1 receives the color tags routes over its IBGP session to PE2. In this example PE1 uses ERO based constraints to force diverse routing of its three LSPs over the provider's core.

"PE2" on page 1812	Remote PE device (R6).	PE2 re-advertises the color tagged routes received by CE2 to PE1 using IBGP. These routes uses the `inet-vpn` family to support a Layer 3 VPN services with color mapped TCs.
"P1" on page 1813"P2" on page 1814"P3" on page 1814	Provider devices P1, P2, and P3 (R3, R4, and R5).	The P1-P3 devices represent the service provider's core network. These are pure transit devices that perform MPLS label switching to transport the CE traffic sent over the L3 VPN.

Topology Illustrations

Figure 115: Service Mapping Using Classful Transport Planes Within a Network Domain



PE1 Configuration Steps

For information about navigating the CLI, see Using the CLI Editor in Configuration Mode

NOTE: For complete configuration on all devices see:

- "Appendix 2: Set Commands on All Devices" on page 1809
- "Appendix 3: Show Configuration Output on PE1" on page 1815

This section highlights the main configuration tasks needed to configure the PE1 device for this example. The first step is common to configuring a basic Layer 3 VPN service. The following set of steps are specific to adding the BGP-CT feature to your Layer 3 VPN. Both PE devices have a similar configuration, here we focus on PE1.

1. First, provision the general Layer 3 VPN:

   a. Configure and number the loopback, core facing, and CE-facing interfaces for IPv4. Be sure to enable the `mpls` family on the core-facing interfaces connecting to the P devices to support MPLS switching.

   b. Configure an autonomous system number.

   c. Configure single area OSPF on the loopback and core-facing interfaces.

   d. Configure RSVP on the loopback and core-facing interfaces.

   e. Configure the IBGP peering session to the remote PE device, PE2. Include the `inet-vpn` address family to support an IPv4 Layer 3 VPN.

   f. Configure a VRF based routing-instance for the CE1 device. Use EBGP as the PE-CE routing protocol.

```
[edit]
set interfaces ge-0/0/1 unit 0 family inet address 10.1.23.1/24
set interfaces ge-0/0/1 unit 0 family mpls

set interfaces ge-0/0/2 unit 0 description "Link from PE1 to P2"
set interfaces ge-0/0/2 unit 0 family inet address 10.1.24.1/24
set interfaces ge-0/0/2 unit 0 family mpls

set interfaces ge-0/0/3 unit 0 description "Link from PE1 to P3"
set interfaces ge-0/0/3 unit 0 family inet address 10.1.25.1/24
set interfaces ge-0/0/3 unit 0 family mpls
```

```
set interfaces lo0 unit 0 family inet address 192.168.0.1/32
```

```
[edit]
set routing-instances CE1_L3vpn instance-type vrf
set routing-instances CE1_L3vpn protocols bgp group CE1 type external
set routing-instances CE1_L3vpn protocols bgp group CE1 peer-as 64510
set routing-instances CE1_L3vpn protocols bgp group CE1 neighbor 172.16.1.1
set routing-instances CE1_L3vpn interface ge-0/0/0.0
set routing-instances CE1_L3vpn route-distinguisher 192.168.0.1:12
set routing-instances CE1_L3vpn vrf-target target:65412:12
```

```
[edit]
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 192.168.0.1
set protocols bgp group ibgp family inet unicast
set protocols bgp group ibgp family inet-vpn unicast
set protocols bgp group ibgp neighbor 192.168.0.2

set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0

set protocols rsvp interface lo0.0
set protocols rsvp interface ge-0/0/1.0
set protocols rsvp interface ge-0/0/2.0
set protocols rsvp interface ge-0/0/3.0
```

```
[edit]
set routing-options route-distinguisher-id 192.168.0.1
set routing-options autonomous-system 65412
```

2. Add classful transport planes to the Layer 3 VPN.

   Configure the gold and bronze transport-classes.

This is a critical step in configuring the classful transport feature. These transport classes are mapped to RSVP signaled (and possibly traffic engineered) LSPs that traverse the provider core. The remote routes learned from CE2 are tagged with color communities that map to these transport classes, and in so doing, to the desired LSP between the PE devices.

```
[edit]
set routing-options transport-class name gold color 100
set routing-options transport-class name bronze color 200
set routing-options resolution preserve-nexthop-hierarchy
```

3. Configure three LSPs from PE1 to PE2 with constrained routing that forces each to traverse a different P router. Two of these LSPs map to the *gold* and *bronze* transport-classes. The gold LSP is routed through P1, the bronze through P2, and the best-effort through the P3 device.

   Once mapped to transport classes the service provider is able to place specific customer traffic, as indicated by a BGP color community, onto a specific LSP. With this color to LSP mapping the service provider can offer a tiered service with different SLAs.

   In our example we use a strict ERO to ensure the three LSPs are diversely routed over the three paths available in the topology.

```
[edit]
set protocols mpls label-switched-path lsp_to_pe2 to 192.168.0.2
set protocols mpls label-switched-path lsp_to_pe2 primary best-effort
set protocols mpls label-switched-path gold_lsp_to_pe2 to 192.168.0.2
set protocols mpls label-switched-path gold_lsp_to_pe2 preference 5
set protocols mpls label-switched-path gold_lsp_to_pe2 primary gold
set protocols mpls label-switched-path gold_lsp_to_pe2 transport-class gold
set protocols mpls label-switched-path bronze_lsp_to_pe2 to 192.168.0.2
set protocols mpls label-switched-path bronze_lsp_to_pe2 preference 5
set protocols mpls label-switched-path bronze_lsp_to_pe2 primary bronze
set protocols mpls label-switched-path bronze_lsp_to_pe2 transport-class bronze
set protocols mpls path gold 10.1.23.2 strict
set protocols mpls path bronze 10.1.24.2 strict
set protocols mpls path best-effort 10.1.25.2 strict
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols mpls interface ge-0/0/3.0
```

4. To facilitate fallback to the default service class (best-effort) tunnel, we configure the gold and bronze transport class tunnels with a lower global preference. In this example the preference value is changed from the default 7 to 5. This allows the use of the best-effort tunnel as a fallback in the

event the gold or bronze tunnels become unusable. Setting a lower (more preferred) preference on the gold and bronze tunnels ensures they are selected for forwarding, even though the service route is able to resolve to the best-effort tunnel.

> **NOTE**: This section focused on the configuration needed on the PE1 device. It should be noted that for theclassful transport next-hop selection function to work at PE1, the remote CE2 device routes must be tagged with a color community. This tagging can occur on the remote PE2 device, or on the remote CE2 device. We show the latter approach here for completeness:

5. Match the color community tags added at the remote CE2 to the transport class definitions for the bronze and gold TCs.

```
[edit]
set policy-options policy-statement adv_direct term 1 from protocol direct
set policy-options policy-statement adv_direct term 1 from route-filter 172.16.0.0/16 orlonger
set policy-options policy-statement adv_direct term 1 then community add map2bronze
set policy-options policy-statement adv_direct term 1 then accept
set policy-options community map2bronze members color:0:200
set policy-options community map2gold members color:0:100
```

Verify Classful Transport Planes

IN THIS SECTION

- Verify Transport Classes and Transport Tunnels | **1791**
- Verify Next Hop Resolution Schemes | **1793**
- Verify Color Tagging and Next Hop Selection for CE2 Routes | **1795**
- Verify End-to-End Connectivity | **1797**
- Confirm Fail Over to Best-Effort | **1798**

> **NOTE**: In this section we focus on commands that demonstrate a working classful transport feature. See "Appendix 1: Troubleshooting" on page 1800 for commands used to verify the underlying functionality needed by the classful transport feature.

You'll use these commands to verify BGP classful transport works correctly.

Table 37: Classful Transport Planes Verification Commands

Command	Verification Task
show route resolution scheme	Display how service class routes are resolved to LSP next hops. Verify the resolution routing tables for a specific route.
show route receiving-protocol bgp *pe2-loopback-address*	Verify that the VPN routes received by PE1 have a BGP color community attached.
show route and show route forwarding-table vpn *vpn*	Verify transport tunnel selection by viewing the protocol nexthop (PNH) for the routes at PE1.
show mpls lsp statistics and show route forwarding-table	Verify the transport tunnel used by a specific transport class route.

Verify Transport Classes and Transport Tunnels

Purpose

PE1 and PE2 use RSVP-signaled MPLS transport tunnels to support a Layer 3 VPN service that is capable of offering differentiated service levels. These service routes have their next hops resolved to a specific MPLS tunnel based on the corresponding service class. The service class is signaled by attaching a BGP color community to VPN customer routes.

In this part you confirm that all three of PE1's LSPs are operational, that they are mapped to the correct transport class, and that they are correctly routed over the provider's core.

Action

From operational mode, enter the `show route 192.168.0.2` command.

```
user@PE1 show route 192.168.0.2
inet.0: 21 destinations, 21 routes (21 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.0.2/32     *[OSPF/10] 00:27:20, metric 2
                       to 10.1.24.2 via ge-0/0/2.0
```

```
                    >  to 10.1.25.2 via ge-0/0/3.0
                       to 10.1.23.2 via ge-0/0/1.0

inet.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.0.2/32     *[RSVP/7/1] 00:13:09, metric 2
                    >  to 10.1.25.2 via ge-0/0/3.0, label-switched-path lsp_to_pe2

junos-rti-tc-100.inet.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.0.2/32     *[RSVP/5/1] 00:13:11, metric 2
                    >  to 10.1.23.2 via ge-0/0/1.0, label-switched-path gold_lsp_to_pe2

junos-rti-tc-200.inet.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.0.2/32     *[RSVP/5/1] 00:13:08, metric 2
                    >  to 10.1.24.2 via ge-0/0/2.0, label-switched-path bronze_lsp_to_pe2
```

Meaning

The output confirms that PE1 has learned three paths to the loopback of PE2 through OSPF. These routes are in the `inet.0` table. You also note that all three LSPs are represented as viable next hops to reach PE2. Note that each of these LSPs is housed in a different routing table. The highlighted portion of the IP next hops (and the corresponding interface name) confirms the desired diverse LSP routing over the core. Traffic mapped to the gold path is sent to 10.1.23.2, while traffic for bronze and BE is sent to 10.1.24.2 and 10.1.25.2, respectively.

The following transport RIBs and transport tunnels are created.

- `junos-rti-tc-100.inet.3` for gold_lsp_to_pe2
- `junos-rti-tc-200.inet.3` for bronze_lsp_to_pe2
- `inet.3` for lsp_to_pe2

Verify Next Hop Resolution Schemes

Purpose

Verify the service routes resolution schemes, the associated mapping community, and how the next hop resolves over the contributing routing tables.

Action

From operational mode, enter the `show route resolution scheme all` command.

```
user@PE1> show route resolution scheme all
Resolution scheme: junos-resol-schem-tc-100-v4-service
  References: 1
  Mapping community: color:0:100
  Resolution Tree index 1, Nodes: 1
  Policy: [__resol-schem-common-import-policy__]
  Contributing routing tables: junos-rti-tc-100.inet.3 inet.3

Resolution scheme: junos-resol-schem-tc-100-v4-transport
  References: 1
  Mapping community: transport-target:0:100
  Resolution Tree index 3, Nodes: 1
  Policy: [__resol-schem-common-import-policy__]
  Contributing routing tables: junos-rti-tc-100.inet.3

Resolution scheme: junos-resol-schem-tc-100-v6-service
  References: 1
  Mapping community: color:0:100
  Resolution Tree index 2, Nodes: 0
  Policy: [__resol-schem-common-import-policy__]
  Contributing routing tables: junos-rti-tc-100.inet6.3 inet6.3

Resolution scheme: junos-resol-schem-tc-100-v6-transport
  References: 1
  Mapping community: transport-target:0:100
  Resolution Tree index 4, Nodes: 0
  Policy: [__resol-schem-common-import-policy__]
  Contributing routing tables: junos-rti-tc-100.inet6.3

Resolution scheme: junos-resol-schem-tc-200-v4-service
  References: 1
```

```
    Mapping community: color:0:200
    Resolution Tree index 5, Nodes: 1
    Policy: [__resol-schem-common-import-policy__]
    Contributing routing tables: junos-rti-tc-200.inet.3 inet.3

  Resolution scheme: junos-resol-schem-tc-200-v4-transport
    References: 1
    Mapping community: transport-target:0:200
    Resolution Tree index 7, Nodes: 1
    Policy: [__resol-schem-common-import-policy__]
    Contributing routing tables: junos-rti-tc-200.inet.3

  Resolution scheme: junos-resol-schem-tc-200-v6-service
    References: 1
    Mapping community: color:0:200
    Resolution Tree index 6, Nodes: 0
    Policy: [__resol-schem-common-import-policy__]
    Contributing routing tables: junos-rti-tc-200.inet6.3 inet6.3

  Resolution scheme: junos-resol-schem-tc-200-v6-transport
    References: 1
    Mapping community: transport-target:0:200
    Resolution Tree index 8, Nodes: 0
    Policy: [__resol-schem-common-import-policy__]
    Contributing routing tables: junos-rti-tc-200.inet6.3
```

Meaning

Focusing on the IPv4 portions of the output, you see the **junos-tc-100 (gold)** transport class has two resolution schemes - **junos-resol-schem-tc-100-v4-service** and **junos-resol-schem-tc-100-v4-transport** – used for the service and transport routes, respectively.

The resolution scheme for gold service routes (**junos-resol-schem-tc-100-v4-service**) provides resolution over *both* the **junos-rti-tc-100.inet.3** and the `inet.3` routing tables (highlighted in the sample output). Listing both the service and BE resolution tables is how fallback occurs when the service class is down. Recall this is why we altered the preference value of the service LSPs (setting the preference to 5 rather than the default 7), to ensure the service route is always preferred over the BE fallback.

Verify Color Tagging and Next Hop Selection for CE2 Routes

Purpose

Confirm that PE2 advertises the loopback route for CE2 with a color community that selects the bronze service class (color 200).

NOTE: In our example we configure the CE2 device to attach the color community. PE2 leaves this community in place when it re-advertises the route to PE1. This means the VPN customer is able to effect their own service class mappings. When desired the PE router can bleach or strip out any communities received from the CE. In this case the PE device needs to be configured to attach the desired color mapping community to CE routes before it re advertises them to PE1.

Action

From operational mode, enter the `show route receive-protocol bgp 192.168.0.2 172.16.255.2 detail` command.

```
user@PE1> show route receive-protocol bgp 192.168.0.2 172.16.255.2 detail
inet.0: 21 destinations, 21 routes (21 active, 0 holddown, 0 hidden)

CE1_L3vpn.inet.0: 5 destinations, 6 routes (5 active, 0 holddown, 0 hidden)
* 172.16.255.2/32 (1 entry, 1 announced)
     Import Accepted
     Route Distinguisher: 192.168.0.2:12
     VPN Label: 299808
     Nexthop: 192.168.0.2
     Localpref: 100
     AS path: 64520 I
     Communities: target:65412:12 color:0:200
```

Display the forwarding table entry for the CE2 loopback in the VPN routing instance at PE1. Confirm the forwarding next hop matches the desired transport class (bronze). Use the `show route forwarding-table vpn CE1_L3vpn destination 172.16.255.2 extensive` command.

```
user@PE1> show route forwarding-table vpn CE1_L3vpn destination 172.16.255.2 extensive
Routing table: CE1_L3vpn.inet [Index 10]
Internet:

Destination:  172.16.255.2/32
```

```
  Route type: user
  Route reference: 0                   Route interface-index: 0
  Multicast RPF nh index: 0
  P2mpidx: 0
  Flags: sent to PFE, prefix load balance
  Next-hop type: indirect              Index: 1048574  Reference: 2
  Nexthop:
  Next-hop type: composite             Index: 662      Reference: 2
  Load Balance Label: Push 299808, None
  Nexthop: 10.1.24.2
  Next-hop type: Push 299872           Index: 653      Reference: 2
  Load Balance Label: None
  Next-hop interface: ge-0/0/2.0
```

Meaning

The highlighted entries confirm traffic matching the CE2 loopback route is sent to 10.1.24.2 using the ge-0/0/2 interface. Recall from the EROs used for the LSPs, this interface and next hop is associated with the bronze LSP and transport class. The `299808` label is used to identify the service VRF. The outer RSVP transport label is `299872`.

You quickly confirm this is the correct RSVP transport label for the bronze class with a `show rsvp session detail name bronze_lsp_to_pe2` command

```
root@PE1> show rsvp session detail name bronze_lsp_to_pe2
Ingress RSVP: 3 sessions

192.168.0.2
  From: 192.168.0.1, LSPstate: Up, ActiveRoute: 0
  LSPname: bronze_lsp_to_pe2, LSPpath: Primary
  LSPtype: Static Configured
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 299872
  Resv style: 1 FF, Label in: -, Label out: 299872
  Time left:    -, Since: Tue Aug 16 12:17:12 2022
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 2 receiver 23256 protocol 0
  PATH rcvfrom: localclient
  Adspec: sent MTU 1500
  Path MTU: received 1500
  PATH sentto: 10.1.24.2 (ge-0/0/2.0) 1 pkts
  RESV rcvfrom: 10.1.24.2 (ge-0/0/2.0) 1 pkts, Entropy label: Yes
```

```
  Explct route: 10.1.24.2 10.1.46.2
  Record route: <self> 10.1.24.2 10.1.46.2
Total 1 displayed, Up 1, Down 0
```

The highlighted portions call out that the bronze LSP is routed through the P2 device and is associated with the indicated RSVP transport label (`299856`) you previously confirmed in the VPN forwarding table for the CE2 loopback address.

Verify End-to-End Connectivity

Purpose

Verify end-to-end connectivity across the provider's domain by pinging between CE1 to CE2. You examine MPLS traffic statistics to provide additional confirmation that the bronze transport class is used.

Action

From operational mode, enter the `ping` command.

```
user@CE1> ping 172.16.255.2 source 172.16.255.1 count 100 rapid
PING 172.16.255.2 (172.16.255.2): 56 data bytes
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!
--- 172.16.255.2 ping statistics ---
100 packets transmitted, 100 packets received, 0% packet loss
round-trip min/avg/max/stddev = 2.647/3.589/30.264/2.695 ms
```

From operational mode at PE1, enter the `show mpls lsp statistics` command.

```
user@PE1> show mpls lsp statistics
Ingress LSP: 3 sessions
To              From            State     Packets            Bytes LSPname
192.168.0.2     192.168.0.1     Up            100



                  8400 bronze_lsp_to_pe2
192.168.0.2     192.168.0.1     Up              0                0 gold_lsp_to_pe2
192.168.0.2     192.168.0.1     Up              0                0 lsp_to_pe2
```

```
Total 3 displayed, Up 3, Down 0
<output truncated for brevity>
```

Action

Trace the route from CE1 to CE2's loopback. Our configuration includes the `icmp-tunneling` statement to support an ICMP based trace route with MPLS hops in the provider core.

```
user@CE1> traceroute no-resolve 172.16.255.2
traceroute to 172.16.255.2 (172.16.255.2), 30 hops max, 52 byte packets
 1  172.16.1.2  2.174 ms  1.775 ms  1.917 ms
 2  10.1.24.2  5.171 ms  5.768 ms  4.900 ms
     MPLS Label=299872 CoS=0 TTL=1 S=0
     MPLS Label=299808 CoS=0 TTL=1 S=1
 3  10.1.46.2  4.707 ms  4.347 ms  4.419 ms
     MPLS Label=299808 CoS=0 TTL=1 S=1
 4  172.16.255.2  5.640 ms  5.851 ms  44.777 ms
```

Meaning

The ping exchange is successful and the statistics confirm use of the bronze transport tunnel. This is expected given the route to CE2 has the 200 color community attached. The trace route results confirm the traffic is forwarded over a LSP, and that this LSP is forwarding through 10.1.24.2. This is the IP address assigned to the P2 device. The forwarding next hop and outer label value confirm this traffic is sent on the bronze service class LSP.

Confirm Fail Over to Best-Effort

Purpose

Bring the bronze transport LSP down to verify that the traffic sent to CE2 fails over to the BE path.

Action

Enter configuration mode and specify an invalid next hop as an ERO for the bronze transport tunnel. The inability to satisfy the ERO requirement brings the related LSP down.

```
[edit]
user@PE1# set protocols mpls path bronze 10.1.66.6 strict
```

Once the change is committed the bronze tunnel is shown down:

```
root@PE1> show mpls lsp ingress
Ingress LSP: 3 sessions
To              From            State Rt P     ActivePath       LSPname
192.168.0.2     0.0.0.0         Dn     0       -                bronze_lsp_to_pe2
192.168.0.2     192.168.0.1     Up     0 *     gold             gold_lsp_to_pe2
192.168.0.2     192.168.0.1     Up     0 *     best-effort      lsp_to_pe2
```

Repeat the `ping` and trace route commands from CE1 to CE2's loopback.

```
root@CE1> ping 172.16.255.2 source 172.16.255.1 count 100 rapid
PING 172.16.255.2 (172.16.255.2): 56 data bytes
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!
--- 172.16.255.2 ping statistics ---
100 packets transmitted, 100 packets received, 0% packet loss
round-trip min/avg/max/stddev = 4.164/5.345/12.348/1.240 ms

root@CE1> traceroute no-resolve 172.16.255.2
traceroute to 172.16.255.2 (172.16.255.2), 30 hops max, 52 byte packets
 1  172.16.1.2  2.493 ms  1.766 ms  1.913 ms
 2  10.1.25.2  5.211 ms  5.016 ms  5.514 ms
     MPLS Label=299808 CoS=0 TTL=1 S=0
     MPLS Label=299808 CoS=0 TTL=1 S=1
 3  10.1.56.2  4.216 ms  4.467 ms  4.551 ms
     MPLS Label=299808 CoS=0 TTL=1 S=1
 4  172.16.255.2  5.492 ms  5.543 ms  5.112 ms
```

Again display MPLS statistics on PE1.

```
user@PE1> show mpls lsp statistics

root@PE1> show mpls lsp statistics
Ingress LSP: 3 sessions
To              From            State     Packets            Bytes LSPname
192.168.0.2     0.0.0.0         Dn             NA               NA bronze_lsp_to_pe2
192.168.0.2     192.168.0.1     Up              0                0 gold_lsp_to_pe2
192.168.0.2     192.168.0.1     Up            100             8400 lsp_to_pe2
```

```
Total 3 displayed, Up 2, Down 1
. . .
```

Meaning

The ping exchange still succeeds, albeit now on a best-effort path. On the PE the statistics confirm use of the best-effort transport tunnel. The trace route shows that PE1 now forwards to the 10.1.25.2 next hop through PE3. This confirms fail over from a colored transport class to the best-effort class in the event of transport tunnel failure.

NOTE: In this section we effected fail over to the BE class by bringing down the LSP mapped to the bronze service class. As an alternative, consider changing the EBGP export policy on the CE2 device to have it attach the gold (100) color community. With this approach you expect to see ping traffic from CE1 to CE2 taking the gold LSP rather than failing over to BE. The below does the trick at CE2 if you prefer this approach. Be sure to commit the changes at CE2.

```
[edit]
root@CE2# delete policy-options policy-statement adv_direct term 1 then community add
map2bronze
root@CE2# set policy-options policy-statement adv_direct term 1 then community add
map2gold
```

Appendix 1: Troubleshooting

Our verification section is based on an assumption that you have a working network, allowing the focus to be placed on confirming the operation of BGP-CT. The BGP-CT feature, in a MPLS-based Layer 3 VPN context, is dependent on a network with working interfaces, IGP, RSVP, MPLS, and BGP.

Table 38 on page 1801 provides guidance on what to look for if your BGP-CT solution is not working as expected. The table is structured from the bottom to the top, starting with basic interface connectivity and ending with successful BGP route exchange between the PE devices.

NOTE: As part of this example you configure a loopback address and router ID. If the device previously had a different RID it can take some time for things to stabilize. Changing the RID is very disruptive and not something that happens often. When in a lab environment its suggested that you issue the `restart routing` operational mode command on all devices right after committing the new RID.

Table 38: Troubleshooting Steps

Functional Layer	Verification Approach
Interfaces and IP addressing	Verify that all interfaces in your topology are operationally up. Verify you can ping both the local and remote end of each link. Like most networks, the protocols and services in this example require a working IPv4 infrastructure. root@PE1> **show interfaces terse | match "(ge-0/0/0|ge-0/0/1| ge-0/0/2|ge-0/0/3)"** ge-0/0/0 up up ge-0/0/0.0 up up inet 172.16.1.2/30 ge-0/0/1 up up ge-0/0/1.0 up up inet 10.1.23.1/24 ge-0/0/2 up up ge-0/0/2.0 up up inet 10.1.24.1/24 ge-0/0/3 up up ge-0/0/3.0 up up inet 10.1.25.1/24 root@PE1> **ping 10.1.23.2 count 1** PING 10.1.23.2 (10.1.23.2): 56 data bytes 64 bytes from 10.1.23.2: icmp_seq=0 ttl=64 time=2.951 ms --- 10.1.23.2 ping statistics --- 1 packets transmitted, 1 packets received, 0% packet loss round-trip min/avg/max/stddev = 2.951/2.951/2.951/0.000 ms root@PE1> **ping 172.16.1.1 routing-instance CE1_L3vpn count 1** PING 172.16.1.1 (172.16.1.1): 56 data bytes 64 bytes from 172.16.1.1: icmp_seq=0 ttl=64 time=2.755 ms --- 172.16.1.1 ping statistics --- 1 packets transmitted, 1 packets received, 0% packet loss round-trip min/avg/max/stddev = 2.755/2.755/2.755/0.000 ms

OSPF (IGP) Routing	Confirm all provider devices have all expected OSPF adjacencies. Use the `show ospf interfaces` and `show ospf neighbors` operational mode commands. Display the routes for the provider loopback addresses and confirm valid OSPF paths for all remote loopback addresses (`show route protocol ospf | match 192.168.0`). Ping from the local loopback to the remote loopback addresses of all provider routers.
	This example uses CSPF based LSPs. This requires that OSPF be configured with the `traffic-engieering` statement. If IS-IS is used as the IGP this statement is not needed.
	
	`root@PE1> **show ospf interface**`
	`Interface State Area DR ID BDR ID Nbrs`
	`ge-0/0/1.0 BDR 0.0.0.0 192.168.0.11 192.168.0.1 1`
	`ge-0/0/2.0 BDR 0.0.0.0 192.168.0.12 192.168.0.1 1`
	`ge-0/0/3.0 DR 0.0.0.0 192.168.0.1 192.168.0.13 1`
	`lo0.0 DRother 0.0.0.0 0.0.0.0 0.0.0.0 0`
	
	`root@PE1> **show ospf neighbor**`
	`Address Interface State ID Pri Dead`
	`10.1.23.2 ge-0/0/1.0 **Full** 192.168.0.11 128 34`
	`10.1.24.2 ge-0/0/2.0 **Full** 192.168.0.12 128 32`
	`10.1.25.2 ge-0/0/3.0 **Full** 192.168.0.13 128 37`
	
	`root@PE1> **show route protocol ospf| match 192.168.0**`
	`192.168.0.2/32 *[OSPF/10] 00:10:15, metric 2`
	`192.168.0.11/32 *[OSPF/10] 00:18:40, metric 1`
	`192.168.0.12/32 *[OSPF/10] 00:18:35, metric 1`
	`192.168.0.13/32 *[OSPF/10] 00:10:15, metric 1`
	`root@PE1> **ping 192.168.0.2 source 192.168.0.1 count 1**`
	`PING 192.168.0.2 (192.168.0.2): 56 data bytes`
	`64 bytes from 192.168.0.2: icmp_seq=0 ttl=63 time=3.045 ms`
	
	`--- 192.168.0.2 ping statistics ---`

	1 packets transmitted, 1 packets received, 0% packet loss round-trip min/avg/max/stddev = 3.045/3.045/3.045/0.000 ms

MPLS and RSVP	Verify all core interfaces are enabled for the `mpls` family. with a `show interfaces terse` command. Also verify that all provider interfaces are enabled under the `protocols mpls` and `protocols RSVP` hierarchies. Use the `show mpls interfaces` and `show rsvp interfaces` commands.
	NOTE: Be sure to confirm that the correct interface unit numbers are listed for the MPLS family and for each protocol. This example uses unit 0, which is the default unit number, on all interfaces.
	
	`root@PE1> **show rsvp interface**`
	`RSVP interface: 4 active`
	`Active Subscr- Static`
	`Available Reserved Highwater`
	`Interface State resv iption BW`
	`BW BW mark`
	`ge-0/0/1.0 Up 1 100% 1000Mbps`
	`1000Mbps 0bps 0bps`
	`ge-0/0/2.0 Up 1 100% 1000Mbps`
	`1000Mbps 0bps 0bps`
	`ge-0/0/3.0 Up 1 100% 1000Mbps`
	`1000Mbps 0bps 0bps`
	`lo0.0 Up 0 100% 0bps`
	`0bps 0bps 0bps`
	
	`root@PE1> **show mpls interface**`
	`Interface State Administrative groups (x: extended)`
	`ge-0/0/1.0 Up <none>`
	`ge-0/0/2.0 Up <none>`
	`ge-0/0/3.0 Up <none>`
	
	On the PE routers confirm that the LSPs are correctly defined to egress at the remote PE device's loopback address. Verify the EROs and any other TE constraints are valid. Use the `show mpls lsp` and `show rsvp session` commands.
	NOTE: Our examples uses CSPF based LSPs. This requires that the IGP supports a TE database (TED). If OSPF is the IGP be sure to include the `traffic-engieering` configuration statement. Alternatively, consider using the `no-cspf` statement in the LSP definition to remove CSPF from the equation.
	
	`root@PE1> **show mpls lsp**`
	`Ingress LSP: 3 sessions`
	`To From State Rt P ActivePath`

```
LSPname
192.168.0.2     192.168.0.1     Up      0 *     bronze
bronze_lsp_to_pe2
192.168.0.2     192.168.0.1     Up      0 *     gold
gold_lsp_to_pe2
192.168.0.2     192.168.0.1     Up      0 *     best-effort
lsp_to_pe2
Total 3 displayed, Up 3, Down 0

Egress LSP: 3 sessions
To              From            State   Rt Style Labelin
Labelout LSPname
192.168.0.1     192.168.0.2     Up       0  1 FF       3
- bronze_lsp_to_pe1
192.168.0.1     192.168.0.2     Up       0  1 FF       3
- gold_lsp_to_pe1
192.168.0.1     192.168.0.2     Up       0  1 FF       3
- lsp_to_pe1
Total 3 displayed, Up 3, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

BGP	Use the `show bgp summary`command on the PE devices to confirm both their EBGP session to the CE, and the IBGP session to the remote PE are established. If BGP is down despite being able to ping, suspect bad peer definition. Recall that loopback peering (for IBGP) requires the `local-address`statement. For EBGP specify directly connected next hops and confirm the local AS number, under `edit routing-options` and the remote AS number, under the EBGP peer group, are specified.
	Confirm the PE-PE session has the `inet-vpn unicast` family enabled. Use the `show route` command to confirm receipt of the remote CE route on the local PE. Use the `detail` switch to confirm proper color community attachment.
	
	`root@PE1> **show bgp summary**`
	`Threading mode: BGP I/O`
	`Default eBGP mode: advertise - accept, receive - accept`
	`Groups: 2 Peers: 2 Down peers: 0`
	`Table Tot Paths Act Paths Suppressed History Damp State Pending`
	`inet.0`
	`0 0 0 0 0 0 0`
	`bgp.l3vpn.0`
	`0 0 0 2 2 0 0`
	`Peer AS InPkt OutPkt OutQ Flaps Last Up/Dwn State|#Active/Received/Accepted/Damped...`
	`**172.16.1.1** 64510 55 55 0 0 23:13 **Establ**`
	`CE1_L3vpn.inet.0: 1/2/2/0`
	`**192.168.0.2** 65412 57 56 0 0 23:11 **Establ**`
	`inet.0: 0/0/0/0`
	`**bgp.l3vpn.0: 2/2/2/0**`
	`**CE1_L3vpn.inet.0: 2/2/2/0**`
	
	The show route advertising and receiving protocol commands are useful when confirming what routes a given BGP speaker advertises or receives, respectively.
	
	`root@PE1> **show route advertising-protocol bgp 192.168.0.2**`
	
	`CE1_L3vpn.inet.0: 5 destinations, 6 routes (5 active, 0 holddown, 0 hidden)`
	`Prefix Nexthop MED`

```
Lclpref    AS path
* 172.16.1.0/30           Self
100        I
* 172.16.255.1/32         Self
100        64510 I

root@PE1> show route receive-protocol bgp 192.168.0.2

inet.0: 21 destinations, 21 routes (21 active, 0 holddown, 0
hidden)

inet.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

CE1_L3vpn.inet.0: 5 destinations, 6 routes (5 active, 0
holddown, 0 hidden)
  Prefix                  Nexthop              MED
Lclpref    AS path
* 172.16.2.0/30           192.168.0.2
100        I
* 172.16.255.2/32         192.168.0.2
100        64520 I

junos-rti-tc-100.inet.3: 1 destinations, 1 routes (1 active, 0
holddown, 0 hidden)

junos-rti-tc-200.inet.3: 1 destinations, 1 routes (1 active, 0
holddown, 0 hidden)

mpls.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)

bgp.l3vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0
hidden)
  Prefix                  Nexthop              MED
Lclpref    AS path
  192.168.0.2:12:172.16.2.0/30
*                         192.168.0.2
100        I
  192.168.0.2:12:172.16.255.2/32
*                         192.168.0.2
100        64520 I
```

Layer 3 VPN	Ensure that the IBGP session supports `family inet-vpn` routes. Confirm the routes advertised by remote PE are imported into the correct instance based on the route target. Ensure the import and export policy used in the routing instance of each PE match on and advertise the correct routes. Some of the displays in the BGP verification section confirm the receipt of remote CE routes and the importation of those routes into the VRF instance.
	
	root@PE1> **show bgp neighbor 192.168.0.2 | match nlri**
	  NLRI for restart configured on peer: inet-unicast inet-vpn-unicast
	  NLRI advertised by peer: inet-unicast inet-vpn-unicast
	  NLRI for this session: inet-unicast inet-vpn-unicast
	root@PE1> show route table CE1_L3vpn.inet
	
	root@PE1> **show route receive-protocol bgp 192.168.0.2 172.16.255.2 detail**
	. . .
	CE1_L3vpn.inet.0: 5 destinations, 6 routes (5 active, 0 holddown, 0 hidden)
	* 172.16.255.2/32 (1 entry, 1 announced)
	  Import Accepted
	  Route Distinguisher: 192.168.0.2:12
	  VPN Label: 299776
	  Nexthop: **192.168.0.2**
	  Localpref: 100
	  AS path: 64520 I
	  Communities: **target:65412:12 color:0:200**
	
	root@PE1> **show route table CE1_L3vpn.inet**
	
	CE1_L3vpn.inet.0: 5 destinations, 6 routes (5 active, 0 holddown, 0 hidden)
	+ = Active Route, - = Last Active, * = Both
	
	172.16.1.0/30 *[Direct/0] 00:30:11
	  > via ge-0/0/0.0
	  [BGP/170] 00:29:57, localpref 100
	  AS path: 64510 I, validation-state: unverified
	  > to 172.16.1.1 via ge-0/0/0.0
	172.16.1.2/32 *[Local/0] 00:30:11
	  Local via ge-0/0/0.0
	172.16.2.0/30 *[BGP/170] 00:21:26, localpref 100, from 192.168.0.2
	  AS path: I, validation-state: unverified

	> to 10.1.25.2 **via ge-0/0/3.0, label-switched-path lsp_to_pe2** 172.16.255.1/32 *[BGP/170] 00:29:57, localpref 100 AS path: 64510 I, validation-state: unverified > to 172.16.1.1 via ge-0/0/0.0 **172.16.255.2/32** *[BGP/170] 00:29:40, localpref 100, from 192.168.0.2 AS path: 64520 I, validation-state: unverified > to 10.1.24.2 via ge-0/0/2.0, **label-switched-path bronze_lsp_to_pe2** Confirm the CE device is receiving and advertising the expected routes using the methods discussed for BGP troubleshooting.

Appendix 2: Set Commands on All Devices

IN THIS SECTION

- CE1 | **1809**
- CE2 | **1810**
- PE1 (DUT) | **1810**
- PE2 | **1812**
- P1 | **1813**
- P2 | **1814**
- P3 | **1814**

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level.

CE1

```
set interfaces ge-0/0/0 unit 0 description "Link from CE1 to PE1 for Layer 3 VPN"
set interfaces ge-0/0/0 unit 0 family inet address 172.16.1.1/30
set interfaces lo0 unit 0 family inet address 172.16.255.1/32
```

```
set policy-options policy-statement adv_direct term 1 from protocol direct
set policy-options policy-statement adv_direct term 1 from route-filter 172.16.0.0/16 orlonger
set policy-options policy-statement adv_direct term 1 then accept
set protocols bgp group ToPE1 type external
set protocols bgp group ToPE1 export adv_direct
set protocols bgp group ToPE1 peer-as 65412
set protocols bgp group ToPE1 neighbor 172.16.1.2
set routing-options router-id 172.16.255.1
set routing-options autonomous-system 64510
set system host-name CE1
```

CE2

```
set interfaces ge-0/0/0 unit 0 description "Link from CE2 to PE2 for Layer 3 VPN"
set interfaces ge-0/0/0 unit 0 family inet address 172.16.2.1/30
set interfaces lo0 unit 0 family inet address 172.16.255.2/32
set policy-options policy-statement adv_direct term 1 from protocol direct
set policy-options policy-statement adv_direct term 1 from route-filter 172.16.0.0/16 orlonger
set policy-options policy-statement adv_direct term 1 then community add map2bronze
set policy-options policy-statement adv_direct term 1 then accept
set policy-options community map2bronze members color:0:200
set policy-options community map2gold members color:0:100
set protocols bgp group PE2 type external
set protocols bgp group PE2 export adv_direct
set protocols bgp group PE2 peer-as 65412
set protocols bgp group PE2 neighbor 172.16.2.2
set routing-options router-id 172.16.255.2
set routing-options autonomous-system 64520
set system host-name CE2
```

PE1 (DUT)

```
set interfaces ge-0/0/0 unit 0 description "Link from PE1 to CE1"
set interfaces ge-0/0/0 unit 0 family inet address 172.16.1.2/30
set interfaces ge-0/0/1 unit 0 description "Link from PE1 to P1"
set interfaces ge-0/0/1 unit 0 family inet address 10.1.23.1/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 description "Link from PE1 to P2"
set interfaces ge-0/0/2 unit 0 family inet address 10.1.24.1/24
```

```
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 description "Link from PE1 to P3"
set interfaces ge-0/0/3 unit 0 family inet address 10.1.25.1/24
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.0.1/32
set routing-instances CE1_L3vpn instance-type vrf
set routing-instances CE1_L3vpn protocols bgp group CE1 type external
set routing-instances CE1_L3vpn protocols bgp group CE1 peer-as 64510
set routing-instances CE1_L3vpn protocols bgp group CE1 neighbor 172.16.1.1
set routing-instances CE1_L3vpn interface ge-0/0/0.0
set routing-instances CE1_L3vpn route-distinguisher 192.168.0.1:12
set routing-instances CE1_L3vpn vrf-target target:65412:12
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 192.168.0.1
set protocols bgp group ibgp family inet unicast
set protocols bgp group ibgp family inet-vpn unicast
set protocols bgp group ibgp neighbor 192.168.0.2
set protocols mpls icmp-tunneling
set protocols mpls label-switched-path lsp_to_pe2 to 192.168.0.2
set protocols mpls label-switched-path lsp_to_pe2 primary best-effort
set protocols mpls label-switched-path gold_lsp_to_pe2 to 192.168.0.2
set protocols mpls label-switched-path gold_lsp_to_pe2 preference 5
set protocols mpls label-switched-path gold_lsp_to_pe2 primary gold
set protocols mpls label-switched-path gold_lsp_to_pe2 transport-class gold
set protocols mpls label-switched-path bronze_lsp_to_pe2 to 192.168.0.2
set protocols mpls label-switched-path bronze_lsp_to_pe2 preference 5
set protocols mpls label-switched-path bronze_lsp_to_pe2 primary bronze
set protocols mpls label-switched-path bronze_lsp_to_pe2 transport-class bronze
set protocols mpls path gold 10.1.23.2 strict
set protocols mpls path bronze 10.1.24.2 strict
set protocols mpls path best-effort 10.1.25.2 strict
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols mpls interface ge-0/0/3.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0
set protocols rsvp interface lo0.0
set protocols rsvp interface ge-0/0/1.0
set protocols rsvp interface ge-0/0/2.0
set protocols rsvp interface ge-0/0/3.0
```

```
set routing-options route-distinguisher-id 192.168.0.1
set routing-options resolution preserve-nexthop-hierarchy
set routing-options router-id 192.168.0.1
set routing-options autonomous-system 65412
set routing-options transport-class name gold color 100
set routing-options transport-class name bronze color 200
set system host-name PE1
```

PE2

```
set interfaces ge-0/0/0 unit 0 description "Link from PE2 to P1"
set interfaces ge-0/0/0 unit 0 family inet address 10.1.36.2/24
set interfaces ge-0/0/0 unit 0 family mpls

set interfaces ge-0/0/1 unit 0 description "Link from PE2 to P2"
set interfaces ge-0/0/1 unit 0 family inet address 10.1.46.2/24
set interfaces ge-0/0/1 unit 0 family mpls

set interfaces ge-0/0/2 unit 0 description "Link from PE2 to P3"
set interfaces ge-0/0/2 unit 0 family inet address 10.1.56.2/24
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 description "Link from PE2 to CE2"
set interfaces ge-0/0/3 unit 0 family inet address 172.16.2.2/30
set interfaces lo0 unit 0 family inet address 192.168.0.2/32
set routing-instances CE2_L3vpn instance-type vrf
set routing-instances CE2_L3vpn protocols bgp group CE2 type external
set routing-instances CE2_L3vpn protocols bgp group CE2 peer-as 64520
set routing-instances CE2_L3vpn protocols bgp group CE2 neighbor 172.16.2.1
set routing-instances CE2_L3vpn interface ge-0/0/3.0
set routing-instances CE2_L3vpn route-distinguisher 192.168.0.2:12
set routing-instances CE2_L3vpn vrf-target target:65412:12
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 192.168.0.2
set protocols bgp group ibgp family inet unicast
set protocols bgp group ibgp family inet-vpn unicast
set protocols bgp group ibgp neighbor 192.168.0.1
set protocols mpls icmp-tunneling
set protocols mpls label-switched-path lsp_to_pe1 to 192.168.0.1
set protocols mpls label-switched-path gold_lsp_to_pe1 to 192.168.0.1
set protocols mpls label-switched-path gold_lsp_to_pe1 transport-class gold
set protocols mpls label-switched-path gold_lsp_to_pe1 preference 5
```

```
set protocols mpls label-switched-path bronze_lsp_to_pe1 to 192.168.0.1
set protocols mpls label-switched-path bronze_lsp_to_pe1 transport-class bronze
set protocols mpls label-switched-path bronze_lsp_to_pe1 preference 5
set protocols mpls interface ge-0/0/0.0
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols rsvp interface lo0.0
set protocols rsvp interface ge-0/0/0.0
set protocols rsvp interface ge-0/0/1.0
set protocols rsvp interface ge-0/0/2.0
set routing-options route-distinguisher-id 192.168.0.2
set routing-options router-id 192.168.0.2
set routing-options autonomous-system 65412
set routing-options transport-class name gold color 100
set routing-options transport-class name bronze color 200
set system host-name PE2
```

P1

```
set interfaces ge-0/0/0 unit 0 description "Link from P1 to PE1"
set interfaces ge-0/0/0 unit 0 family inet address 10.1.23.2/24
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 description "Link from P1 to PE2"
set interfaces ge-0/0/1 unit 0 family inet address 10.1.36.1/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.0.11/32
set protocols mpls icmp-tunneling
set protocols mpls interface ge-0/0/0.0
set protocols mpls interface ge-0/0/1.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols rsvp interface lo0.0
set protocols rsvp interface ge-0/0/0.0
set protocols rsvp interface ge-0/0/1.0
```

```
set routing-options router-id 192.168.0.11
set system host-name P1
```

P2

```
set interfaces ge-0/0/0 unit 0 description "Link from P2 to PE1"
set interfaces ge-0/0/0 unit 0 family inet address 10.1.24.2/24
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 description "Link from P2 to PE2"
set interfaces ge-0/0/1 unit 0 family inet address 10.1.46.1/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.0.12/32
set protocols mpls icmp-tunneling
set protocols mpls interface ge-0/0/0.0
set protocols mpls interface ge-0/0/1.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols rsvp interface lo0.0
set protocols rsvp interface ge-0/0/0.0
set protocols rsvp interface ge-0/0/1.0
set routing-options router-id 192.168.0.12
set system host-name P2
```

P3

```
set interfaces ge-0/0/0 unit 0 description "Link from P3 to PE1"
set interfaces ge-0/0/0 unit 0 family inet address 10.1.25.2/24
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 description "Link from P3 to PE2"
set interfaces ge-0/0/1 unit 0 family inet address 10.1.56.1/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.0.13/32
set protocols mpls icmp-tunneling
set protocols mpls interface ge-0/0/0.0
set protocols mpls interface ge-0/0/1.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
```

```
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols rsvp interface lo0.0
set protocols rsvp interface ge-0/0/0.0
set protocols rsvp interface ge-0/0/1.0
set routing-options router-id 192.168.0.13
set system host-name P3
```

Appendix 3: Show Configuration Output on PE1

IN THIS SECTION

- The Complete PE1 configuration in Curly Brace Format | **1815**

The Complete PE1 configuration in Curly Brace Format

```
user@PE1# show | no-more
system {
    host-name PE1;
}
interfaces {
    ge-0/0/0 {
        unit 0 {
            description "Link from PE1 to CE1";
            family inet {
                address 172.16.1.2/30;
            }
        }
    }
    ge-0/0/1 {
        unit 0 {
            description "Link from PE1 to P1";
            family inet {
                address 10.1.23.1/24;
            }
            family mpls;
        }
    }
    ge-0/0/2 {
```

```
        unit 0 {
            description "Link from PE1 to P2";
            family inet {
                address 10.1.24.1/24;
            }
            family mpls;
        }
    }
    ge-0/0/3 {
        unit 0 {
            description "Link from PE1 to P3";
            family inet {
                address 10.1.25.1/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 192.168.0.1/32;
            }
        }
    }
}
routing-instances {
    CE1_L3vpn {
        instance-type vrf;
        protocols {
            bgp {
                group CE1 {
                    type external;
                    peer-as 64510;
                    neighbor 172.16.1.1;
                }
            }
        }
        interface ge-0/0/0.0;
        route-distinguisher 192.168.0.1:12;
        vrf-target target:65412:12;
    }
}
routing-options {
```

```
    route-distinguisher-id 192.168.0.1;
    resolution {
        preserve-nexthop-hierarchy;
    }
    router-id 192.168.0.1;
    autonomous-system 65412;
    transport-class {
        name gold {
            color 100;
        }
        name bronze {
            color 200;
        }
    }
}
protocols {
    bgp {
        group ibgp {
            type internal;
            local-address 192.168.0.1;
            family inet {
                unicast;
            }
            family inet-vpn {
                unicast;
            }
            neighbor 192.168.0.2;
        }
    }
    mpls {
        label-switched-path lsp_to_pe2 {
            to 192.168.0.2;
            primary best-effort;
        }
        label-switched-path gold_lsp_to_pe2 {
            to 192.168.0.2;
            preference 5;
            primary gold;
            transport-class gold;
        }
        label-switched-path bronze_lsp_to_pe2 {
            to 192.168.0.2;
            preference 5;
```

```
            primary bronze;
            transport-class bronze;
        }
        path gold {
            10.1.23.2 strict;
        }
        path bronze {
            10.1.24.2 strict;
            10.1.66.6 strict;
        }
        path best-effort {
            10.1.25.2 strict;
        }
        icmp-tunneling;
        interface ge-0/0/1.0;
        interface ge-0/0/2.0;
        interface ge-0/0/3.0;
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface lo0.0;
            interface ge-0/0/1.0;
            interface ge-0/0/2.0;
            interface ge-0/0/3.0;
        }
    }
    rsvp {
        interface lo0.0;
        interface ge-0/0/1.0;
        interface ge-0/0/2.0;
        interface ge-0/0/3.0;
    }
}
```

SEE ALSO

- BGP Classful Transport Demonstration in Juniper Vlabs
- IETF Specification: BGP Classful Transport Planes

Color-Based Mapping of VPN Services Overview

IN THIS SECTION

- VPN Service Coloring | **1819**
- Specifying VPN Service Mapping Mode | **1822**
- Color-IP Protocol Next Hop Resolution | **1824**
- Fallback to IP Protocol Next Hop Resolution | **1825**
- BGP Labeled Unicast Color-based Mapping over SR-TE and IS-IS Underlay | **1825**
- Supported and Unsupported Features for Color-Based Mapping of VPN Services | **1825**

You can specify color as a protocol next hop constraint (in addition to the IPv4 or IPv6 address) for resolving transport tunnels over static colored and BGP segment routing traffic-engineered (SR-TE) LSPs. This is called the color-IP protocol next hop resolution, where you are required to configure a resolution-map and apply to the VPN services. With this feature, you can enable color-based traffic steering of Layer 2 and Layer 3 VPN services.

VPN Service Coloring

In general, a VPN service may be assigned a color on the egress router where the VPN NLRI is advertised, or on an ingress router where the VPN NLRI is received and processed.

You can assign a color to the VPN services at different levels:

- Per routing instance.
- Per BGP group.
- Per BGP neighbor.
- Per prefix.
- Set of prefixes.

Once you assign a color, the color is attached to a VPN service in the form of BGP color extended community.

You can assign multiple colors to a VPN service, referred to as multi-color VPN services. In such cases, the smallest color value is considered as the color of the VPN service, and all other colors are ignored.

Multiple colors are assigned by egress and/or ingress devices through multiple policies in the following order:

- BGP export policy on the egress device.
- BGP import policy on the ingress device.
- VRF import policy on the ingress device.

The two modes of VPN service coloring are:

Egress Color Assignment

In this mode, the egress device (that is, the advertiser of the VPN NLRI) is responsible for coloring the VPN service. To enable this mode, you can define a routing policy, and apply it in the VPN service's routing-instance vrf-export, group export, or group neighbor export at the [edit protocols bgp] hierarchy level. The VPN NLRI is advertised by BGP with the specified color extended community.

For example:

```
[edit policy-options]
community red-comm {
  members color:0:50;
}
```

```
[edit policy-options]
policy-statement pol-color {
  term t1 {
    from {
     [any match conditions];
    }
    then {
     community add red-comm;
     accept;
    }
  }
}
```

```
[edit routing-instances]
vpn-X {
...
```

```
  vrf-export pol-color ...;
}
```

OR

NOTE: When you apply the routing policy as an export policy of a BGP group or BGP neighbor, you must include the `vpn-apply-export` statement at the BGP, BGP group, or BGP neighbor level in order for the policy to take an effect on the VPN NLRI.

```
[edit protocols bgp]
group PEs {
...
  neighbor PE-A {
  export pol-color ...;
  vpn-apply-export;
  }
}
```

The routing policies are applied to Layer 3 VPN prefix NLRIs, Layer 2 VPN NRLIs, and EVPN NLRIs. The color extended community is inherited by all the VPN routes, imported, and installed in the target VRFs on one or multiple ingress devices.

Ingress Color Assignment

In this mode, the ingress device (that is, the receiver of the VPN NLRI) is responsible for coloring the VPN service. To enable this mode, you can define a routing policy, and apply it to the VPN service's routing-instance `vrf-import`, group import, or group neighbor import at the `[edit protocols bgp]` hierarchy level. All the VPN routes matching the routing policy is attached with the specified color extended community.

For example:

```
[edit policy-options]
community red-comm {
  members color:0:50;
}
```

```
[edit policy-options]
policy-statement pol-color {
```

```
  term t1 {
    from {
    [any match conditions];
}
then {
  community add red-comm;
    accept;
    }
  }
}
```

```
[edit routing-instances]
vpn-Y {
...
  vrf-import pol-color ...;
}
```

OR

```
[edit protocols bgp]
group PEs {
...
  neighbor PE-B {
    import pol-color ...;
  }
}
```

Specifying VPN Service Mapping Mode

To specify flexible VPN service mapping modes, you must define a policy using the resolution-map statement, and refer the policy in a VPN service's routing-instance vrf-import, group import, or group neighbor import at the [edit protocols bgp] hierarchy level. All the VPN routes matching the routing policy are attached with the specified resolution-map.

For example:

```
[edit policy-options]
resolution-map map-A {
  <mode-1>;
  <mode-2>;
```

```
  ...
}
policy-statement pol-resolution {
  term t1 {
    from {
    [any match conditions];
}
then {
  resolution-map map-A;
    accept;
    }
  }
}
```

You can apply import policy to the VPN service's routing-instance.

```
[edit routing-instances]
vpn-Y {
...
  vrf-import pol-resolution ...;
}
```

You can also apply the import policy to a BGP group or BGP neighbor.

```
[edit protocols bgp]
group PEs {
...
  neighbor PE-B {
  import pol-resolution ...;
  }
}
```

NOTE: Each VPN service mapping mode should have a unique name defined in the resolution-map. Only a single entry of IP-color is supported in the resolution-map, where the VPN route(s) are resolved using a colored-IP protocol next hop in the form of ip-address:color over the inetcolor.0 and inet6color.0 routing tables.

Color-IP Protocol Next Hop Resolution

The protocol next hop resolution process is enhanced to support colored-IP protocol next hop resolution. For a colored VPN service, the protocol next hop resolution process takes a color and a resolution-map, builds a colored-IP protocol next hop in the form of `ip-address:color`, and resolves the protocol next hop in the inetcolor.0 and inet6color.0 routing tables.

You must configure a policy to support multipath resolution of colored Layer 2 VPN, Layer 3 VPN, or EVPN services over colored LSPs. The policy must then be applied with the relevant RIB table as the resolver import policy.

For example:

```
[edit policy-options]
policy-statement mpath {
  then multipath-resolve;
}
```

```
[edit routing-options]
resolution {
  rib bgp.l3vpn.0 {
    inetcolor-import mpath;
  }
}

resolution {
  rib bgp.l3vpn-inet6.0 {
    inet6color-import mpath;
  }
}

resolution {
  rib bgp.l2vpn.0 {
    inetcolor-import mpath;
  }
}

resolution {
  rib mpls.0 {
    inetcolor-import mpath;
  }
}
```

```
resolution {
  rib bgp.evpn.0 {
    inetcolor-import mpath;
  }
}
```

Fallback to IP Protocol Next Hop Resolution

If a colored VPN service does not have a resolution-map applied to it, the VPN service ignores its color and falls back to the IP protocol next hop resolution. Conversely, if a non-colored VPN service has a resolution-map applied to it, the resolution-map is ignored, and the VPN service uses the IP protocol next hop resolution.

The fallback is a simple process from colored SR-TE LSPs to LDP LSPs, by using a RIB group for LDP to install routes in inet{6}color.0 routing tables. A longest prefix match for a colored-IP protocol next hop ensures that if a colored SR-TE LSP route does not exist, an LDP route with a matching IP address should be returned.

BGP Labeled Unicast Color-based Mapping over SR-TE and IS-IS Underlay

BGP Labeled Unicast (BGP-LU) can resolve IPv4 or IPv6 routes over segment routing–traffic engineering (SR-TE) with IS-IS underlay for both IPv4 and IPv6 address families. BGP-LU supports mapping a BGP community color and defining a `resolution map` for SR-TE. A colored protocol next hop is constructed and it is resolved on a colored SR-TE tunnel in the `inetcolor.0` or `inet6color.0` table. Thus BGP-LU resolves protocol next hop over SR-TE tunnels for packet transport. BGP uses `inet.3` and `inet6.3` tables for non-color based mapping.

Supported and Unsupported Features for Color-Based Mapping of VPN Services

The following features and functionality are supported with color-based mapping of VPN services:

- BGP Layer 3 VPN
- BGP Layer 2 VPN (Kompella Layer 2 VPN)
- BGP EVPN
- Resolution-map with a single IP-color option.
- Colored IPv4 and IPv6 protocol next hop resolution.
- Routing information base (also known as routing table) group based fallback to LDP LSP in inetcolor.0 or inet6color.0 routing tables.

- Colored SR-TE LSP.
- Virtual platforms.
- 64-bit Junos OS.
- Logical systems.
- BGP Labeled Unicast

The following features and functionality are not supported with color-based mapping of VPN services:

- Colored MPLS LSPs, such as RSVP, LDP, BGP-LU, static.
- Layer 2 circuit
- FEC-129 BGP auto-discovered and LDP-signaled Layer 2 VPN.
- VPLS
- MVPN
- IPv4 and IPv6 using resolution-map.

SEE ALSO

Understanding Static Segment Routing LSP in MPLS Networks

resolution-map

RELATED DOCUMENTATION

Basic MPLS Configuration | 46

DiffServ-Aware Traffic Engineering Configuration

IN THIS SECTION

DiffServ-Aware Traffic Engineering Introduction | 1827

DiffServ-Aware Traffic Engineering Terminology | 1827

- DiffServ-Aware Traffic Engineering Features | **1829**
- Configuring Link Down Notification for Optics Options Alarm or Warning | **1830**
- DiffServ-Aware Traffic Engineered LSPs Overview | **1830**
- DiffServ-Aware Traffic Engineered LSPs Operation | **1831**
- Configuring Routers for DiffServ-Aware Traffic Engineering | **1832**
- Configuring LSPs for DiffServ-Aware Traffic Engineering | **1837**

DiffServ-Aware Traffic Engineering Introduction

Differentiated Services (DiffServ)-aware traffic engineering provides a way to guarantee a specified level of service over an MPLS network. The routers providing DiffServ-aware traffic engineering are part of a differentiated services network domain. All routers participating in a differentiated services domain must have DiffServ-aware traffic engineering enabled.

To help ensure that the specified service level is provided, it is necessary to ensure that no more than the amount of traffic specified is sent over the differentiated services domain. You can accomplish this goal by configuring a policer to police or rate-limit the volume of traffic transiting the differentiated service domain. For more information about how to configure policers for label-switched paths (LSPs), see *Configuring Policers for LSPs*.

This feature can help to improve the quality of Internet services such as voice over IP (VoIP). It also makes it possible to better emulate an Asynchronous Transfer Mode (ATM) circuit over an MPLS network.

DiffServ-Aware Traffic Engineering Terminology

IN THIS SECTION

- Bandwidth model | **1828**
- CAC | **1828**
- Class type | **1828**
- Differentiated Services | **1828**
- Differentiated Services domain | **1828**
- DiffServ-aware traffic engineering | **1828**
- Multiclass LSP | **1828**
- MAM | **1829**

RDM | 1829
Traffic engineering class | 1829
Traffic engineering class map | 1829

Bandwidth model

The bandwidth model determines the values of the available bandwidth advertised by the interior gateway protocols (IGPs).

CAC

Call admission control (CAC) checks to ensure there is adequate bandwidth on the path before the LSP is established. If the bandwidth is insufficient, the LSP is not established and an error is reported.

Class type

A collection of traffic flows that is treated equivalently in a differentiated services domain. A class type maps to a queue and is much like a class-of-service (CoS) forwarding class in concept. It is also known as a traffic class.

Differentiated Services

Differentiated Services make it possible to give different treatment to traffic based on the EXP bits in the MPLS header. Traffic must be marked appropriately and CoS must be configured.

Differentiated Services domain

The routers in a network that have Differentiated Services enabled.

DiffServ-aware traffic engineering

A type of constraint-based routing. It can enforce different bandwidth constraints for different classes of traffic. It can also do CAC on each traffic engineering class when an LSP is established.

Multiclass LSP

A multiclass LSP functions like a standard LSP, but it also allows you to reserve bandwidth from multiple class types. The EXP bits of the MPLS header are used to distinguish between class types.

MAM

The maximum allocation bandwidth constraint model divides the available bandwidth between the different classes. Sharing of bandwidth between the class types is not allowed.

RDM

The Russian dolls bandwidth constraint model makes efficient use of bandwidth by allowing the class types to share bandwidth.

Traffic engineering class

A paired class type and priority.

Traffic engineering class map

A map between the class types, priorities, and traffic engineering classes. The traffic engineering class mapping must be consistent across the Differentiated Services domain.

DiffServ-Aware Traffic Engineering Features

DiffServ-aware traffic engineering provides the following features:

- Traffic engineering at a per-class level rather than at an aggregate level
- Different bandwidth constraints for different class types (traffic classes)
- Different queuing behaviors per class, allowing the router to forward traffic based on the class type

In comparison, standard traffic engineering does not consider CoS, and it completes its work on an aggregate basis across all Differentiated Service classes.

DiffServ-aware traffic engineering provides the following advantages:

- Traffic engineering can be performed on a specific class type instead of at the aggregate level.
- Bandwidth constraints can be enforced on each specific class type.
- It forwards traffic based on the EXP bits.

This makes it possible to guarantee service and bandwidth across an MPLS network. With DiffServ-aware traffic engineering, among other services, you can provide ATM circuit emulation, VoIP, and a guaranteed bandwidth service.

The following describes how the IGP, Constrained Shortest Path First (CSPF), and RSVP participate in DiffServ-aware traffic engineering:

- The IGP can advertise the unreserved bandwidth for each traffic engineering class to the other members of the differentiated services domain. The traffic engineering database stores this information.
- A CSPF calculation is performed considering the bandwidth constraints for each class type. If all the constraints are met, the CSPF calculation is considered successful.
- When RSVP signals an LSP, it requests bandwidth for specified class types.

Configuring Link Down Notification for Optics Options Alarm or Warning

To configure this option, include the `alarm` or `warning` statement at the `[edit interfaces ge-` *`fpc/pic/port`* `optics-options]` hierarchy level:

```
[edit interfaces]
ge-fpc/pic/port {
    optics-options {
        alarm alarm-name {
            (syslog | link-down);
            }
        warning warning-name {
            (syslog | link-down);
            }
    }
}
```

DiffServ-Aware Traffic Engineered LSPs Overview

A DiffServ-aware traffic engineered LSP is an LSP configured with a bandwidth reservation for a specific class type. This LSP can carry traffic for a single class type. On the packets, the class type is specified by the EXP bits (also known as the class-of-service bits) and the per-hop behavior (PHB) associated with the EXP bits. The mapping between the EXP bits and the PHB is static, rather than being signaled in RSVP.

The class type must be configured consistently across the Differentiated Services domain, meaning the class type configuration must be consistent from router to router in the network. You can unambiguously map a class type to a queue. On each node router, the class-of-service queue configuration for an interface translates to the available bandwidth for a particular class type on that link.

For more information about topics related to LSPs and DiffServ-aware traffic engineering, see the following:

- For forwarding classes and *class of service*, see the Junos OS Class of Service User Guide for Routing Devices.
- For EXP bits, see "MPLS Label Allocation" on page 524.
- For differentiated services, see RFC 3270, *Multi-Protocol Label Switching (MPLS) Support of Differentiated Services*.
- For information about how the IGPs and RSVP have been modified to support Differentiated Services-aware MPLS traffic engineering, see RFC 4124, *Protocol Extensions for Support of Differentiated-Service-Aware MPLS Traffic Engineering*.

DiffServ-Aware Traffic Engineered LSPs Operation

When configuring a DiffServ-aware traffic engineered LSP, you specify the class type and the bandwidth associated with it. The following occurs when an LSP is established with bandwidth reservation from a specific class type:

1. The IGPs advertise how much unreserved bandwidth is available for the traffic engineering classes.
2. When calculating the path for an LSP, CSPF is used to ensure that the bandwidth constraints are met for the class type carried by the LSP at the specified priority level.

   CSPF also checks to ensure that the bandwidth model is configured consistently on each router participating in the LSP. If the bandwidth model is inconsistent, CSPF does not compute the path (except for LSPs from class type ct0).
3. Once a path is found, RSVP signals the LSP using the Classtype object in the path message. At each node in the path, the available bandwidth for the class types is adjusted as the path is set up.

An LSP that requires bandwidth from a particular class (except class type ct0) cannot be established through routers that do not understand the Classtype object. Preventing the use of routers that do not understand the Classtype object helps to ensure consistency throughout the Differentiated Services domain by preventing the LSP from using a router that cannot support Differentiated Services.

By default, LSPs are signaled with setup priority 7 and holding priority 0. An LSP configured with these values cannot preempt another LSP at setup time and cannot be preempted.

It is possible to have both LSPs configured for DiffServ-aware traffic engineering and regular LSPs configured at the same time on the same physical interfaces. For this type of heterogeneous environment, regular LSPs carry best-effort traffic by default. Traffic carried in the regular LSPs must have the correct EXP settings (either by remarking the EXP settings or by assuming that the traffic arrived with the correct EXP settings from the upstream router).

Configuring Routers for DiffServ-Aware Traffic Engineering

IN THIS SECTION

- Configuring the Bandwidth Model | **1833**
- Configuring Traffic Engineering Classes | **1834**
- Configuring Class of Service for DiffServ-Aware Traffic Engineering | **1836**

To configure DiffServ-aware traffic engineering, include the `diffserv-te` statement:

```
diffserv-te {
    bandwidth-model {
        extended-mam;
        mam;
        rdm;
    }
    te-class-matrix {
        traffic-class {
            te number {
                priority priority;
                traffic-class ct number priority priority;
            }
        }
    }
}
```

You can include this statement at the following hierarchy levels:

- `[edit protocols mpls]`
- `[edit logical-systems logical-system-name protocols mpls]`

You must include the `diffserv-te` statement in the configuration on all routers participating in the Differentiated Services domain. However, you are not required to configure the traffic engineering class matrix (by including the `te-class-matrix` statement at the `[edit protocols mpls diffserv-te]` or `[edit logical-systems logical-system-name protocols mpls diffserv-te]` hierarchy level).

NOTE: To prevent the possibility of an incorrect configuration when migrating to Diffserv-aware traffic engineering, a policy control failure error might be triggered if there is conflict between the old LSPs and the newly configured TE-class matrix.

An old node might request an LSP with setup and hold priorities in such a way that the combination of the ct0 class and the priority does not match with the configured TE-class matrix. All LSPs on the router that are configured prior to configuring diffserv-aware traffic engineering are designated as being from class ct0.

The error appears in the RSVP tracing logs as a `Session preempted` error. For the router where the error originates, the error could appear as follows:

```
Jun 17 16:35:59 RSVP error for session 10.255.245.6(port/tunnel ID 31133)  Proto 0:
(class ct0, priority 2) is not a valid TE-class  Jun 17 16:35:59 RSVP originate
PathErr 192.168.37.22->192.168.37.23 Session preempted
```

For the router receiving the error, the error can appear as follows:

```
Jun 17 16:37:51 RSVP recv PathErr 192.168.37.22->192.168.37.23 Session  preempted LSP
to-f(2/31133)
```

To configure DiffServ-aware traffic engineering, complete the procedures in the following sections:

Configuring the Bandwidth Model

You must configure a bandwidth model on all routers participating in the Differentiated Services domain. The bandwidth models available are MAM, extended MAM, and RDM:

- Maximum allocation bandwidth constraints model (MAM)—Defined in RFC 4125, *Maximum Allocation Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering*.
- Extended MAM—A proprietary bandwidth model that behaves much like standard MAM. If you configure multiclass LSPs, you must configure the extended MAM bandwidth model.
- Russian-dolls bandwidth allocation model (RDM)—Makes efficient use of bandwidth by allowing the class types to share bandwidth. RDM is defined in RFC 4127, *Russian Dolls Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering*.

To configure a bandwidth model, include the `bandwidth-model` statement and specify one of the bandwidth model options:

```
bandwidth-model {
    extended-mam;
    mam;
    rdm;
}
```

You can include this statement at the following hierarchy levels:

- `[edit protocols mpls diffserv-te]`
- `[edit logical-systems logical-system-name protocols mpls diffserv-te]`

> **NOTE:** If you change the bandwidth model on an ingress router, all the LSPs enabled on the router are taken down and resignaled.

Configuring Traffic Engineering Classes

Configuring traffic engineering classes is optional. Table 39 on page 1834 shows the default values for everything in the traffic engineering class matrix. The default mapping is expressed in terms of the default forwarding classes defined in the CoS configuration.

Table 39: Default Values for the Traffic Engineering Class Matrix

Traffic Engineering Class	Class Type	Queue	Priority
te0	ct0	0	7
te1	ct1	1	7
te2	ct2	2	7
te3	ct3	3	7
te4	ct0	0	0

Table 39: Default Values for the Traffic Engineering Class Matrix *(Continued)*

Traffic Engineering Class	Class Type	Queue	Priority
te5	ct1	1	0
te6	ct2	2	0
te7	ct3	3	0

If you want to override the default mappings, you can configure traffic engineering classes 0 through 7. For each traffic engineering class, you configure a class type (or queue) from 0 through 3. For each class type, you configure a priority from 0 through 7.

To configure traffic engineering classes explicitly, include the `te-class-matrix` statement:

```
te-class-matrix {
    tenumber {
        priority priority;
        traffic-class {
            ctnumber priority priority;
        }
    }
}
```

You can include this statement at the following hierarchy levels:

- `[edit protocols mpls diffserv-te]`
- `[edit logical-systems logical-system-name protocols mpls diffserv-te]`

The following example shows how to configure traffic engineering class `te0` with class type `ct1` and a priority of `4`:

```
[edit protocols mpls diffserv-te]
te-class-matrix {
    te0 traffic-class ct1 priority 4;
}
```

NOTE: If you explicitly configure a value for one of the traffic engineering classes, all the default values in the traffic engineering class matrix are dropped.

When you explicitly configure traffic engineering classes, you must also configure a bandwidth model; otherwise, the configuration commit operation fails.

Requirements and Limitations for the Traffic Engineering Class Matrix

When you configure a traffic engineering class matrix, be aware of the following requirements and limitations:

- A mapping configuration is local and affects only the router on which it is configured. It does not affect other systems participating in the differentiated services domain. However, for a Differentiated Services domain to function properly, you need to configure the same traffic engineering class matrix on all the routers participating in the same domain.
- When explicitly configuring traffic engineering classes, you must configure the classes in sequence (`te0`, `te1`, `te2`, `te3`, and so on); otherwise, the configuration commit operation fails.

The first traffic engineering class you configure must be `te0`; otherwise, the configuration commit operation fails.

Configuring Class of Service for DiffServ-Aware Traffic Engineering

To configure DiffServ-aware traffic engineering, you must also configure class of service. The following example illustrates a class-of-service configuration that would allocate 25 percent of the link bandwidth to each class:

```
class-of-service {
    interfaces {
        all {
            scheduler-map simple-map;
        }
    }
    scheduler-maps {
        simple-map {
            forwarding-class assured-forwarding scheduler simple_sched;
                forwarding-class best-effort scheduler simple_sched;
                forwarding-class network-control scheduler simple_sched;
            forwarding-class expedited-forwarding scheduler simple_sched;
        }
    }
```

```
    schedulers {
        simple_sched {
            transmit-rate percent 25;
            buffer-size percent 25;
        }
    }
}
```

Configuring LSPs for DiffServ-Aware Traffic Engineering

IN THIS SECTION

- Configuring Class of Service for the Interfaces | **1838**
- Configuring IGP | **1838**
- Configuring Traffic-Engineered LSPs | **1838**
- Configuring Policing for LSPs | **1839**
- Configuring Fast Reroute for Traffic-Engineered LSPs | **1839**

You must configure the Differentiated Services domain (see "Configuring Routers for DiffServ-Aware Traffic Engineering" on page 1832) before you can enable DiffServ-aware traffic engineering for LSPs. The Differentiated Services domain provides the underlying class types and corresponding traffic engineering classes that you reference in the LSP configuration. The traffic engineering classes must be configured consistently on each router participating in the Differentiated Services domain for the LSP to function properly.

NOTE: You must configure either MAM or RDM as the bandwidth model when you configure DiffServ-aware traffic engineering for LSPs. See "Configuring the Bandwidth Model" on page 1832.

The actual data transmitted over this Differentiated Services domain is carried by an LSP. Each LSP relies on the EXP bits of the MPLS packets to enable DiffServ-aware traffic engineering. Each LSP can carry traffic for a single class type.

All the routers participating in the LSP must be Juniper Networks routers running Junos OS Release 6.3 or later. The network can include routers from other vendors and Juniper Networks routers running earlier versions of the Junos OS. However, the DiffServ-aware traffic engineering LSP cannot traverse these routers.

NOTE: You cannot simultaneously configure multiclass LSPs and DiffServ-aware traffic engineering LSPs on the same router.

To enable DiffServ-aware traffic engineering for LSPs, you need to configure the following:

Configuring Class of Service for the Interfaces

The existing class-of-service (CoS) infrastructure ensures that traffic that is consistently marked receives the scheduling guarantees for its class. The classification, marking, and scheduling necessary to accomplish this are configured using the existing Junos OS CoS features.

NOTE: The Junos OS does not support CoS on ATM interfaces.

For information about how to configure CoS, see the Junos OS Class of Service User Guide for Routing Devices.

Configuring IGP

You can configure either IS-IS or OSPF as the IGP. The IS-IS and OSPF configurations for routers supporting LSPs are standard. For information about how to configure these protocols, see the Junos OS Routing Protocols Library for Routing Devices.

Configuring Traffic-Engineered LSPs

You configure an LSP by using the standard LSP configuration statements and procedures. To configure DiffServ-aware traffic engineering for the LSP, specify a class type bandwidth constraint by including the `bandwidth` statement:

```
label-switched-path lsp-name {
    bandwidth {
        ctnumber bps;
    }
}
```

For a list of hierarchy levels at which you can include the `bandwidth` statement, see the statement summary sections for this statement.

If you do not specify a bandwidth for a class type, `ct0` is automatically specified as the queue for the LSP. You can configure only one class type for each LSP, unlike multiclass LSPs.

The class type statements specify bandwidth (in bits per second) for the following classes:

- `ct0`—Bandwidth reserved for class 0
- `ct1`—Bandwidth reserved for class 1
- `ct2`—Bandwidth reserved for class 2
- `ct3`—Bandwidth reserved for class 3

You can configure setup and holding priorities for an LSP, but the following restrictions apply:

- The combination of class and priority must be one of the configured traffic engineering classes. The default setup priority is 7 and the default holding priority is 0.
- Configuring an invalid combination of class type and priority causes the commit operation to fail.
- Automatic bandwidth allocation is not supported. If you configure automatic bandwidth allocation, the commit operation fails.
- LSPs configured with the `bandwidth` statement but without specifying a class type use the default class type `ct0`.
- For migration issues, see Internet draft draft-ietf-tewg-diff-te-proto-07.txt.

Configuring Policing for LSPs

Policing allows you to control the amount of traffic forwarded through a particular LSP. Policing helps to ensure that the amount of traffic forwarded through an LSP never exceeds the requested bandwidth allocation. You can configure multiple policers for each LSP.

For information about how to configure a policer for an LSP, see *Configuring Policers for LSPs*.

Configuring Fast Reroute for Traffic-Engineered LSPs

You can configure fast reroute for traffic engineered LSPs (LSPs carrying a single class of traffic). It is also possible to reserve bandwidth on the detour path for the class of traffic when fast reroute is enabled. The same class type number is used for both the traffic engineered LSP and its detour.

If you configure the router to reserve bandwidth for the detour path, a check is made to ensure that the link is capable of handling DiffServ-aware traffic engineering and for CoS capability before accepting it as a potential detour path. Unsupported links are not used.

You can configure the amount of bandwidth to reserve for detours using either the `bandwidth` statement or the `bandwidth-percent` statement. You can only configure one these statements at a time. If you do not configure either the `bandwidth` statement or the `bandwidth-percent` statement, the default setting is to not

reserve bandwidth for the detour path (the bandwidth guarantee will be lost if traffic is switched to the detour).

When you configure the `bandwidth` statement, you can specify the specific amount of bandwidth (in bits per second [bps]) you want to reserve for the detour path. For information, see "Configuring Fast Reroute" on page 580.

The `bandwidth-percent` statement allows you to specify the bandwidth of the detour path as a percentage of the bandwidth configured for the protected path. For example, if you configure 100 millions bps of bandwidth for the protected path and configure 20 for the `bandwidth-percent` statement, the detour path will have 20 million bps of bandwidth reserved for its use.

To configure the percent of bandwidth used by the detour path based on the bandwidth of the protected path, include the `bandwidth-percent` statement:

```
bandwidth-percent percentage;
```

You can include this statement at the following hierarchy levels:

- `[edit protocols mpls label-switched-path lsp-name fast-reroute]`
- `[edit logical-systems logical-system-name protocols mpls label-switched-path lsp-name fast-reroute]`

RELATED DOCUMENTATION

Basic MPLS Configuration | 46

7

PART

MPLS Transport Profile

- Operation, Administration, and Maintenance (OAM) for MPLS | **1842**
- MPLS Pseudowires | **1863**
- Class-of-Service (CoS) for MPLS | **1950**
- Generalized MPLS (GMPLS) | **1992**

CHAPTER 13

Operation, Administration, and Maintenance (OAM) for MPLS

IN THIS CHAPTER

- MPLS OAM Configuration | **1842**

MPLS OAM Configuration

IN THIS SECTION

- Configuring the MPLS Transport Profile for OAM | **1842**
- Configuring OAM Ingress Policies for LDP | **1860**
- Tracing MPLS and LSP Packets and Operations | **1861**

Configuring the MPLS Transport Profile for OAM

IN THIS SECTION

- MPLS Transport Profile Overview | **1842**
- Example: Configuring the MPLS Transport Profile for OAM | **1843**

MPLS Transport Profile Overview

RFC 5654, *Requirements of an MPLS Transport Profile*, describes the requirements for the MPLS Transport Profile (MPLS-TP) that extends capabilities for Operation, Administration, and Maintenance

(OAM) when MPLS is used for transport services and transport network operations. These capabilities help in troubleshooting and maintenance of a pseudowire or label-switched path (LSP).

MPLS-TP mechanisms for OAM contain two main components:

- Generic Associated Channel Label (GAL)—A special label that enables an exception mechanism that informs the egress *label-switching router* (LSR) that a packet it receives on an LSP belongs to an associated control channel or the control plane.
- Generic Associated Channel Header (G-Ach)—A special header field that identifies the type of payload contained in the MPLS label-switched paths (LSPs). G-Ach has the same format as a pseudowire associated control channel header.

For more information about MPLS-TP, see RFC 5654, *Requirements of an MPLS Transport Profile*. For specific information about GAL and G-Ach, see RFC 5586, *MPLS Generic Associated Channel*.

The following capabilities are supported in the Junos OS implementation of MPLS-TP:

- MPLS-TP OAM can send and receive packets with GAL and G-Ach, without IP encapsulation.
- Two unidirectional RSVP LSPs between a pair of routers can be associated with each other to create an associated bidirectional LSP for binding a path for the GAL and G-Ach OAM messages. A single Bidirectional Forwarding Detection (BFD) session is established for the associated bidirectional LSP.

Example: Configuring the MPLS Transport Profile for OAM

IN THIS SECTION

- Requirements | **1843**
- Overview | **1844**
- Configuration | **1846**
- Verification | **1857**

This example shows how to configure the MPLS Transport Profile (MPLS-TP) for sending and receiving of OAM GAL and G-Ach messages across a label-switched path (LSP).

Requirements

This example uses the following hardware and software components:

- Six devices that can be a combination of M Series, MX Series, and T Series routers
- Junos OS Release 12.1 or later running on the devices

Overview

IN THIS SECTION

- Topology | **1846**

Junos OS Release 12.1 and later support MPLS Transport Profile (MPLS-TP) Operation, Administration, and Maintenance (OAM) capabilities. MPLS-TP introduces new capabilities for OAM when MPLS is used for transport services and transport network operations. This includes configuring Generic Associated Channel Label (GAL) and Generic Associated Channel Header (G-Ach) for OAM messages.

This example shows how to configure MPLS-TP OAM capability to send and receive GAL and G-Ach OAM messages without IP encapsulation. In addition, it also shows how to associate two unidirectional RSVP label-switched paths (LSPs) between a pair of routers to create an associated bidirectional LSP for binding a path for the GAL and G-Ach OAM messages.

Junos OS Release 12.1 and later support the following MPLS-TP capabilities:

- MPLS-TP OAM capability and the infrastructure required for MPLS applications to send and receive packets with GAL and G-Ach, without IP encapsulation.
- LSP-ping and Bidirectional Forwarding Detection (BFD) applications to send and receive packets using GAL and G-Ach, without IP encapsulation on transport LSPs.
- The association of two unidirectional RSVP LSPs, between a pair of routers, with each other to create an associated bidirectional LSP for binding a path for the GAL and G-Ach OAM messages. The associated bidirectional LSP model is supported only for associating the primary paths. A single BFD session is established for the associated bidirectional LSP.

Junos OS Release 12.1 and later does not support the following MPLS-TP capabilities:

- Point-to-multipoint RSVP LSPs and BGP LSPs
- Loss Measurement and Delay Measurement

You can enable GAL and G-Ach OAM operation using the following configuration statements:

- `mpls-tp-mode`—Include this statement at the `[edit protocols mpls oam]` hierarchy level to enable GAL and G-Ach OAM operation, without IP encapsulation, on all LSPs in the MPLS network.

```
[edit protocols mpls oam]
mpls-tp-mode;
```

Include this statement at the [edit protocols mpls label-switched-path *lsp-name* oam] hierarchy level to enable GAL and G-Ach OAM operation without IP encapsulation on a specific LSP in the network.

```
[edit protocols mpls label-switched-path lsp-name oam]
mpls-tp-mode;
```

NOTE: Starting with Junos OS Release 16.1, MPLS-TP supports two additional channel types for the default LSPING (0x0008) channel type under the mpls-tp-mode statement. These additional channel types provide on-demand connectivity verification (CV) with and without IP/UDP encapsulation.

- On-demand CV (0x0025)—This channel type is a new pseudowire channel type and is used for on-demand CV without IP/UDP encapsulation, where IP addressing is not available or non-IP encapsulation is preferred.
- IPv4 (0x0021)—This channel type uses the IP/UDP encapsulation and provides interoperability support with other vendor devices using IP addressing.

The GACH-TLV is used along with the default LSPING channel type. As per RFC 7026, GACH-TLV is deprecated for 0x0021 and 0x0025 channel types.

To configure a channel type for MPLS-TP, include the lsping-channel-type *channel-type* statement at the [edit protocols mpls label-switched-path *lsp-name* oam mpls-tp-mode] and [edit protocols mpls oam mpls-tp-mode] hierarchy levels.

- associate-lsp *lsp-name* from *from-ip-address*—Include this statement at the [edit protocols mpls label-switched-path *lsp-name*] hierarchy level to configure associated bidirectional LSPs on the two ends of the LSP.

```
[edit protocols mpls label-switched-path lsp-name ]
associate-lsp lsp-name {
    from from-ip-address;
}
```

The from *from-ip-address* configuration for the LSP is optional. If omitted, it is derived from the to address of the ingress LSP configuration.

- transit-lsp-association—Include this statement at the [edit protocols mpls]

hierarchy level to associate two LSPs at a transit router.

```
[edit protocols mpls]
transit-lsp-association transit-association-lsp-group-name {
    lsp-name-1 name-of-associated-lsp-1;
    from-1     address-of-associated-lsp-1;
    lsp-name-2 name-of-associated-lsp-2;
    from-2     address-of-associated-lsp-2;
}
```

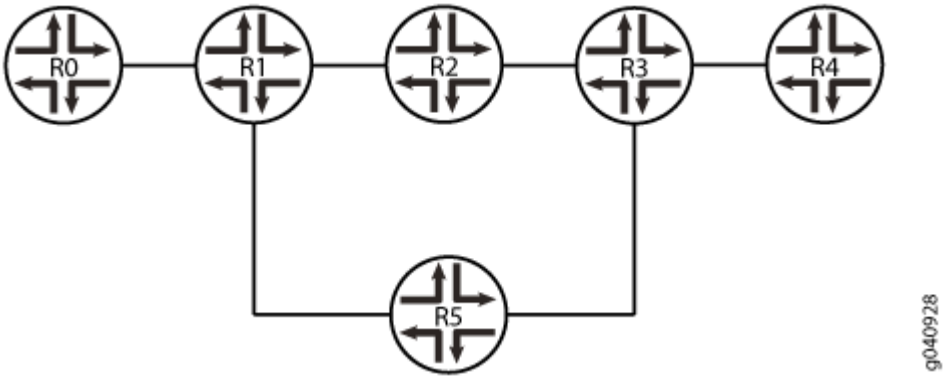
The association of the LSPs in the transit nodes is useful for the return LSP path for TTL-expired LSP ping packets or traceroute.

In this example, R0 is the ingress router and R4 is the egress router. R1, R2, R3, and R5 are transit routers. The associated bidirectional LSP is established between the transit routers for sending and receiving the GAL and G-Ach OAM messages.

Figure 116 on page 1846 shows the topology used in this example.

Topology

Figure 116: MPLS-TP OAM Associated Bidirectional LSPs



Configuration

IN THIS SECTION

- CLI Quick Configuration | **1847**
- Configuring Device R0 | **1851**
- Configuring Device R1 | **1854**

CLI Quick Configuration

NOTE: This example shows the configuration on all devices and shows step-by-step procedures for configuring the ingress router, R0, and transit router R1. Repeat the step-by-step procedure described for the ingress router, R0, on the egress router, R4. Repeat the step-by-step procedure for the transit router, R1, on the other transit routers, R2, R3, and R5. Be sure to modify the appropriate interface names, addresses, and other parameters appropriately.

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the `[edit]` hierarchy level.

Router R0

```
set interfaces ge-4/1/1 unit 0 family inet address 10.10.11.1/30
set interfaces ge-4/1/1 unit 0 family iso
set interfaces ge-4/1/1 unit 0 family inet6
set interfaces ge-4/1/1 unit 0 family mpls
set interfaces ge-5/0/0 unit 0 family inet address 10.10.10.1/30
set interfaces ge-5/0/0 unit 0 family iso
set interfaces ge-5/0/0 unit 0 family inet6
set interfaces ge-5/0/0 unit 0 family mpls
set protocols rsvp interface ge-5/0/0.0
set protocols rsvp interface ge-4/1/1.0
set protocols mpls label-switched-path r0-to-r4 to 10.255.8.86
set protocols mpls label-switched-path r0-to-r4 oam mpls-tp-mode
set protocols mpls label-switched-path r0-to-r4 associate-lsp r4-to-r0 from 10.255.8.86
set protocols mpls interface ge-5/0/0.0
set protocols mpls interface ge-4/1/1.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-5/0/0.0
set protocols ospf area 0.0.0.0 interface ge-4/1/1.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

Router R1

```
set interfaces ge-0/0/5 unit 0 family inet address 10.10.10.2/30
set interfaces ge-0/0/5 unit 0 family iso
set interfaces ge-0/0/5 unit 0 family inet6
set interfaces ge-0/0/5 unit 0 family mpls
```

```
set interfaces ge-0/2/2 unit 0 family inet address 10.10.12.2/30
set interfaces ge-0/2/2 unit 0 family iso
set interfaces ge-0/2/2 unit 0 family inet6
set interfaces ge-0/2/2 unit 0 family mpls
set interfaces ge-1/0/2 unit 0 family inet address 10.10.13.2/30
set interfaces ge-1/0/2 unit 0 family iso
set interfaces ge-1/0/2 unit 0 family inet6
set interfaces ge-1/0/2 unit 0 family mpls
set interfaces ge-2/0/2 unit 0 family inet address 10.10.11.2/30
set interfaces ge-2/0/2 unit 0 family iso
set interfaces ge-2/0/2 unit 0 family inet6
set interfaces ge-2/0/2 unit 0 family mpls
set protocols rsvp interface ge-0/2/2.0
set protocols rsvp interface ge-0/0/5.0
set protocols rsvp interface ge-1/0/2.0
set protocols rsvp interface ge-2/0/2.0
set protocols mpls transit-lsp-association trace1 lsp-name-1 r0-to-r4
set protocols mpls transit-lsp-association trace1 from-1 10.255.8.207
set protocols mpls transit-lsp-association trace1 lsp-name-2 r4-to-r0
set protocols mpls transit-lsp-association trace1 from-2 10.255.8.86
set protocols mpls interface ge-0/0/5.0
set protocols mpls interface ge-2/0/2.0
set protocols mpls interface ge-1/0/2.0
set protocols mpls interface ge-0/2/2.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/0/5.0
set protocols ospf area 0.0.0.0 interface ge-0/2/2.0 metric 100
set protocols ospf area 0.0.0.0 interface ge-1/0/2.0
set protocols ospf area 0.0.0.0 interface ge-2/0/2.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

Router R2

```
set interfaces ge-0/2/3 unit 0 family inet address 10.10.13.1/30
set interfaces ge-0/2/3 unit 0 family iso
set interfaces ge-0/2/3 unit 0 family inet6
set interfaces ge-0/2/3 unit 0 family mpls
set interfaces ge-1/3/2 unit 0 family inet address 10.10.14.1/30
set interfaces ge-1/3/2 unit 0 family iso
set interfaces ge-1/3/2 unit 0 family inet6
set interfaces ge-1/3/2 unit 0 family mpls
set interfaces ge-1/3/4 unit 0 family inet address 10.10.15.1/30
```

```
set interfaces ge-1/3/4 unit 0 family iso
set interfaces ge-1/3/4 unit 0 family inet6
set interfaces ge-1/3/4 unit 0 family mpls
set protocols rsvp interface ge-0/2/3.0
set protocols rsvp interface ge-1/3/2.0
set protocols rsvp interface ge-1/3/4.0
set protocols mpls transit-lsp-association trace1 lsp-name-1 r0-to-r4
set protocols mpls transit-lsp-association trace1 from-1 10.255.8.207
set protocols mpls transit-lsp-association trace1 lsp-name-2 r4-to-r0
set protocols mpls transit-lsp-association trace1 from-2 10.255.8.86
set protocols mpls interface ge-0/2/3.0
set protocols mpls interface ge-1/3/2.0
set protocols mpls interface ge-1/3/4.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/2/3.0
set protocols ospf area 0.0.0.0 interface ge-1/3/2.0
set protocols ospf area 0.0.0.0 interface ge-1/3/4.0 metric 100
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

Router R3

```
set interfaces ge-1/2/1 unit 0 family inet address 10.10.16.2/30
set interfaces ge-1/2/1 unit 0 family iso
set interfaces ge-1/2/1 unit 0 family inet6
set interfaces ge-1/2/1 unit 0 family mpls
set interfaces ge-2/0/7 unit 0 family inet address 10.10.17.2/30
set interfaces ge-2/0/7 unit 0 family iso
set interfaces ge-2/0/7 unit 0 family inet6
set interfaces ge-2/0/7 unit 0 family mpls
set interfaces ge-2/2/0 unit 0 family inet address 10.10.14.2/30
set interfaces ge-2/2/0 unit 0 family iso
set interfaces ge-2/2/0 unit 0 family inet6
set interfaces ge-2/2/0 unit 0 family mpls
set protocols rsvp interface ge-2/2/0.0
set protocols rsvp interface ge-1/2/1.0
set protocols rsvp interface ge-2/0/7.0
set protocols mpls transit-lsp-association trace1 lsp-name-1 r0-to-r4
set protocols mpls transit-lsp-association trace1 from-1 10.255.8.207
set protocols mpls transit-lsp-association trace1 lsp-name-2 r4-to-r0
set protocols mpls transit-lsp-association trace1 from-2 10.255.8.86
set protocols mpls interface ge-2/2/0.0
set protocols mpls interface ge-1/2/1.0
```

```
set protocols mpls interface ge-2/0/7.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-2/2/0.0
set protocols ospf area 0.0.0.0 interface ge-1/2/1.0
set protocols ospf area 0.0.0.0 interface ge-2/0/7.0 metric 100
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

Router R4

```
set interfaces ge-0/0/3 unit 0 family inet address 10.10.16.1/30
set interfaces ge-0/0/3 unit 0 family iso
set interfaces ge-0/0/3 unit 0 family inet6
set interfaces ge-0/0/3 unit 0 family mpls
set protocols rsvp interface ge-0/0/3.0
set protocols mpls label-switched-path r4-to-r0 to 10.255.8.207
set protocols mpls label-switched-path r4-to-r0 oam mpls-tp-mode
set protocols mpls label-switched-path r4-to-r0 associate-lsp r0-to-r4 from 10.255.8.207
set protocols mpls interface ge-0/0/3.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

Router R5

```
set interfaces ge-1/2/0 unit 0 family inet address 10.10.15.2/30
set interfaces ge-1/2/0 unit 0 family iso
set interfaces ge-1/2/0 unit 0 family inet6
set interfaces ge-1/2/0 unit 0 family mpls
set interfaces ge-2/0/0 unit 0 family inet address 10.10.12.1/30
set interfaces ge-2/0/0 unit 0 family iso
set interfaces ge-2/0/0 unit 0 family inet6
set interfaces ge-2/0/0 unit 0 family mpls
set interfaces ge-4/0/7 unit 0 family inet address 10.10.17.1/30
set interfaces ge-4/0/7 unit 0 family iso
set interfaces ge-4/0/7 unit 0 family inet6
set interfaces ge-4/0/7 unit 0 family mpls
set protocols rsvp interface ge-2/0/0.0
set protocols rsvp interface ge-1/2/0.0
set protocols rsvp interface ge-4/0/7.0
set protocols mpls transit-lsp-association trace1 lsp-name-1 r0-to-r4
set protocols mpls transit-lsp-association trace1 from-1 10.255.8.207
```

```
set protocols mpls transit-lsp-association trace1 lsp-name-2 r4-to-r0
set protocols mpls transit-lsp-association trace1 from-2 10.255.8.86
set protocols mpls interface ge-2/0/0.0
set protocols mpls interface ge-1/2/0.0
set protocols mpls interface ge-4/0/7.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-2/0/0.0 metric 100
set protocols ospf area 0.0.0.0 interface ge-1/2/0.0 metric 100
set protocols ospf area 0.0.0.0 interface ge-4/0/7.0 metric 100
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

Configuring Device R0

Step-by-Step Procedure

To configure the ingress router, R0:

1. Configure the interfaces.

```
[edit interfaces]
user@R0# set ge-4/1/1 unit 0 family inet address 10.10.11.1/30
user@R0# set ge-4/1/1 unit 0 family iso
user@R0# set ge-4/1/1 unit 0 family inet6
user@R0# set ge-4/1/1 unit 0 family mpls
user@R0# set ge-5/0/0 unit 0 family inet address 10.10.10.1/30
user@R0# set ge-5/0/0 unit 0 family iso
user@R0# set ge-5/0/0 unit 0 family inet6
user@R0# set ge-5/0/0 unit 0 family mpls
```

2. Configure MPLS on the interfaces.

```
[edit protocols mpls]
user@R0# set interface ge-5/0/0.0
user@R0# set interface ge-4/1/1.0
```

3. Configure an interior gateway protocol, such as OSPF.

```
[edit protocols ospf]
user@R0# set traffic-engineering
```

```
user@R0# set area 0.0.0.0 interface ge-5/0/0.0
user@R0# set area 0.0.0.0 interface ge-4/1/1.0
user@R0# set area 0.0.0.0 interface lo0.0 passive
```

4. Configure a signaling protocol, such as RSVP.

```
[edit protocols rsvp]
user@R0# set interface ge-5/0/0.0
user@R0# set interface ge-4/1/1.0
```

5. Configure the LSP.

```
[edit protocols mpls]
user@R0# set label-switched-path r0-to-r4 to 10.255.8.86
```

6. Enable GAL and G-Ach OAM operation without IP encapsulation on the LSPs.

```
[edit protocols mpls]
user@R0# set label-switched-path r0-to-r4 oam mpls-tp-mode
```

7. Configure associated bidirectional LSPs on the two ends of the LSP.

```
[edit protocols mpls]
user@R0# set label-switched-path r0-to-r4 associate-lsp to-r0 from 10.255.8.86
```

8. After you are done configuring the device, commit the configuration.

```
[edit]
user@R0#  commit
```

Results

Confirm your configuration by issuing the `show interfaces` and `show protocols` commands.

```
user@R0# show interfaces
ge-4/1/1 {
```

```
    unit 0 {
        family inet {
            address 10.10.11.1/30;
        }
        family iso;
        family inet6;
        family mpls;
    }
}
ge-5/0/0 {
    unit 0 {
        family inet {
            address 10.10.10.1/30;
        }
        family iso;
        family inet6;
        family mpls;
    }
}
```

```
user@R0# show protocols
rsvp {
    interface ge-5/0/0.0;
    interface ge-4/1/1.0;
}
mpls {
    label-switched-path r0-to-r4 {
        to 10.255.8.86;
        oam mpls-tp-mode;
        associate-lsp r4-to-r0 {
            from 10.255.8.86;
        }
    }
    interface ge-4/1/1.0;
    interface ge-5/0/0.0;
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface ge-5/0/0.0;
        interface ge-4/1/1.0;
```

```
        interface lo0.0 {
            passive;
        }
    }
}
```

Configuring Device R1

Step-by-Step Procedure

To configure the transit router, R1:

1. Configure the interfaces.

```
[edit interfaces]
user@R1# set ge-0/0/5 unit 0 family inet address 10.10.10.2/30
user@R1# set ge-0/0/5 unit 0 family iso
user@R1# set ge-0/0/5 unit 0 family inet6
user@R1# set ge-0/0/5 unit 0 family mpls
user@R1# set ge-0/2/2 unit 0 family inet address 10.10.12.2/30
user@R1# set ge-0/2/2 unit 0 family iso
user@R1# set ge-0/2/2 unit 0 family inet6
user@R1# set ge-0/2/2 unit 0 family mpls
user@R1# set ge-2/0/2 unit 0 family inet address 10.10.11.2/30
user@R1# set ge-2/0/2 unit 0 family iso
user@R1# set ge-2/0/2 unit 0 family inet6
user@R1# set ge-2/0/2 unit 0 family mpls
user@R1# set ge-1/0/2 unit 0 family inet address 10.10.13.2/30
user@R1# set ge-1/0/2 unit 0 family iso
user@R1# set ge-1/0/2 unit 0 family inet6
user@R1# set ge-1/0/2 unit 0 family mpls
```

2. Configure MPLS on the interfaces.

```
[edit protocols mpls]
user@R1# set interface ge-0/0/5.0
user@R1# set interface ge-2/0/2.0
user@R1# set interface ge-1/0/2.0
user@R1# set interface ge-0/2/2.0
```

3. Configure an interior gateway protocol, such as OSPF.

```
[edit protocols ospf]
user@R1# set traffic-engineering
user@R1# set area 0.0.0.0 interface ge-0/0/5.0
user@R1# set area 0.0.0.0 interface ge-2/0/2.0
user@R1# set area 0.0.0.0 interface ge-1/0/2.0
user@R1# set area 0.0.0.0 interface ge-0/2/2.0 metric 100
user@R1# set area 0.0.0.0 interface lo0.0 passive
```

4. Configure a signaling protocol, such as RSVP.

```
[edit protocols rsvp]
user@R1# set interface ge-0/0/5.0
user@R1# set interface ge-2/0/2.0
user@R1# set interface ge-1/0/2.0
user@R1# set interface ge-0/2/2.0
```

5. Configure the association of the two LSPs on the transit router.

```
[edit protocols mpls]
user@R1# set transit-lsp-association trace1 lsp-name-1 r0-to-r4
user@R1# set transit-lsp-association trace1 from-1 10.255.8.207
user@R1# set transit-lsp-association trace1 lsp-name-2 r4-to-r0
user@R1# set transit-lsp-association trace1 from-2 10.255.8.86
```

6. If you are done configuring the device, commit the configuration.

```
[edit]
user@R1# commit
```

Results

Confirm your configuration by issuing the `show interfaces` and `show protocols` commands.

```
user@R1# show interfaces
ge-0/0/5 {
    unit 0 {
```

```
        family inet {
            address 10.10.10.2/30;
        }
        family iso;
        family inet6;
        family mpls;
    }
}
ge-0/2/2 {
    unit 0 {
        family inet {
            address 10.10.12.2/30;
        }
        family iso;
        family inet6;
        family mpls;
    }
}
ge-2/0/2 {
    unit 0 {
        family inet {
            address 10.10.11.2/30;
        }
        family iso;
        family inet6;
        family mpls;
    }
}
ge-1/0/2 {
    unit 0 {
        family inet {
            address 10.10.13.2/30;
        }
        family iso;
        family inet6;
        family mpls;
    }
}
```

```
user@R1# show protocols
rsvp {
```

```
    interface ge-0/0/5.0;
    interface ge-2/0/2.0;
    interface ge-1/0/2.0;
    interface ge-0/2/2.0;
}
mpls {
    transit-lsp-association trace1 {
        lsp-name-1 r0-to-r4;
        from-1 10.255.8.207;
        lsp-name-2 r4-to-r0;
        from-2 10.255.8.86;
    }
    interface ge-0/0/5.0;
    interface ge-2/0/2.0;
    interface ge-1/0/2.0;
    interface ge-0/2/2.0;
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface ge-0/0/5.0;
        interface ge-1/0/2.0;
        interface ge-2/0/2.0;
        interface ge-0/2/2.0 {
            metric 100;
        }
        interface lo0.0 {
            passive;
        }
    }
}
```

Verification

IN THIS SECTION

- Verifying Associated Bidirectional LSPs | **1858**

Confirm that the configuration is working properly.

Verifying Associated Bidirectional LSPs

Purpose

Verify that the associated bidirectional LSP configuration is working properly.

Action

```
user@host> show mpls lsp
Ingress LSP: 1 sessions
To              From            State Rt P   ActivePath   LSPname
10.10.11.1  10.255.8.86               Up     0 *               r0-to-r4 Assoc-Bidir
Total 1 displayed, Up 1, Down 0

Egress LSP: 1 sessions
To         From           State Rt Style Labelin Labelout LSPname
10.10.16.1  10.255.8.207  Up    0  1 FF       3           r4-to-r0 Assoc-Bidir
Total 2 displayed, Up 2, Down 0

Transit LSP: 1 sessions
To          From           State  Rt Style Labelin Labelout LSPname
10.10.10.2  10.255.8.168  Up      1  1 FF  301264        3 r0-to-r4 Assoc-Bidir
Total 3 displayed, Up 3, Down 0
```

```
user@host> show mpls lsp detail
Ingress LSP: 1 sessions

10.10.11.1
  From: 10.255.8.86, State: Up, ActiveRoute: 0, LSPname: r0-to-r4
  Associated Bidirectional
  Associated LSP: r0-to-r4, 10.255.8.86
  ActivePath:  (primary)
  LSPtype: Static Configured
  LoadBalance: Random
  Encoding type: Packet, Switching type: PSC-1, GPID: Unknown
 *Primary                   State: Up

Egress LSP: 1 sessions
```

```
10.255.102.29
  From: 10.255.102.172, LSPstate: Up, ActiveRoute: 0
    LSPname: r4-to-r0, LSPpath: Primary
    Associated Bidirectional
    Associated LSP: 10.10.16.1, to-r0>
    Suggested label received: -, Suggested label sent: -
    Recovery label received: -, Recovery label sent: -
    Resv style: 1 FF, Label in: 3, Label out: -
    Time left:  144, Since: Fri Jun 17 21:41:05 2011
    Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
    Port number: sender 6 receiver 14468 protocol 0
    PATH rcvfrom: 10.10.13.1 (ge-2/0/0.0) 84 pkts
    Adspec: received MTU 1500
    PATH sentto: localclient
    RESV rcvfrom: localclient
    Record route: 10.10.14.2 10.10.13.1 <self>

Transit LSP: 1 sessions

10.255.102.30
  From: 10.255.102.172, LSPstate: Up, ActiveRoute: 1
    LSPname: to_airstream, LSPpath: Primary
    Associated Bidirectional
    Associated LSP: r0-to-r4, 10.255.8.168
    Suggested label received: -, Suggested label
    Recovery label received: -, Recovery label sent: 3
    Resv style: 1 FF, Label in: 301264, Label out: 3
    Time left:  132, Since: Fri Jun 17 21:40:56 2011
    Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
    Port number: sender 28 receiver 14465 protocol 0
    PATH rcvfrom: 10.10.13.1 (ge-2/0/0.0) 84 pkts
    Adspec: received MTU 1500 sent MTU 1500
    PATH sentto: 10.10.10.1 (ge-3/0/0.0) 84 pkts
    RESV rcvfrom: 10.10.10.1 (ge-3/0/0.0) 84 pkts
    Explct route: 10.10.10.1
    Record route: 10.10.16.1 10.10.15.2 10.10.13.1 <self> 10.10.10.1
```

```
user@host> show mpls lsp bidirectional
Ingress LSP: 1 session
To              From            State Rt P     ActivePath       LSPname
10.255.8.86     10.255.8.207    Up     0 *                      r0-to-r4
```

```
 Assoc-Bidir
Total 1 displayed, Up 1, Down 0
Aug 28 06:56:26 [TRACE] [R0 coleman re0]
Egress LSP: 1 session
To              From            State   Rt Style Labelin Labelout LSPname
10.255.8.207    10.255.8.86      Up       0  1 FF       3        - to-r0
 Assoc-Bidir
Total 1 displayed, Up 1, Down 0
Aug 28 06:56:26 [TRACE] [R0 coleman re0]
Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Meaning

The output of the `show mpls lsp`, `show mpls detail`, and `show mpls bidirectional` commands displays the details of the associated bidirectional LSPs and the LSP association information.

Configuring OAM Ingress Policies for LDP

Using the `ingress-policy` statement, you can configure an Operation, Administration, and Management (OAM) policy to choose which forwarding equivalence classes (FECs) need to have OAM enabled. If the FEC passes through the policy or if the FEC is explicitly configured, OAM is enabled for a FEC. For FECs chosen using a policy, the BFD parameters configured under `[edit protocols ldp oam bfd-liveness-detection]` are applied.

You configure the OAM ingress policy at the `[edit policy-options]` hierarchy level. To configure an OAM ingress policy, include the `ingress-policy` statement:

```
ingress-policy ingress-policy-name;
```

You can configure this statement at the following hierarchy levels:

- `[edit protocols ldp oam]`
- `[edit logical-systems logical-system-name protocols ldp oam]`

NOTE: ACX Series routers do not support `[edit logical-systems]` hierarchy level.

Tracing MPLS and LSP Packets and Operations

To trace MPLS and LSP packets and operations, include the `traceoptions` statement:

```
traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag;
}
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

You can specify the following MPLS-specific flags in the MPLS `traceoptions` statement:

- `all`—Trace all operations.
- `connection`—Trace all circuit cross-connect (CCC) activity.
- `connection-detail`—Trace detailed CCC activity.
- `cspf`—Trace CSPF computations.
- `cspf-link`—Trace links visited during CSPF computations.
- `cspf-node`—Trace nodes visited during CSPF computations.
- `error`—Trace MPLS error conditions.
- `graceful-restart`—Trace MPLS graceful restart events.
- `lsping`—Trace LSP ping packets and return codes.
- `nsr-synchronization`—Trace nonstop routing (NSR) synchronization events.
- `nsr-synchronization-detail`—Trace NSR synchronization events in detail.
- `state`—Trace all LSP state transitions.
- `static`—Trace static label-switched path.

When you configure trace options to track an MPLS LSP using the `cspf` option, the CSPF log displays information about the MPLS LSP using the term “generalized MPLS” (GMPLS). For example, a message in the CSPF log might state that the “link passes GMPLS constraints”. Generalized MPLS (GMPLS) is a superset of MPLS, so this message is normal and does not affect proper MPLS LSP operation.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
16.1	Starting with Junos OS Release 16.1, MPLS-TP supports two additional channel types for the default LSPING (0x0008) channel type under the mpls-tp-mode statement.

RELATED DOCUMENTATION

Basic MPLS Configuration | 46

CHAPTER 14

MPLS Pseudowires

IN THIS CHAPTER

- MPLS Pseudowires Configuration | **1863**
- Pseudowire Headend Termination (PWHT) Configuration | **1934**
- Pseudowire over Aggregated Ethernet Overview | **1946**

MPLS Pseudowires Configuration

IN THIS SECTION

- Ethernet Pseudowire Overview | **1863**
- Example: Ethernet Pseudowire Base Configuration | **1864**
- Pseudowire Overview for ACX Series Universal Metro Routers | **1868**
- Understanding Multisegment Pseudowire for FEC 129 | **1869**
- Example: Configuring a Multisegment Pseudowire | **1875**
- MPLS Stitching For Virtual Machine Connection | **1924**
- TDM Pseudowires Overview | **1926**
- Example: TDM Pseudowire Base Configuration | **1926**
- Configuring Load Balancing for Ethernet Pseudowires | **1931**
- Configuring Load Balancing Based on MAC Addresses | **1933**

Ethernet Pseudowire Overview

Starting in Junos OS Release 14.1X53 and Junos OS Release 16.1, an Ethernet pseudowire is used to carry Ethernet or 802.3 Protocol Data Units (PDUs) over an MPLS network enabling service providers to offer emulated Ethernet services over existing MPLS networks. Ethernet or 802.3 PDUs are

encapsulated within the pseudowire to provide a point-to-point Ethernet service. For the point-to-point Ethernet service, the following fault management features are supported:

- The IEEE 802.3ah standard for Operation, Administration, and Management (OAM). You can configure IEEE 802.3ah OAM link-fault management on Ethernet point-to-point direct links or links across Ethernet repeaters.

  Ethernet OAM link-fault management can be used for physical link-level fault detection and management. It uses a new, optional sublayer in the data link layer of the OSI model. Ethernet OAM can be implemented on any full-duplex point-to-point or emulated point-to-point Ethernet link. A system-wide implementation is not required; OAM can be deployed on particular interfaces of a router. Transmitted Ethernet OAM messages or OAM PDUs are of standard length, untagged Ethernet frames within the normal frame length limits in the range 64–1518 bytes.

- Ethernet connectivity fault management (CFM) to monitor the physical link between two routers.

  - Connection protection using the continuity check protocol for fault monitoring . The continuity check protocol is a neighbor discovery and health check protocol that discovers and maintains adjacencies at the VLAN or link level.

  - Path protection using the linktrace protocol for path discovery and fault verification . Similar to IP traceroute, the linktrace protocol maps the path taken to a destination MAC address through one or more bridged networks between the source and destination.

Example: Ethernet Pseudowire Base Configuration

IN THIS SECTION

- Requirements | **1864**
- Overview of an Ethernet Pseudowire Base Configuration | **1865**
- Configuring an Ethernet Pseudowire | **1865**

Requirements

The following is a list of the hardware and software requirements for this configuration.

- One ACX Series router
- Junos OS Release 12.2 or later

Overview of an Ethernet Pseudowire Base Configuration

The configuration shown here is the base configuration of an Ethernet pseudowire with Ethernet cross-connect for physical interface encapsulation on an ACX Series router. This configuration is for one provider edge router. To complete the configuration of an Ethernet pseudowire, you need to repeat this configuration on an other provider edge router in the Multiprotocol Label Switched (MPLS) network.

Configuring an Ethernet Pseudowire

IN THIS SECTION

- Procedure | **1865**
- Results | **1867**

Procedure

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them in a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level:

```
set interfaces ge-0/1/1 encapsulation ethernet-ccc
set interfaces ge-0/1/1 unit 0
set interfaces ge-0/2/0 unit 0 family inet address 20.1.1.2/24
set interfaces ge-0/2/0 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 70.1.1.1/32
set protocols rsvp interface ge-0/2/0.0
set protocols mpls no-cspf
set protocols mpls label-switched-path PE1-to-PE2 to 40.1.1.1
set protocols mpls interface ge-0/2/0.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/2/0.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ldp interface ge-0/2/0.0
set protocols ldp interface lo0.0
set protocols l2circuit neighbor 40.1.1.1 interface ge-0/1/1.0 virtual-circuit-id 1
```

NOTE: To configure an Ethernet pseudowire with 802.1Q tagging for cross-connect logical interface encapsulation, include the `vlan-ccc` statement at the [`edit interfaces ge-0/1/1 unit 0 encapsulation`] hierarchy level instead of the `ethernet-ccc` statement shown in this example.

Step-by-Step Procedure

1. Create two Gigabit Ethernet interfaces, set the encapsulation mode on one interface and MPLS on the other interface. Create the loopback (`lo0`) interface:

```
[edit]
user@host# edit interfaces
[edit interfaces]
user@host# set ge-0/1/1 encapsulation ethernet-ccc
user@host# set ge-0/1/1 unit 0
user@host# set ge-0/2/0 unit 0 family inet address 20.1.1.2/24
user@host# set ge-0/2/0 unit 0 family mpls
user@host# set lo0 unit 0 family inet address 70.1.1.1/32
```

2. Enable the MPLS and RSVP protocols on the interface configured with MPLS—`ge-0/2/0.0`:

```
[edit]
user@host# edit protocols
[edit protocols]
user@host# set rsvp interface ge-0/2/0.0
user@host# set mpls interface ge-0/2/0.0
```

3. Configure LDP. If you configure RSVP for a pseudowire, you must also configure LDP:

```
[edit protocols]
user@host# set protocols ldp interface ge-0/2/0.0
user@host# set protocols ldp interface lo0.0
```

4. Configure a point-to-point label-switched path (LSP) and disable constrained-path LSP computation:

```
[edit protocols]
user@host# set mpls label-switched-path PE1-to-PE2 to 40.1.1.1
user@host# set mpls no-cspf
```

5. Configure OSPF and enable traffic engineering on the MPLS interface—`ge-0/2/0.0`, and on the loopback (`lo0`) interface:

```
[edit protocols]
user@host# set ospf traffic-engineering
user@host# set ospf area 0.0.0.0 interface ge-0/2/0.0
user@host# set ospf area 0.0.0.0 interface lo0.0 passive
```

6. Uniquely identify a Layer 2 circuit for the Ethernet pseudowire:

```
[edit protocols]
user@host#  set l2circuit neighbor 40.1.1.1 interface ge-0/1/1.0 virtual-circuit-id 1
```

Results

```
[edit]
user@host# show
interfaces {
    ge-0/1/1 {
        encapsulation ethernet-ccc;
        unit 0;
    }
    ge-0/2/0 {
        unit 0 {
            family inet {
                address 20.1.1.2/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
```

```
            family inet {
                address 70.1.1.1/32;
            }
        }
    }
}
protocols {
    rsvp {
        interface ge-0/2/0.0;
    }
    mpls {
        no-cspf;
        label-switched-path PE1-to-PE2 {
            to 40.1.1.1;
        }
        interface ge-0/2/0.0;
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface ge-0/2/0.0;
            interface lo0.0 {
                passive;
            }
        }
    }
    ldp {
        interface ge-0/2/0.0;
        interface lo0.0;
    }
    l2circuit {
        neighbor 40.1.1.1 {
            interface ge-0/1/1.0 {
                virtual-circuit-id 1;
            }
        }
    }
}
```

Pseudowire Overview for ACX Series Universal Metro Routers

A pseudowire is a Layer 2 circuit or service, which emulates the essential attributes of a telecommunications service— such as a T1 line, over an MPLS packet-switched network. The

pseudowire is intended to provide only the minimum necessary functionality to emulate the wire with the required degree of faithfulness for the given service definition. On the ACX Series routers, Ethernet, Asynchronous Transfer Mode (ATM), and time-division multiplexing (TDM) pseudowires are supported. The following pseudowire features are supported:

- Pseudowire transport service carrying Layer 1 and Layer 2 information over an IP and MPLS network infrastructure. Only similar end points are supported on the ACX Series—for example, T1 to T1, ATM to ATM, and Ethernet to Ethernet.
- Redundant pseudowires backup connections between PE routers and CE devices, maintaining Layer 2 circuits and services after certain types of failures. Pseudowire redundancy improves the reliability of certain types of networks (metro for example) where a single point of failure could interrupt service for multiple customers. The following pseudowire redundancy features are supported:
  - Maintenance of Layer 2 circuit services after certain types of failures with a standby pseudowire, which backs up the connection between PE routers and CE devices.
  - In case of failure, a protect interface, which backs up the primary interface. Network traffic uses the primary interface only so long as the primary interface functions. If the primary interface fails, traffic is switched to the protect interface.
  - Hot and cold standby enabling swift cut over to the backup or standby pseudowire.
- Ethernet connectivity fault management (CFM), which can be used to monitor the physical link between two routers. The following major features of CFM for Ethernet pseudowires only are supported:
  - Connection protection using the continuity check protocol for fault monitoring. The continuity check protocol is a neighbor discovery and health check protocol that discovers and maintains adjacencies at the VLAN or link level.
  - Path protection using the linktrace protocol for path discovery and fault verification. Similar to IP traceroute, the linktrace protocol maps the path taken to a destination MAC address through one or more bridged networks between the source and destination.

Understanding Multisegment Pseudowire for FEC 129

IN THIS SECTION

- Understanding Multisegment Pseudowire | **1870**
- Using FEC 129 for Multisegment Pseudowire | **1871**
- Establishing a Multisegment Pseudowire Overview | **1872**
- Pseudowire Status Support for Multisegment Pseudowire | **1872**

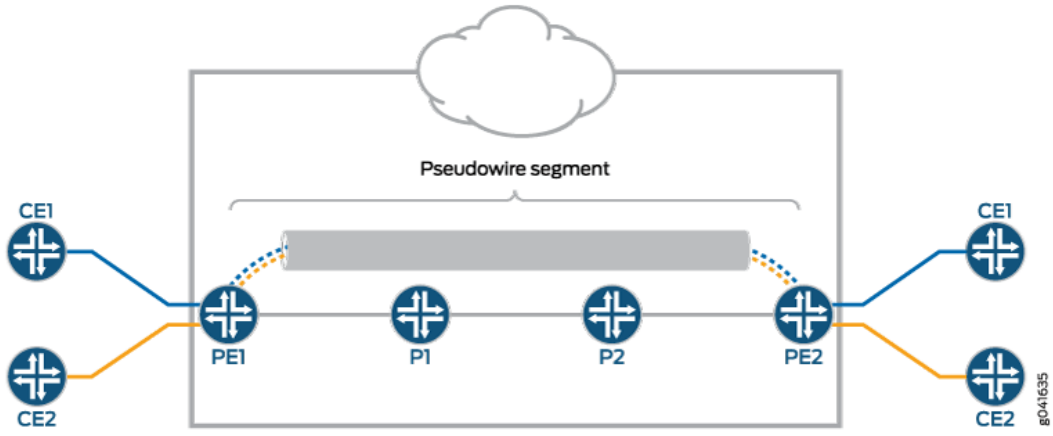
Pseudowire TLV Support for MS-PW | 1873
Supported and Unsupported Features | 1874

Understanding Multisegment Pseudowire

A pseudowire is a Layer 2 circuit or service that emulates the essential attributes of a telecommunications service, such as a T1 line, over an MPLS packet-switched network (PSN). The pseudowire is intended to provide only the minimum necessary functionality to emulate the wire with the required resiliency requirements for the given service definition.

When a pseudowire originates and terminates on the edge of the same PSN, the pseudowire label is unchanged between the originating and terminating provider edge (T-PE) devices. This is called a single-segment pseudowire (SS-PW). Figure 117 on page 1870 illustrates an SS-PW established between two PE routers. The pseudowires between the PE1 and PE2 routers are located within the same autonomous system (AS).

Figure 117: L2VPN Pseudowire



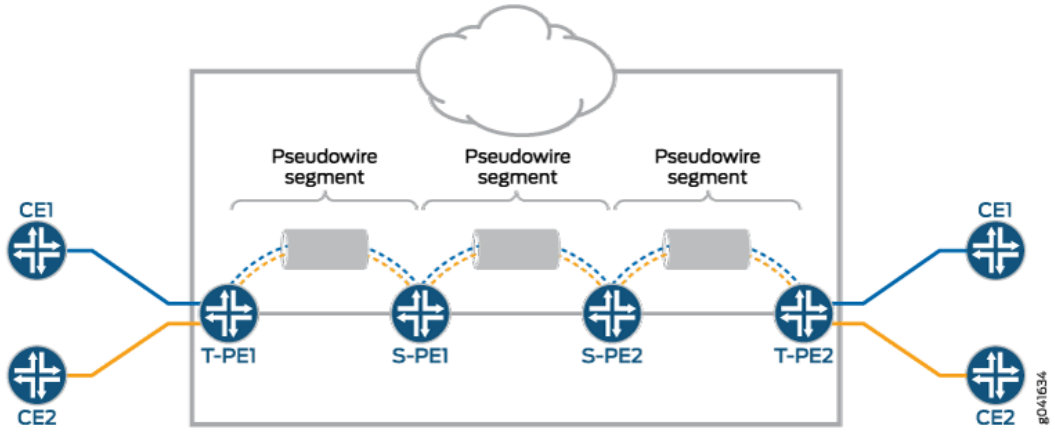
In cases where it is impossible to establish a single pseudowire from a local to a remote PE, either because it is unfeasible or undesirable to establish a single control plane between the two PEs, a multisegment pseudowire (MS-PW) is used.

An MS-PW is a set of two or more contiguous SS-PWs that are made to function as a single point-to-point pseudowire. It is also known as switched pseudowire. MS-PWs can go across different regions or network domains. A region can be considered as an interior gateway protocol (IGP) area or a BGP autonomous system that belongs to the same or different administrative domain. An MS-PW spans

multiple cores or ASs of the same or different carrier networks. A Layer 2 VPN MS-PW can include up to 254 pseudowire segments.

Figure 118 on page 1871 illustrates a set of two or more pseudowire segments that function as a single pseudowire. The end routers are called terminating PE (T-PE) routers, and the switching routers are called switching PE (S-PE) routers. The S-PE router terminates the tunnels of the preceding and succeeding pseudowire segments in an MS-PW. The S-PE router can switch the control and data planes of the preceding and succeeding pseudowire segments of the MS-PW. An MS-PW is declared to be up when all the single-segment pseudowires are up.

Figure 118: Multisegment Pseudowire



Using FEC 129 for Multisegment Pseudowire

Currently, there are two types of attachment circuit identifiers (AIIs) defined under FEC 129:

- Type 1 AII
- Type 2 AII

The support of an MS-PW for FEC 129 uses type 2 AII. A type 2 AII is globally unique by definition of RFC 5003.

Single-segment pseudowires (SS-PWs) using FEC 129 on an MPLS PSN can use both type 1 and type 2 AII. For an MS-PW using FEC 129, a pseudowire itself is identified as a pair of endpoints. This requires that the pseudowire endpoints be uniquely identified.

In the case of a dynamically placed MS-PW, there is a requirement for the identifiers of attachment circuits to be globally unique, for the purposes of reachability and manageability of the pseudowire. Thus, individual globally unique addresses are allocated to all the attachment circuits and S-PEs that make up an MS-PW.

Type 2 AII is composed of three fields:

- Global_ID—Global identification, which is usually the AS number.
- Prefix—IPv4 address, which is usually the router ID.
- AC_ID—Local attachment circuit, which is a user-configurable value.

Since type 2 AII already contains the T-PE's IP address and it is globally unique, from the FEC 129 pseudowire signaling point of view, the combination (AGI, SAII, TAII) uniquely identifies an MS-PW across all interconnected pseudowire domains.

Establishing a Multisegment Pseudowire Overview

An MS-PW is established by dynamically and automatically selecting the predefined S-PEs and placing the MS-PW between two T-PE devices.

When S-PEs are dynamically selected, each S-PE is automatically discovered and selected using the BGP autodiscovery feature, without the requirement of provisioning the FEC 129 pseudowire-related information on all the S-PEs. BGP is used to propagate pseudowire address information throughout the PSN.

Since there is no manual provisioning of FEC 129 pseudowire information on the S-PEs, the Attachment Group Identifier (AGI) and Attachment Individual Identifier (AII) are reused automatically, and choosing the same set of S-PEs for the pseudowire in both the forwarding and reverse direction is achieved through the active and passive role of each T-PE device.

- Active—The T-PE initiates an LDP label mapping message.
- Passive—The T-PE does not initiate an LDP label mapping message until it receives a label mapping message initiated by the active T-PE. The passive T-PE sends its label mapping message to the same S-PE from where it received the label mapping message originated from its active T-PE. This ensures that the same set of S-PEs are used in the reverse direction.

Pseudowire Status Support for Multisegment Pseudowire

Pseudowire Status Behavior on T-PE

The following pseudowire status messages are relevant on the T-PE:

- 0x00000010—Local PSN-facing pseudowire (egress) transmit fault.
- 0x00000001—Generic nonforwarding fault code. This is set as the local fault code. The local fault code is set at the local T-PE, and LDP sends a pseudowire status TLV message with the same fault code to the remote T-PE.

- Fault codes are bit-wise OR'ed and stored as remote pseudowire status codes.

Pseudowire Status Behavior on S-PE

The S-PE initiates the pseudowire status messages that indicate the pseudowire faults. The SP-PE in the pseudowire notification message hints where the fault was originated.

- When a local fault is detected by the S-PE, a pseudowire status message is sent in both directions along the pseudowire. Since there are no attachment circuits on an S-PE, only the following status messages are relevant:
  - 0x00000008—Local PSN-facing pseudowire (ingress) receive fault.
  - 0x00000010—Local PSN-facing pseudowire (egress) transmit fault.
- To indicate which SS-PW is at fault, an LDP SP-PE TLV is attached with the pseudowire status code in the LDP notification message. The pseudowire status is passed along from one pseudowire to another unchanged by the control plane switching function.
- If an S-PE initiates a pseudowire status notification message with one particular pseudowire status bit, then for the pseudowire status code an S-PE receives, the same bit is processed locally and not forwarded until the S-PE's original status error is cleared.
- An S-PE keeps only two pseudowire status codes for each SS-PW it is involved in – local pseudowire status code and remote pseudowire status code. The value of the remote pseudowire status code is the result of logic or operation of the pseudowire status codes in the chain of SS-PWs preceding this segment. This status code is incrementally updated by each S-PE upon receipt and communicated to the next S-PE. The local pseudowire status is generated locally based on its local pseudowire status.
- Only transmit fault is detected at the SP-PE. When there is no MPLS LSP to reach the next segment, a local transmit fault is detected. The transmit fault is sent to the next downstream segment, and the receive fault is sent to the upstream segment.
- Remote failures received on an S-PE are just passed along the MS-PW unchanged. Local failures are sent to both segments of the pseudowire that the S-PE is involved in.

Pseudowire TLV Support for MS-PW

MS-PW provides the following support for the LDP SP-PE TLV [RFC 6073]:

- The LDP SP-PE TLVs for an MS-PW include:
  - Local IP address
  - Remote IP address

- An SP-PE adds the LDP SP-PE TLV to the label mapping message. Each SP-PE appends the local LDP SP-PE TLV to the SP-PE list it received from the other segment.
- The pseudowire status notification message includes the LDP SP-PE TLV when the notification is generated at the SP-PE.

Supported and Unsupported Features

Junos OS supports the following features with MS-PW:

- MPLS PSN for each SS-PW that builds up the MS-PW.
- The same pseudowire encapsulation for each SS-PW in an MS-PW – Ethernet or VLAN-CCC.
- The generalized PWid FEC with T-LDP as an end-to-end pseudowire signaling protocol to set up each SS-PW.
- MP-BGP to autodiscover the two endpoint PEs for each SS-PW associated with the MS-PW.
- Standard MPLS operation to stitch two side-by-side SS-PWs to form an MS-PW.
- Automatic discovery of S-PE so that the MS-PW can be dynamically placed.
- Minimum provisioning of S-PE.
- Operation, administration, and maintenance (OAM) mechanisms, including end-to-end MPLS ping or end-to-any-S-PE MPLS ping, MPLS path trace, end-to-end VCCV, and Bidirectional Forwarding Detection (BFD).
- Pseudowire swithing point (SP) PE TLV for the MS-PW.
- Composite next hop on MS-PW.
- Pseudowire status TLV for MS-PW.

Junos OS does not support the following MS-PW functionality:

- Mix of LDP FEC 128 and LDP FEC 129.
- Static pseudowire where each label is provisioned staticially.
- Graceful Routing Engine switchover.
- Nonstop active routing.
- Multihoming.
- Partial connectivity verification (originating from an S-PE) in OAM.

Example: Configuring a Multisegment Pseudowire

IN THIS SECTION

- Requirements | **1875**
- Overview | **1876**
- Configuration | **1882**
- Verification | **1910**
- Troubleshooting | **1921**

This example shows how to configure a dynamic multisegment pseudowire (MS-PW), where the stitching provider edge (S-PE) devices are automatically and dynamically discovered by BGP, and pseudowires are signaled by LDP using FEC 129. This arrangement requires minimum provisioning on the S-PEs, thereby reducing the configuration burden that is associated with statically configured Layer 2 circuits while still using LDP as the underlying signaling protocol.

Requirements

This example uses the following hardware and software components:

- Six routers that can be a combination of M Series Multiservice Edge Routers, MX Series 5G Universal Routing Platforms, T Series Core Routers, or PTX Series Packet Transport Routers.
  - Two remote PE devices configured as terminating PEs (T-PEs).
  - Two S-PEs configured as:
    - Route reflectors, in the case of interarea configuration.
    - AS boundary routers or route reflectors, in the case of inter-AS configuration.
- Junos OS Release 13.3 or later running on all the devices.

Before you begin:

1. Configure the device interfaces.
2. Configure OSPF or any other IGP protocol.
3. Configure BGP.
4. Configure LDP.

5. Configure MPLS.

Overview

Starting with Junos OS Release 13.3, you can configure an MS-PW using FEC 129 with LDP signaling and BGP autodiscovery in an MPLS packet-switched network (PSN). The MS-PW feature also provides operation, administration, and management (OAM) capabilities, such as ping, traceroute, and BFD, from the T-PE devices.

To enable autodiscovery of S-PEs in an MS-PW, include the `auto-discovery-mspw` statement at the `[edit protocols bgp group group-name family l2vpn]` hierarchy level.

```
family l2vpn {
    auto-discovery-mspw;
}
```

The automatic selection of S-PE and dynamic setting up of an MS-PW rely heavily on BGP. BGP network layer reachability information (NLRI) constructed for the FEC 129 pseudowire to autodiscover the S-PE is called an MS-PW NLRI [draft-ietf-pwe3-dynamic-ms-pw-15.txt]. The MS-PW NLRI is essentially a prefix consisting of a route distinguisher (RD) and FEC 129 source attachment identifier (SAII). It is referred to as a BGP autodiscovery (BGP-AD) route and is encoded as `RD:SAII`.

Only T-PEs that are provisioned with type 2 AIIs initiate their own MS-PW NLRI respectively. Since a type 2 AII is globally unique, an MS-PW NLRI is used to identify a PE device to which the type 2 AII is provisioned. The difference between a type 1 AII and a type 2 AII requires that a new address family indicator (AFI) and subsequent address family identifier (SAFI) be defined in BGP to support an MS-PW. The proposed AFI and SAFI value pair used to identify the MS-PW NLRI is 25 and 6, respectively (pending IANA allocation).

The AFI and SAFI values support autodiscovery of S-PEs and should be configured on both T-PEs that originate the routes, and the S-PEs that participate in the signaling.

Figure 119 on page 1877 illustrates an inter-area MS-PW setup between two remote PE routers—T-PE1 and T-PE2. The Provider (P) routers are P1 and P2, and the S-PE routers are S-PE1 and S-PE2. The MS-PW is established between T-PE1 and T-PE2, and all the devices belong to the same AS—AS 100. Since S-PE1 and S-PE2 belong to the same AS, they act as route reflectors and are also known as RR 1 and RR 2, respectively.

Figure 120 on page 1877 illustrates an inter-AS MS-PW setup. The MS-PW is established between T-PE1 and T-PE2, where T-PE1, P1, and S-PE1 belong to AS 1, and S-PE2, P2, and T-PE2 belong to AS 2. Since S-PE1 and S-PE2 belong to different ASs, they are configured as ASBR routers and are also known as ASBR 1 and ASBR 2, respectively.

Figure 119: Interarea Multisegment Pseudowire

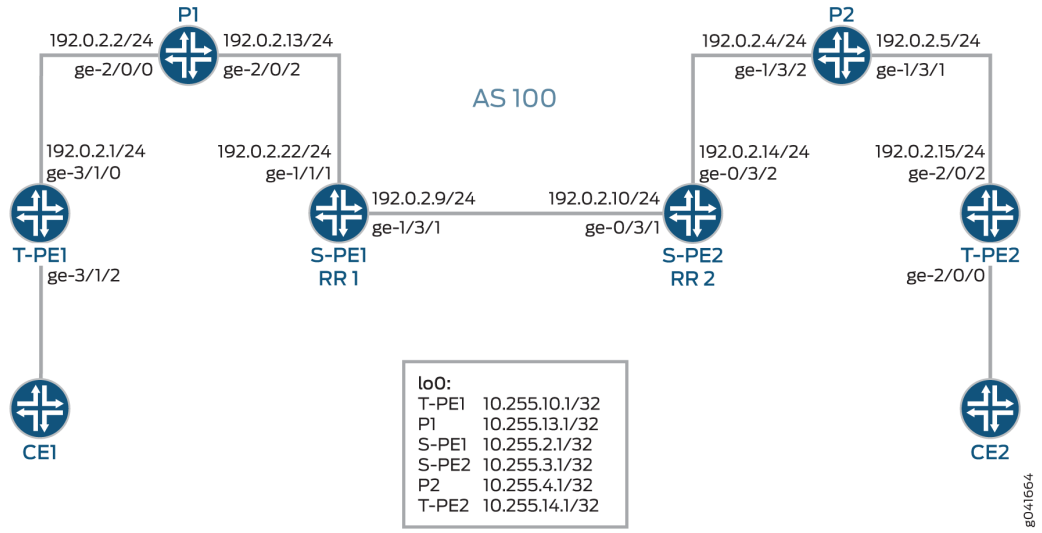
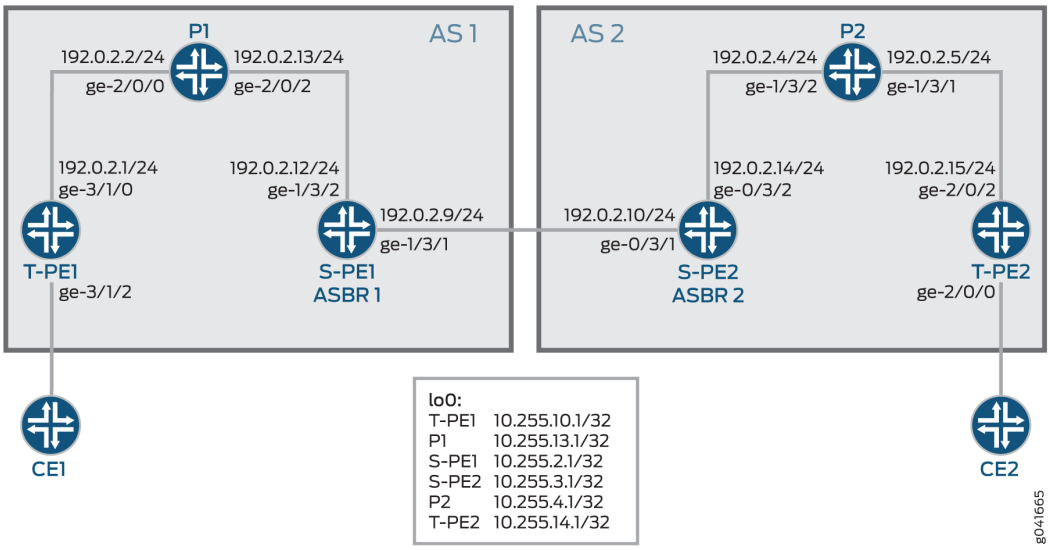


Figure 120: Inter-AS Multisegment Pseudowire



The following sections provide information about how an MS-PW is established in an interarea and inter-AS scenario.

Minimum Configuration Requirements on S-PE

In order to dynamically discover both ends of an SS-PW and set up a T-LDP session dynamically, the following is required:

- For interarea MS-PW, each S-PE plays both an ABR and BGP route reflector role.

  In the interarea case, as seen in Figure 119 on page 1877, the S-PE plays a BGP route reflector role and reflects the BGP-AD route to its client. A BGP-AD route advertised by one T-PE eventually reaches its remote T-PE. Because of the next-hop-self set by each S-PE, the S-PE or T-PE that receives a BGP-AD route can always discover the S-PE that advertises the BGP-AD in its local AS or local area through the BGP next hop.

- For inter-AS MS-PW, each S-PE plays either an ASBR or a BGP route reflector role.

  In an MS-PW, the two T-PEs initiate a BGP-AD route respectively. When the S-PE receives the BGP-AD route through either the IBGP session with the T-PE or through a regular BGP-RR, it sets the next-hop-self before re-advertising the BGP-AD route to one or more of its EBGP peers in the inter-AS case, as seen in Figure 120 on page 1877.

- Each S-PE must set next-hop-self when re-advertising or reflecting a BGP-AD route for the MS-PW.

Active and Passive Role of T-PE

To ensure that the same set of S-PEs are being used for a MS-PW in both directions, the two T-PEs play different roles in terms of FEC 129 signaling. This is to avoid different paths being chosen by T-PE1 and T-PE2 when each S-PE is dynamically selected for an MS-PW.

When an MS-PW is signaled using FEC 129, each T-PE might independently start signaling the MS-PW. The signaling procedure can result in an attempt to set up each direction of the MS-PW through different S-PEs.

To avoid this situation, one of the T-PEs must start the pseudowire signaling (active role), while the other waits to receive the LDP label mapping before sending the respective pseudowire LDP label mapping message (passive role). When the MS-PW path is dynamically placed, the active T-PE (the Source T-PE) and the passive T-PE (the Target T-PE) must be identified before signaling is initiated for a given MS-PW. The determination of which T-PE assumes the active role is done based on the SAII value, where the T-PE that has a larger SAII value plays the active role.

In this example, the SAII values of T-PE1 and T-PE 2 are `800:800:800` and `700:700:700`, respectively. Since T-PE1 has a higher SAII value, it assumes the active role and T-PE2 assumes the passive role.

Directions for Establishing an MS-PW

The directions used by the S-PE for setting up the MS-PW are:

- Forwarding direction—From an active T-PE to a passive T-PE.

  In this direction, the S-PEs perform a BGP-AD route lookup to determine the next-hop S-PE to send the label mapping message.

- Reverse direction—From a passive T-PE to an active T-PE.

In this direction, the S-PEs do not perform a BGP-AD route lookup, because the label mapping messages are received from the T-PEs, and the stitching routes are installed in the S-PEs.

In this example, the MS-PW is established in the forwarding direction from T-PE1 to T-PE2. When the MS-PW is placed from T-PE2 to T-PE1, the MS-PW is established in the reverse direction.

Autodiscovery and Dynamic Selection of S-PE

A new AFI and SAFI value is defined in BGP to support the MS-PWs based on type 2 AII. This new address family supports autodiscovery of S-PEs. This address family must be configured on both the TPEs and SPEs.

It is the responsibility of the Layer 2 VPN component to dynamically select the next S-PE to use along the MS-PW in the forwarding direction.

- In the forwarding direction, the selection of the next S-PE is based on the BGP-AD route advertised by the BGP and pseudowire FEC information sent by the LDP. The BGP-AD route is initiated by the passive T-PE (T-PE2) in the reverse direction while the pseudowire FEC information is sent by LDP from the active T-PE (T-PE1) in the forwarding direction.
- In the reverse direction, the next S-PE (S-PE2) or the active T-PE (T-PE1) is obtained by looking up the S-PE (S-PE1) that it used to set up the pseudowire in the forwarding direction.

Provisioning a T-PE

To support FEC 129 type 2 AII, the T-PE needs to configure its remote T-PE's IP address, a global ID, and an attachment circuit ID. Explicit paths where a set of S-PEs to use is explicitly specified on a T-PE is not supported. This eliminates the need to provision each S-PE with a type 2 AII.

Stitching an MS-PW

An S-PE performs the following MPLS label operations before forwarding the received label mapping message to the next S-PE:

1. Pops the MPLS tunnel label.
2. Pops the VC label.
3. Pushes a new VC label.
4. Pushes an MPLS tunnel label used for the next segment.

Establishing an MS-PW

After completing the necessary configuration, an MS-PW is established in the following manner:

1. The SAII values are exchanged between T-PE1 and T-PE2 using BGP.

T-PE1 assumes the active T-PE role, because it is configured with a higher SAII value. T-PE2 becomes the passive T-PE.

2. T-PE1 receives the BGP-AD route originated by T-PE2. It compares the AII values obtained from T-PE2 in the received BGP-AD route against the AII values provisioned locally.
3. If the AII values match, T-PE1 performs a BGP-AD route lookup to elect the first S-PE (S-PE1).
4. T-PE1 sends an LDP label mapping message to S-PE1.
5. Using the BGP-AD route originated from T-PE2, and the LDP label mapping message received from T-PE1, S-PE1 selects the next S-PE (S-PE2) in the forwarding direction.

   To do this, S-PE1 compares SAII obtained from the BGP-AD route against the TAI from the LDP label mapping message.
6. If the AII values match, S-PE1 finds S-PE2 through the BGP next hop associated with the BGP-AD route.
7. The process of selecting S-PE goes on until the last S-PE establishes a T-LDP session with T-PE2. When T-PE2 receives the LDP label mapping message from the last S-PE (S-PE2), it initiates its own label mapping message and sends it back to S-PE2.
8. When all the label mapping messages are received on S-PE1 and S-PE2, the S-PEs install the stitching routes. Thus, when the MS-PW is established in the reverse direction, the S-PEs need not perform BGP-AD route lookup to determine its next hop as it did in the forwarding direction.

OAM Support for an MS-PW

After the MS-PW is established, the following OAM capabilities can be executed from the T-PE devices:

- Ping
  - End-to-End Connectivity Verification Between T-PEs

    If T-PE1, S-PEs, and T-PE2 support Control Word (CW), the pseudowire control plane automatically negotiates the use of the CW. Virtual Circuit Connectivity Verification (VCCV) Control Channel (CC) Type 3 will function correctly whether or not the CW is enabled on the pseudowire. However, VCCV Type 1, which is used for end-to-end verification only, is only supported if the CW is enabled.

    The following is a sample:

    Ping from T-P1 to T-PE2

```
user@T-PE1> ping mpls l2vpn fec129 instance instance-name local-id SAII of T-PE1 remote-pe-address address of T-PE2 remote-id TAII of T-PE2
```

or

```
user@T-PE1> ping mpls l2vpn fec129 interface CE1-facing interface
```

- Partial Connectivity Verification from T-PE to Any S-PE

  To trace part of an MS-PW, the TTL of the pseudowire label can be used to force the VCCV message to pop out at an intermediate node. When the TTL expires, the S-PE can determine that the packet is a VCCV packet either by checking the CW or by checking for a valid IP header with UDP destination port 3502 (if the CW is not in use). The packet should then be diverted to VCCV processing.

  If T-PE1 sends a VCCV message with the TTL of the pseudowire label equal to 1, the TTL expires at the S-PE. T-PE1 can thus verify the first segment of the pseudowire.

  The VCCV packet is built according to RFC 4379. All the information necessary to build the VCCV LSP ping packet is collected by inspecting the S-PE TLVs. This use of the TTL is subject to the caution expressed in RFC 5085. If a penultimate LSR between S-PEs or between an S-PE and a T-PE manipulates the pseudowire label TTL, the VCCV message might not emerge from the MS-PW at the correct S-PE.

  The following is a sample:

  Ping from T-PE1 to S-PE

  ```
  user@T-PE1> ping mpls l2vpn fec129 interface CE1-facing interface bottom-label-ttl segment
  ```

  The `bottom-label-ttl` value is 1 for S-PE1 and 2 for S-PE2.

  The `bottom-label-ttl` statement sets the correct VC label TTL, so the packets are popped to the correct SS-PW for VCCV processing.

  NOTE: Junos OS supports VCCV Type 1 and Type 3 for the MS-PW OAM capability. VCCV Type 2 is not supported.

- Traceroute

Traceroute tests each S-PE along the path of the MS-PW in a single operation similar to LSP trace. This operation is able to determine the actual data path of the MS-PW, and is used for dynamically signaled MS-PWs.

```
user@T-PE1> traceroute mpls l2vpn fec129 interface CE1-facing interface
```

- Bidirectional Forwarding Detection

  Bidirectional Forwarding Detection (BFD) is a detection protocol designed to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols. In addition to fast forwarding path failure detection, BFD provides a consistent failure detection method for network administrators. The router or switch can be configured to log a system log (syslog) message when BFD goes down.

```
user@T-PE1> show bfd session extensive
```

Configuration

IN THIS SECTION

- Configuring an Interarea MS-PW | **1882**
- Configuring an Inter-AS MS-PW | **1896**

Configuring an Interarea MS-PW

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the `[edit]` hierarchy level.

T-PE1

```
set interfaces ge-3/1/0 unit 0 family inet address 192.0.2.1/24
set interfaces ge-3/1/0 unit 0 family mpls
set interfaces ge-3/1/2 encapsulation ethernet-ccc
set interfaces ge-3/1/2 unit 0
```

```
set interfaces lo0 unit 0 family inet address 10.255.10.1/32 primary
set routing-options autonomous-system 100
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp family l2vpn auto-discovery-mspw
set protocols bgp group mspw type internal
set protocols bgp group mspw local-address 10.255.10.1
set protocols bgp group mspw neighbor 10.255.2.1
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols ldp interface lo0.0
set routing-instances ms-pw instance-type l2vpn
set routing-instances ms-pw interface ge-3/1/2.0
set routing-instances ms-pw route-distinguisher 10.10.10.10:15
set routing-instances ms-pw l2vpn-id l2vpn-id:100:15
set routing-instances ms-pw vrf-target target:100:115
set routing-instances ms-pw protocols l2vpn site CE1 source-attachment-identifier 800:800:800
set routing-instances ms-pw protocols l2vpn site CE1 interface ge-3/1/2.0 target-attachment-
identifier 700:700:700
set routing-instances ms-pw protocols l2vpn pseudowire-status-tlv
set routing-instances ms-pw protocols l2vpn oam bfd-liveness-detection minimum-interval 300
```

P1

```
set interfaces ge-2/0/0 unit 0 family inet address 192.0.2.2/24
set interfaces ge-2/0/0 unit 0 family mpls
set interfaces ge-2/0/2 unit 0 family inet address 192.0.2.13/24
set interfaces ge-2/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.13.1/32 primary
set routing-options autonomous-system 100
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols ldp interface lo0.0
```

S-PE1 (RR 1)

```
set interfaces ge-1/3/1 unit 0 family inet address 192.0.2.9/24
set interfaces ge-1/3/1 unit 0 family mpls
set interfaces ge-1/3/2 unit 0 family inet address 192.0.2.22/24
set interfaces ge-1/3/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.2.1/32 primary
set routing-options autonomous-system 100
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp family l2vpn auto-discovery-mspw
set protocols bgp group mspw type internal
set protocols bgp group mspw local-address 10.255.2.1
set protocols bgp group mspw export next-hop-self
set protocols bgp group mspw cluster 203.0.113.0
set protocols bgp group mspw neighbor 10.255.10.1
set protocols bgp group mspw neighbor 10.255.3.1
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols ldp interface lo0.0
set policy-options policy-statement next-hop-self then next-hop self
set policy-options policy-statement send-inet0 from protocol bgp
set policy-options policy-statement send-inet0 then accept
```

S-PE2 (RR 2)

```
set interfaces ge-0/3/1 unit 0 family inet address 192.0.2.10/24
set interfaces ge-0/3/1 unit 0 family mpls
set interfaces ge-0/3/2 unit 0 family inet address 192.0.2.14/24
set interfaces ge-0/3/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.3.1/32 primary
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp family l2vpn auto-discovery-mspw
set protocols bgp group mspw type internal
set protocols bgp group mspw local-address 10.255.3.1
set protocols bgp group mspw export next-hop-self
set protocols bgp group mspw cluster 198.51.100.0
```

```
set protocols bgp group mspw neighbor 10.255.2.1
set protocols bgp group mspw neighbor 10.255.14.1
set protocols bgp group int type internal
set protocols bgp group int local-address 10.255.3.1
set protocols bgp group int neighbor 10.255.2.1
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols ldp interface lo0.0
set policy-options policy-statement next-hop-self then next-hop self
set policy-options policy-statement send-inet0 from protocol bgp
set policy-options policy-statement send-inet0 then accept
```

P2

```
set interfaces ge-1/3/1 unit 0 family inet address 192.0.2.5/24
set interfaces ge-1/3/1 unit 0 family mpls
set interfaces ge-1/3/2 unit 0 family inet address 192.0.2.4/24
set interfaces ge-1/3/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.4.1/32 primary
set routing-options autonomous-system 100
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols ldp interface lo0.0
```

T-PE2

```
set interfaces ge-2/0/0 encapsulation ethernet-ccc
set interfaces ge-2/0/0 unit 0
set interfaces ge-2/0/2 unit 0 family inet address 192.0.2.15/24
set interfaces ge-2/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.14.1/32 primary
set routing-options autonomous-system 100
set protocols mpls interface all
```

```
set protocols mpls interface fxp0.0 disable
set protocols bgp family l2vpn auto-discovery-mspw
set protocols bgp group mspw type internal
set protocols bgp group mspw local-address 10.255.14.1
set protocols bgp group mspw neighbor 10.255.3.1
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols ldp interface lo0.0
set routing-instances ms-pw instance-type l2vpn
set routing-instances ms-pw interface ge-2/0/0.0
set routing-instances ms-pw route-distinguisher 10.10.10.10:15
set routing-instances ms-pw l2vpn-id l2vpn-id:100:15
set routing-instances ms-pw vrf-target target:100:115
set routing-instances ms-pw protocols l2vpn site CE2 source-attachment-identifier 700:700:700
set routing-instances ms-pw protocols l2vpn site CE2 interface ge-2/0/0.0 target-attachment-
identifier 800:800:800
set routing-instances ms-pw protocols l2vpn pseudowire-status-tlv
set routing-instances ms-pw protocols l2vpn oam bfd-liveness-detection minimum-interval 300
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode*.

To configure T-PE1 in the interarea scenario:

NOTE: Repeat this procedure for the T-PE2 device in the MPLS domain, after modifying the appropriate interface names, addresses, and other parameters.

1. Configure the T-PE1 interfaces.

```
[edit interfaces]
user@T-PE1# set ge-3/1/0 unit 0 family inet address 192.0.2.1/24
user@T-PE1# set ge-3/1/0 unit 0 family mpls
user@T-PE1# set ge-3/1/2 encapsulation ethernet-ccc
user@T-PE1# set ge-3/1/2 unit 0
user@T-PE1# set lo0 unit 0 family inet address 10.255.10.1/32 primary
```

2. Set the autonomous system number.

```
[edit routing-options]
user@T-PE1# set autonomous-system 100
```

3. Enable MPLS on all the interfaces of T-PE1, excluding the management interface.

```
[edit protocols]
user@T-PE1# set mpls interface all
user@T-PE1# set mpls interface fxp0.0 disable
```

4. Enable autodiscovery of intermediate S-PEs that make up the MS-PW using BGP.

```
[edit protocols]
user@T-PE1# set bgp family l2vpn auto-discovery-mspw
```

5. Configure the BGP group for T-PE1.

```
[edit protocols]
user@T-PE1# set bgp group mspw type internal
```

6. Assign local and neighbor addresses to the mspw group for T-PE1 to peer with S-PE1.

```
[edit protocols]
user@T-PE1# set bgp group mspw local-address 10.255.10.1
user@T-PE1# set bgp group mspw neighbor 10.255.2.1
```

7. Configure OSPF on all the interfaces of T-PE1, excluding the management interface.

```
[edit protocols]
user@T-PE1# set ospf area 0.0.0.0 interface lo0.0
user@T-PE1# set ospf area 0.0.0.0 interface all
user@T-PE1# set ospf area 0.0.0.0 interface fxp0.0 disable
```

8. Configure LDP on all the interfaces of T-PE1, excluding the management interface.

```
[edit protocols]
user@T-PE1# set ldp interface all
user@T-PE1# set ldp interface fxp0.0 disable
user@T-PE1# set ldp interface lo0.0
```

9. Configure the Layer 2 VPN routing instance on T-PE1.

```
[edit routing-instances]
user@T-PE1# set ms-pw instance-type l2vpn
```

10. Assign the interface name for the mspw routing instance.

```
[edit routing-instances]
user@T-PE1# set ms-pw interface ge-3/1/2.0
```

11. Configure the route distinguisher for the mspw routing instance.

```
[edit routing-instances]
user@T-PE1# set ms-pw route-distinguisher 10.10.10.10:15
```

12. Configure the Layer 2 VPN ID community for FEC 129 MS-PW.

```
[edit routing-instances]
user@T-PE1# set ms-pw l2vpn-id l2vpn-id:100:15
```

13. Configure a VPN routing and forwarding (VRF) target for the mspw routing instance.

```
[edit routing-instances]
user@T-PE1# set ms-pw vrf-target target:100:115
```

14. Configure the source attachment identifier (SAI) value using Layer 2 VPN as the routing protocol for the mspw routing instance.

```
[edit routing-instances]
user@T-PE1# set ms-pw protocols l2vpn site CE1 source-attachment-identifier 800:800:800
```

15. Assign the interface name that connects the CE1 site to the VPN, and configure the target attachment identifier (TAI) value using Layer 2 VPN as the routing protocol for the mspw routing instance.

```
[edit routing-instances]
user@T-PE1# set ms-pw protocols l2vpn site CE1 interface ge-3/1/2.0 target-attachment-
identifier 700:700:700
```

16. (Optional) Configure T-PE1 to send MS-PW status TLVs.

```
[edit routing-instances]
user@T-PE1# set ms-pw protocols l2vpn pseudowire-status-tlv
```

17. (Optional) Configure OAM capabilities for the VPN.

```
[edit routing-instances]
user@T-PE1# set ms-pw protocols l2vpn oam bfd-liveness-detection minimum-interval 300
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode*.

To configure S-PE1 (RR 1) in the interarea scenario:

NOTE: Repeat this procedure for the S-PE2 (RR 2) device in the MPLS domain, after modifying the appropriate interface names, addresses, and other parameters.

1. Configure the S-PE1 interfaces.

```
[edit interfaces]
user@S-PE1# set ge-1/3/1 unit 0 family inet address 192.0.2.9/24
user@S-PE1# set ge-1/3/1 unit 0 family mpls
user@S-PE1# set ge-1/3/2 unit 0 family inet address 192.0.2.22/24
user@S-PE1# set ge-1/3/2 unit 0 family mpls
user@S-PE1# set lo0 unit 0 family inet address 10.255.2.1/32 primary
```

2. Set the autonomous system number.

```
[edit routing-options]
user@S-PE1# set autonomous-system 100
```

3. Enable MPLS on all the interfaces of T-PE1, excluding the management interface.

```
[edit protocols]
user@S-PE1# set mpls interface all
user@S-PE1# set mpls interface fxp0.0 disable
```

4. Enable autodiscovery of S-PE using BGP.

```
[edit protocols]
user@S-PE1# set bgp family l2vpn auto-discovery-mspw
```

5. Configure the BGP group for S-PE1.

```
[edit protocols]
user@S-PE1# set bgp group mspw type internal
```

6. Configure S-PE1 to act as a route reflector.

```
[edit protocols]
user@S-PE1# set bgp group mspw export next-hop-self
user@S-PE1# set bgp group mspw cluster 203.0.113.0
```

7. Assign local and neighbor addresses to the mspw group for S-PE1 to peer with T-PE1 and S-PE2.

```
[edit protocols]
user@S-PE1# set bgp group mspw local-address 10.255.2.1
user@S-PE1# set bgp group mspw neighbor 10.255.10.1 (to T-PE1)
user@S-PE1# set bgp group mspw neighbor 10.255.3.1 (to S-PE2)
```

8. Configure OSPF on all the interfaces of S-PE1, excluding the management interface.

```
[edit protocols]
user@S-PE1# set ospf area 0.0.0.0 interface all
user@S-PE1# set ospf area 0.0.0.0 interface fxp0.0 disable
user@S-PE1# set ospf area 0.0.0.0 interface lo0.0
```

9. Configure LDP on all the interfaces of S-PE1, excluding the management interface.

```
[edit protocols]
user@S-PE1# set ldp interface all
user@S-PE1# set ldp interface fxp0.0 disable
user@S-PE1# set ldp interface lo0.0
```

10. Define the policy for enabling next-hop-self and accepting BGP traffic on S-PE1.

```
[edit policy-options]
user@S-PE1# set policy-statement next-hop-self then next-hop self
user@S-PE1# set policy-statement send-inet0 from protocol bgp
user@S-PE1# set policy-statement send-inet0 then accept
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols`, `show routing-instances`, `show routing-options`, and `show policy-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

T-PE1

```
user@T-PE1# show interfaces
ge-3/1/0 {
```

```
    unit 0 {
        family inet {
            address 192.0.2.1/24;
        }
        family mpls;
    }
}
ge-3/1/2 {
    encapsulation ethernet-ccc;
    unit 0;
}
lo0 {
    unit 0 {
        family inet {
            address 10.255.10.1/32 {
                primary;
            }
        }
    }
}
```

```
user@T-PE1# show routing-options
autonomous-system 100;
```

```
user@T-PE1# show protocols
mpls {
    interface all;
    interface fxp0.0 {
        disable;
    }
}
bgp {
    family l2vpn {
        auto-discovery-mspw;
    }
    group mspw {
        type internal;
        local-address 10.255.10.1;
        neighbor 10.255.2.1;
    }
```

```
}
ospf {
    area 0.0.0.0 {
        interface all;
        interface fxp0.0 {
            disable;
        }
        interface lo0.0;
    }
}
ldp {
    interface all;
    interface fxp0.0 {
        disable;
    }
    interface lo0.0;
}
```

```
user@T-PE1# show routing-instances
ms-pw {
    instance-type l2vpn;
    interface ge-3/1/2.0;
    route-distinguisher 10.10.10.10:15;
    l2vpn-id l2vpn-id:100:15;
    vrf-target target:100:115;
    protocols {
        l2vpn {
            site CE1 {
                source-attachment-identifier 800:800:800;
                interface ge-3/1/2.0 {
                    target-attachment-identifier 700:700:700;
                }
            }
            pseudowire-status-tlv;
            oam {
                bfd-liveness-detection {
                    minimum-interval 300;
                }
            }
        }
```

```
    }
}
```

S-PE1 (RR 1)

```
user@S-PE1# show interfaces
ge-1/3/1 {
    unit 0 {
        family inet {
            address 192.0.2.9/24;
        }
        family mpls;
    }
}
ge-1/3/2 {
    unit 0 {
        family inet {
            address 192.0.2.22/24;
        }
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.255.2.1/32 {
                primary;
            }
        }
    }
}
```

```
user@S-PE1# show routing-options
autonomous-system 100;
```

```
user@S-PE1# show protocols
mpls {
    interface all;
    interface fxp0.0 {
```

```
        disable;
    }
}
bgp {
    family l2vpn {
        auto-discovery-mspw;
    }
    group mspw {
        type internal;
        local-address 10.255.2.1;
        export next-hop-self;
        cluster 203.0.113.0;
        neighbor 10.255.10.1;
        neighbor 10.255.3.1;
    }
}
ospf {
    area 0.0.0.0 {
        interface lo0.0;
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
}
ldp {
    interface all;
    interface fxp0.0 {
        disable;
    }
    interface lo0.0;
}
```

```
user@S-PE1# show policy-options
policy-statement next-hop-self {
    then {
        next-hop self;
    }
}
policy-statement send-inet0 {
    from protocol bgp;
```

```
    then accept;
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Configuring an Inter-AS MS-PW

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level.

T-PE1

```
set interfaces ge-3/1/0 unit 0 family inet address 192.0.2.1/24
set interfaces ge-3/1/0 unit 0 family mpls
set interfaces ge-3/1/2 encapsulation ethernet-ccc
set interfaces ge-3/1/2 unit 0
set interfaces lo0 unit 0 family inet address 10.255.10.1/32 primary
set routing-options autonomous-system 1
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp family l2vpn auto-discovery-mspw
set protocols bgp group mspw type internal
set protocols bgp group mspw local-address 10.255.10.1
set protocols bgp group mspw neighbor 10.255.2.1
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols ldp interface lo0.0
set routing-instances ms-pw instance-type l2vpn
set routing-instances ms-pw interface ge-3/1/2.0
set routing-instances ms-pw route-distinguisher 10.10.10.10:15
set routing-instances ms-pw l2vpn-id l2vpn-id:100:15
set routing-instances ms-pw vrf-target target:100:115
set routing-instances ms-pw protocols l2vpn site CE1 source-attachment-identifier 800:800:800
set routing-instances ms-pw protocols l2vpn site CE1 interface ge-3/1/2.0 target-attachment-
identifier 700:700:700
```

```
set routing-instances ms-pw protocols l2vpn pseudowire-status-tlv
set routing-instances ms-pw protocols l2vpn oam bfd-liveness-detection minimum-interval 300
```

P1

```
set interfaces ge-2/0/0 unit 0 family inet address 192.0.2.2/24
set interfaces ge-2/0/0 unit 0 family mpls
set interfaces ge-2/0/2 unit 0 family inet address 192.0.2.13/24
set interfaces ge-2/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.13.1/32 primary
set routing-options autonomous-system 1
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols ldp interface lo0.0
```

S-PE1 (ASBR 1)

```
set interfaces ge-1/3/1 unit 0 family inet address 192.0.2.9/24
set interfaces ge-1/3/1 unit 0 family mpls
set interfaces ge-1/3/2 unit 0 family inet address 192.0.2.22/24
set interfaces ge-1/3/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.2.1/32 primary
set routing-options autonomous-system 1
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp family l2vpn auto-discovery-mspw
set protocols bgp group to_T-PE1 type internal
set protocols bgp group to_T-PE1 local-address 10.255.2.1
set protocols bgp group to_T-PE1 export next-hop-self
set protocols bgp group to_T-PE1 neighbor 10.255.10.1
set protocols bgp group to_S-PE2 type external
set protocols bgp group to_S-PE2 local-address 10.255.2.1
set protocols bgp group to_S-PE2 peer-as 2
set protocols bgp group to_S-PE2 neighbor 10.255.3.1 multihop ttl 1
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface all
```

```
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols ldp interface lo0.0
set policy-options policy-statement next-hop-self then next-hop self
```

S-PE2 (ASBR 2)

```
set interfaces ge-0/3/1 unit 0 family inet address 192.0.2.10/24
set interfaces ge-0/3/1 unit 0 family mpls
set interfaces ge-0/3/2 unit 0 family inet address 192.0.2.14/24
set interfaces ge-0/3/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.3.1/32 primary
set routing-options autonomous-system 2
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp family l2vpn auto-discovery-mspw
set protocols bgp group to_T-PE2 type internal
set protocols bgp group to_T-PE2 local-address 10.255.3.1
set protocols bgp group to_T-PE2 export next-hop-self
set protocols bgp group to_T-PE2 neighbor 10.255.14.1
set protocols bgp group to_S-PE1 type external
set protocols bgp group to_S-PE1 local-address 10.255.3.1
set protocols bgp group to_S-PE1 peer-as 1
set protocols bgp group to_S-PE1 neighbor 10.255.2.1 multihop ttl 1
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols ldp interface lo0.0
set policy-options policy-statement next-hop-self then next-hop self
```

P2

```
set interfaces ge-1/3/1 unit 0 family inet address 192.0.2.5/24
set interfaces ge-1/3/1 unit 0 family mpls
set interfaces ge-1/3/2 unit 0 family inet address 192.0.2.4/24
set interfaces ge-1/3/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.4.1/32 primary
set routing-options autonomous-system 2
```

```
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols ldp interface lo0.0
```

T-PE2

```
set interfaces ge-2/0/0 encapsulation ethernet-ccc
set interfaces ge-2/0/0 unit 0
set interfaces ge-2/0/2 unit 0 family inet address 192.0.2.15/24
set interfaces ge-2/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.14.1/32 primary
set routing-options autonomous-system 2
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp family l2vpn auto-discovery-mspw
set protocols bgp group mspw type internal
set protocols bgp group mspw local-address 10.255.14.1
set protocols bgp group mspw neighbor 10.255.3.1
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols ldp interface lo0.0
set routing-instances ms-pw instance-type l2vpn
set routing-instances ms-pw interface ge-2/0/0.0
set routing-instances ms-pw route-distinguisher 10.10.10.10:15
set routing-instances ms-pw l2vpn-id l2vpn-id:100:15
set routing-instances ms-pw vrf-target target:100:115
set routing-instances ms-pw protocols l2vpn site CE2 source-attachment-identifier 700:700:700
set routing-instances ms-pw protocols l2vpn site CE2 interface ge-2/0/0.0 target-attachment-
identifier 800:800:800
set routing-instances ms-pw protocols l2vpn pseudowire-status-tlv
set routing-instances ms-pw protocols l2vpn oam bfd-liveness-detection minimum-interval 300
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode*.

To configure the T-PE1 router in the inter-AS scenario:

NOTE: Repeat this procedure for the T-PE2 device in the MPLS domain, after modifying the appropriate interface names, addresses, and other parameters.

1. Configure the T-PE1 interfaces.

```
[edit interfaces]
user@T-PE1# set ge-3/1/0 unit 0 family inet address 192.0.2.1/24
user@T-PE1# set ge-3/1/0 unit 0 family mpls
user@T-PE1# set ge-3/1/2 encapsulation ethernet-ccc
user@T-PE1# set ge-3/1/2 unit 0
user@T-PE1# set lo0 unit 0 family inet address 10.255.10.1/32 primary
```

2. Set the autonomous system number.

```
[edit routing-options]
user@T-PE1# set autonomous-system 1
```

3. Enable MPLS on all the interfaces of T-PE1, excluding the management interface.

```
[edit protocols]
user@T-PE1# set mpls interface all
user@T-PE1# set mpls interface fxp0.0 disable
```

4. Enable autodiscovery of intermediate S-PEs that make up the MS-PW using BGP.

```
[edit protocols]
user@T-PE1# set bgp family l2vpn auto-discovery-mspw
```

5. Configure the BGP group for T-PE1.

```
[edit protocols]
user@T-PE1# set bgp group mspw type internal
```

6. Assign local and neighbor addresses to the mspw group for T-PE1 to peer with S-PE1.

```
[edit protocols]
user@T-PE1# set bgp group mspw local-address 10.255.10.1
user@T-PE1# set bgp group mspw neighbor 10.255.2.1
```

7. Configure OSPF on all the interfaces of T-PE1, excluding the management interface.

```
[edit protocols]
user@T-PE1# set ospf area 0.0.0.0 interface lo0.0
user@T-PE1# set ospf area 0.0.0.0 interface all
user@T-PE1# set ospf area 0.0.0.0 interface fxp0.0 disable
```

8. Configure LDP on all the interfaces of T-PE1, excluding the management interface.

```
[edit protocols]
user@T-PE1# set ldp interface all
user@T-PE1# set ldp interface fxp0.0 disable
user@T-PE1# set ldp interface lo0.0
```

9. Configure the Layer 2 VPN routing instance on T-PE1.

```
[edit routing-instances]
user@T-PE1# set ms-pw instance-type l2vpn
```

10. Assign the interface name for the mspw routing instance.

```
[edit routing-instances]
user@T-PE1# set ms-pw interface ge-3/1/2.0
```

11. Configure the route distinguisher for the mspw routing instance.

```
[edit routing-instances]
user@T-PE1# set ms-pw route-distinguisher 10.10.10.10:15
```

12. Configure the Layer 2 VPN ID community for FEC 129 MS-PW.

```
[edit routing-instances]
user@T-PE1# set ms-pw l2vpn-id l2vpn-id:100:15
```

13. Configure a VPN routing and forwarding (VRF) target for the mspw routing instance.

```
[edit routing-instances]
user@T-PE1# set ms-pw vrf-target target:100:115
```

14. Configure the source attachment identifier (SAI) value using Layer 2 VPN as the routing protocol for the mspw routing instance.

```
[edit routing-instances]
user@T-PE1# set ms-pw protocols l2vpn site CE1 source-attachment-identifier 800:800:800
```

15. Assign the interface name that connects the CE1 site to the VPN, and configure the target attachment identifier (TAI) value using Layer 2 VPN as the routing protocol for the mspw routing instance.

```
[edit routing-instances]
user@T-PE1# set ms-pw protocols l2vpn site CE1 interface ge-3/1/2.0 target-attachment-
identifier 700:700:700
```

16. (Optional) Configure T-PE1 to send MS-PW status TLVs.

```
[edit routing-instances]
user@T-PE1# set ms-pw protocols l2vpn pseudowire-status-tlv
```

17. (Optional) Configure OAM capabilities for the VPN.

```
[edit routing-instances]
user@T-PE1# set ms-pw protocols l2vpn oam bfd-liveness-detection minimum-interval 300
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode*.

To configure S-PE1 (ASBR 1) in the inter-AS scenario:

NOTE: Repeat this procedure for the S-PE2 (ASBR 2) device in the MPLS domain, after modifying the appropriate interface names, addresses, and other parameters.

1. Configure S-PE1 (ASBR 1) interfaces.

```
[edit interfaces]
user@S-PE1# set ge-1/3/1 unit 0 family inet address 192.0.2.9/24
user@S-PE1# set ge-1/3/1 unit 0 family mpls
user@S-PE1# set ge-1/3/2 unit 0 family inet address 192.0.2.22/24
user@S-PE1# set ge-1/3/2 unit 0 family mpls
user@S-PE1# set lo0 unit 0 family inet address 10.255.2.1/32 primary
```

2. Set the autonomous system number.

```
[edit routing-options]
user@S-PE1# set autonomous-system 1
```

3. Enable MPLS on all the interfaces of S-PE1 (ASBR 1), excluding the management interface.

```
[edit protocols]
user@S-PE1# set mpls interface all
user@S-PE1# set mpls interface fxp0.0 disable
```

4. Enable autodiscovery of S-PE using BGP.

```
[edit protocols]
user@S-PE1# set bgp family l2vpn auto-discovery-mspw
```

5. Configure the IBGP group for S-PE1 (ASBR 1) to peer with T-PE1.

```
[edit protocols]
user@S-PE1# set bgp group to_T-PE1 type internal
```

6. Configure the IBGP group parameters.

```
[edit protocols]
user@S-PE1# set bgp group to_T-PE1 local-address 10.255.2.1
user@S-PE1# set bgp group to_T-PE1 export next-hop-self
user@S-PE1# set bgp group to_T-PE1 neighbor 10.255.10.1
```

7. Configure the EBGP group for S-PE1 (ASBR 1) to peer with S-PE2 (ASBR 2).

```
[edit protocols]
user@S-PE1# set bgp group to_S-PE2 type external
```

8. Configure the EBGP group parameters.

```
[edit protocols]
user@S-PE1# set bgp group to_S-PE2 local-address 10.255.2.1
user@S-PE1# set bgp group to_S-PE2 peer-as 2
user@S-PE1# set bgp group to_S-PE2 neighbor 10.255.3.1 multihop ttl 1
```

9. Configure OSPF on all the interfaces of S-PE1 (ASBR 1), excluding the management interface.

```
[edit protocols]
user@S-PE1# set ospf area 0.0.0.0 interface all
user@S-PE1# set ospf area 0.0.0.0 interface fxp0.0 disable
user@S-PE1# set ospf area 0.0.0.0 interface lo0.0 passive
```

10. Configure LDP on all the interfaces of S-PE1 (ASBR 1), excluding the management interface.

```
[edit protocols]
user@S-PE1# set ldp interface all
user@S-PE1# set ldp interface fxp0.0 disable
user@S-PE1# set ldp interface lo0.0
```

11. Define the policy for enabling next-hop-self on S-PE1 (ASBR 1).

```
[edit policy-options]
user@S-PE1# set policy-statement next-hop-self then next-hop self
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols`, `show routing-instances`, `show routing-options`, and `show policy-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

T-PE1

```
user@T-PE1# show interfaces
ge-3/1/0 {
    unit 0 {
        family inet {
            address 192.0.2.1/24;
        }
        family mpls;
    }
}
ge-3/1/2 {
    encapsulation ethernet-ccc;
    unit 0;
}
lo0 {
    unit 0 {
        family inet {
            address 10.255.10.1/32 {
                primary;
            }
        }
```

```
    }
}
```

```
user@T-PE1# show routing-options
autonomous-system 1;
```

```
user@T-PE1# show protocols
mpls {
    interface all;
    interface fxp0.0 {
        disable;
    }
}
bgp {
    family l2vpn {
        auto-discovery-mspw;
    }
    group mspw {
        type internal;
        local-address 10.255.10.1;
        neighbor 10.255.2.1;
    }
}
ospf {
    area 0.0.0.0 {
        interface all;
        interface fxp0.0 {
            disable;
        }
        interface lo0.0;
    }
}
ldp {
    interface all;
    interface fxp0.0 {
        disable;
    }
```

```
    interface lo0.0;
}
```

```
user@T-PE1# show routing-instances
ms-pw {
    instance-type l2vpn;
    interface ge-3/1/2.0;
    route-distinguisher 10.10.10.10:15;
    l2vpn-id l2vpn-id:100:15;
    vrf-target target:100:115;
    protocols {
        l2vpn {
            site CE1 {
                source-attachment-identifier 800:800:800;
                interface ge-3/1/2.0 {
                    target-attachment-identifier 700:700:700;
                }
            }
            pseudowire-status-tlv;
            oam {
                bfd-liveness-detection {
                    minimum-interval 300;
                }
            }
        }
    }
}
```

S-PE1 (RR 1)

```
user@S-PE1# show interfaces
ge-1/3/1 {
    unit 0 {
        family inet {
            address 192.0.2.9/24;
        }
        family mpls;
    }
}
ge-1/3/2 {
    unit 0 {
```

```
        family inet {
            address 192.0.2.22/24;
        }
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.255.2.1/32 {
                primary;
            }
        }
    }
}
```

```
user@T-PE1# show routing-options
autonomous-system 1;
```

```
user@S-PE1# show protocols
mpls {
    interface all;
    interface fxp0.0 {
        disable;
    }
}
bgp {
    family l2vpn {
        auto-discovery-mspw;
    }
    group to_T-PE1 {
        type internal;
        local-address 10.255.2.1;
        export next-hop-self;
        neighbor 10.255.10.1;
    }
    group to_S-PE2 {
        type external;
        local-address 10.255.2.1;
        peer-as 2;
```

```
        neighbor 10.255.3.1 {
            multihop {
                ttl 1;
            }
        }
    }
}
ospf {
    area 0.0.0.0 {
        interface lo0.0 {
            passive;
        }
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
}
ldp {
    interface all;
    interface fxp0.0 {
        disable;
    }
    interface lo0.0;
}
```

```
user@T-PE1# show policy-options
policy-statement next-hop-self {
    then {
        next-hop self;
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

IN THIS SECTION

- Verifying the Routes | **1910**
- Verifying the LDP Database | **1913**
- Checking the MS-PW Connections on T-PE1 | **1914**
- Checking the MS-PW Connections on S-PE1 | **1916**
- Checking the MS-PW Connections on S-PE2 | **1918**
- Checking the MS-PW Connections on T-PE2 | **1919**

Confirm that the configuration is working properly.

Verifying the Routes

Purpose

Verify that the expected routes are learned.

Action

From operational mode, run the `show route` command for the `bgp.l2vpn.1`, `ldp.l2vpn.1`, `mpls.0`, and `ms-pw.l2vpn.1` routing tables.

From operational mode, run the `show route table bgp.l2vpn.1` command.

```
user@T-PE1> show route table bgp.l2vpn.1
bgp.l2vpn.1: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.10.10:15:700:0.0.2.188:700/160 AD2
                   *[BGP/170] 16:13:11, localpref 100, from 10.255.2.1
                      AS path: 2 I, validation-state: unverified
                    > to 203.0.113.2 via ge-3/1/0.0, Push 300016
```

From operational mode, run the `show route table ldp.l2vpn.1` command.

```
user@T-PE1> show route table ldp.l2vpn.1
ldp.l2vpn.1: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.255.2.1:CtrlWord:5:100:15:700:0.0.2.188:700:800:0.0.3.32:800/304 PW2
                   *[LDP/9] 16:21:27
                      Discard
```

From operational mode, run the `show route table mpls.0` command.

```
user@T-PE1> show route table mpls.0
mpls.0: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 1w6d 00:28:26, metric 1
                      Receive
1                  *[MPLS/0] 1w6d 00:28:26, metric 1
                      Receive
2                  *[MPLS/0] 1w6d 00:28:26, metric 1
                      Receive
13                 *[MPLS/0] 1w6d 00:28:26, metric 1
                      Receive
299920             *[LDP/9] 1w5d 01:26:08, metric 1
                    > to 203.0.113.2 via ge-3/1/0.0, Pop
299920(S=0)        *[LDP/9] 1w5d 01:26:08, metric 1
                    > to 203.0.113.2 via ge-3/1/0.0, Pop
299936             *[LDP/9] 1w5d 01:26:08, metric 1
                    > to 203.0.113.2 via ge-3/1/0.0, Swap 300016
300096             *[LDP/9] 16:22:35, metric 1
                    > to 203.0.113.2 via ge-3/1/0.0, Swap 300128
300112             *[LDP/9] 16:22:35, metric 1
                    > to 203.0.113.2 via ge-3/1/0.0, Swap 300144
300128             *[LDP/9] 16:22:35, metric 1
                    > to 203.0.113.2 via ge-3/1/0.0, Swap 300160
300144             *[L2VPN/7] 16:22:33
                    > via ge-3/1/2.0, Pop       Offset: 4
ge-3/1/2.0         *[L2VPN/7] 16:22:33, metric2 1
                    > to 203.0.113.2 via ge-3/1/0.0, Push 300176, Push 300016(top) Offset: 252
```

From operational mode, run the `show route table ms-pw.l2vpn.1` command.

```
user@T-PE1> show route table ms-pw.l2vpn.1
ms-pw.l2vpn.1: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.10.10:15:700:0.0.2.188:700/160 AD2
                   *[BGP/170] 16:23:27, localpref 100, from 10.255.2.1
                      AS path: 2 I, validation-state: unverified
                    > to 203.0.113.2 via ge-3/1/0.0, Push 300016
10.10.10.10:15:800:0.0.3.32:800/160 AD2
                   *[L2VPN/170] 1w5d 23:25:19, metric2 1
                      Indirect
10.255.2.1:CtrlWord:5:100:15:700:0.0.2.188:700:800:0.0.3.32:800/304 PW2
                   *[LDP/9] 16:23:25
                      Discard
10.255.2.1:CtrlWord:5:100:15:800:0.0.3.32:800:700:0.0.2.188:700/304 PW2
                   *[L2VPN/7] 16:23:27, metric2 1
                    > to 203.0.113.2 via ge-3/1/0.0, Push 300016
```

Meaning

The output shows all the learned routes, including the autodiscovery (AD) routes.

The AD2 prefix format is `RD:SAII-type2`, where:

- `RD` is the route distinguisher value.
- `SAII-type2` is the type 2 source attachment identifier value.

The PW2 prefix format is `Neighbor_Addr:C:PWtype:l2vpn-id:SAII-type2:TAII-type2`, where:

- `Neighbor_Addr` is the loopback address of neighboring S-PE device.
- `C` indicates if Control Word (CW) is enabled or not.
  - `C` is `CtrlWord` if CW is set.
  - `C` is `NoCtrlWord` if CW is not set.
- `PWtype` indicates the type of the pseudowire.
  - `PWtype` is `4` if it is in Ethernet tagged mode.
  - `PWtype` is `5` if it is Ethernet only.

- `l2vpn-id` is the Layer 2 VPN ID for the MS-PW routing instance.
- `SAII-type2` is the type 2 source attachment identifier value.
- `TAII-type2` is the type 2 target attachment identifier value.

Verifying the LDP Database

Purpose

Verify the MS-PW labels received by T-PE1 from S-PE1 and sent from T-PE1 to S-PE1.

Action

From operational mode, run the `show ldp database` command.

```
user@T-PE1> show ldp database
Input label database, 10.255.10.1:0--10.255.2.1:0
  Label     Prefix
      3      10.255.2.1/32
 300112      10.255.3.1/32
 300128      10.255.4.1/32
 299968      10.255.10.1/32
 299904      10.255.13.1/32
 300144      10.255.14.1/32
 300176      FEC129 CtrlWord ETHERNET 000a0064:0000000f 000002bc:000002bc:000002bc
00000320:00000320:00000320

Output label database, 10.255.10.1:0--10.255.2.1:0
  Label     Prefix
 299936      10.255.2.1/32
 300096      10.255.3.1/32
 300112      10.255.4.1/32
      3      10.255.10.1/32
 299920      10.255.13.1/32
 300128      10.255.14.1/32
 300144      FEC129 CtrlWord ETHERNET 000a0064:0000000f 00000320:00000320:00000320
000002bc:000002bc:000002bc

Input label database, 10.255.10.1:0--10.255.13.1:0
  Label     Prefix
 300016      10.255.2.1/32
```

```
 300128      10.255.3.1/32
 300144      10.255.4.1/32
 300080      10.255.10.1/32
      3      10.255.13.1/32
 300160      10.255.14.1/32

Output label database, 10.255.10.1:0--10.255.13.1:0
  Label     Prefix
 299936      10.255.2.1/32
 300096      10.255.3.1/32
 300112      10.255.4.1/32
      3      10.255.10.1/32
 299920      10.255.13.1/32
 300128      10.255.14.1/32
```

Meaning

The labels with `FEC129` prefix are related to the MS-PW.

Checking the MS-PW Connections on T-PE1

Purpose

Make sure that all of the FEC 129 MS-PW connections come up correctly.

Action

From operational mode, run the `show l2vpn connections extensive` command.

```
user@T-PE1> show l2vpn connections extensive
Layer-2 VPN connections:

Legend for connection status (St)
EI -- encapsulation invalid      NC -- interface encapsulation not CCC/TCC/VPLS
EM -- encapsulation mismatch     WE -- interface and instance encaps not same
VC-Dn -- Virtual circuit down    NP -- interface hardware not present
CM -- control-word mismatch      -> -- only outbound connection is up
CN -- circuit not provisioned    <- -- only inbound connection is up
OR -- out of range               Up -- operational
OL -- no outgoing label          Dn -- down
LD -- local site signaled down   CF -- call admission control failure
```

```
RD -- remote site signaled down  SC -- local and remote site ID collision
LN -- local site not designated  LM -- local site ID not minimum designated
RN -- remote site not designated RM -- remote site ID not minimum designated
XX -- unknown connection status  IL -- no incoming label
MM -- MTU mismatch               MI -- Mesh-Group ID not available
BK -- Backup connection          ST -- Standby connection
PF -- Profile parse failure      PB -- Profile busy
RS -- remote site standby        SN -- Static Neighbor
LB -- Local site not best-site   RB -- Remote site not best-site
VM -- VLAN ID mismatch

Legend for interface status
Up -- operational
Dn -- down

Instance: ms-pw
  L2vpn-id: 100:15
    Number of local interfaces: 1
    Number of local interfaces up: 1
    ge-3/1/2.0
  Local source-attachment-id: 800:0.0.3.32:800 (CE1)
    Target-attachment-id      Type  St     Time last up          # Up trans
    700:0.0.2.188:700         rmt   Up     Sep 18 01:10:55 2013           1
      Remote PE: 10.255.2.1, Negotiated control-word: Yes (Null)
      Incoming label: 300048, Outgoing label: 300016
      Negotiated PW status TLV: Yes
      local PW status code: 0x00000000, Neighbor PW status code: 0x00000000
      Local interface: ge-3/1/2.0, Status: Up, Encapsulation: ETHERNET
      Pseudowire Switching Points :
        Local address          Remote address        Status
        10.255.2.1             10.255.3.1            forwarding
        10.255.3.1             10.255.14.1           forwarding
    Connection History:
        Sep 18 01:10:55 2013  status update timer
        Sep 18 01:10:55 2013  PE route changed
        Sep 18 01:10:55 2013  Out lbl Update                    300016
        Sep 18 01:10:55 2013  In lbl Update                     300048
        Sep 18 01:10:55 2013  loc intf up                   ge-3/1/2.0
```

Check the following fields in the output to verify that MS-PW is established between the T-PE devices:

- **Target-attachment-id**—Check if the TAI value is the SAI value of T-PE2.

- **Remote PE**—Check if the T-PE2 loopback address is listed.
- **Negotiated PW status TLV**—Ensure that the value is Yes.
- **Pseudowire Switching Points**—Check if the switching points are listed from S-PE1 to S-PE2 and from S-PE2 to T-PE2.

Meaning

MS-PW is established between T-PE1 and T-PE2 in the forwarding direction.

Checking the MS-PW Connections on S-PE1

Purpose

Make sure that all of the FEC 129 MS-PW connections come up correctly for the mspw routing instance.

Action

From operational mode, run the show l2vpn connections instance *__MSPW__* extensive command.

```
user@S-PE1> show l2vpn connections instance __MSPW__ extensive
Layer-2 VPN connections:

Legend for connection status (St)
EI -- encapsulation invalid      NC -- interface encapsulation not CCC/TCC/VPLS
EM -- encapsulation mismatch     WE -- interface and instance encaps not same
VC-Dn -- Virtual circuit down    NP -- interface hardware not present
CM -- control-word mismatch      -> -- only outbound connection is up
CN -- circuit not provisioned    <- -- only inbound connection is up
OR -- out of range               Up -- operational
OL -- no outgoing label          Dn -- down
LD -- local site signaled down   CF -- call admission control failure
RD -- remote site signaled down  SC -- local and remote site ID collision
LN -- local site not designated  LM -- local site ID not minimum designated
RN -- remote site not designated RM -- remote site ID not minimum designated
XX -- unknown connection status  IL -- no incoming label
MM -- MTU mismatch               MI -- Mesh-Group ID not available
BK -- Backup connection          ST -- Standby connection
PF -- Profile parse failure      PB -- Profile busy
RS -- remote site standby        SN -- Static Neighbor
```

```
LB -- Local site not best-site   RB -- Remote site not best-site
VM -- VLAN ID mismatch

Legend for interface status
Up -- operational
Dn -- down

Instance: __MSPW__
  L2vpn-id: 100:15
  Local source-attachment-id: 700:0.0.2.188:700
    Target-attachment-id      Type  St     Time last up          # Up trans
    800:0.0.3.32:800          rmt   Up     Sep 18 01:17:38 2013           1
      Remote PE: 10.255.10.1, Negotiated control-word: Yes (Null), Encapsulation: ETHERNET
      Incoming label: 300016, Outgoing label: 300048
      Negotiated PW status TLV: Yes
      local PW status code: 0x00000000, Neighbor PW status code: 0x00000000
  Local source-attachment-id: 800:0.0.3.32:800
    Target-attachment-id      Type  St     Time last up          # Up trans
    700:0.0.2.188:700         rmt   Up     Sep 18 01:17:38 2013           1
      Remote PE: 10.255.3.1, Negotiated control-word: Yes (Null), Encapsulation: ETHERNET
      Incoming label: 300000, Outgoing label: 300064
      Negotiated PW status TLV: Yes
      local PW status code: 0x00000000, Neighbor PW status code: 0x00000000
      Pseudowire Switching Points :
        Local address          Remote address        Status
        10.255.3.1             10.255.14.1           forwarding
```

Check the following fields in the output to verify that MS-PW is established between the T-PE devices:

- **Target-attachment-id**—Check if the TAI value is the SAI value of T-PE2.
- **Remote PE**—Check if the T-PE1 and S-PE2 loopback addresses are listed.
- **Negotiated PW status TLV**—Ensure that the value is Yes.
- **Pseudowire Switching Points**—Check if the switching points are listed from S-PE2 to T-PE2.

Meaning

MS-PW is established between T-PE1 and T-PE2 in the forwarding direction.

Checking the MS-PW Connections on S-PE2

Purpose

Make sure that all of the FEC 129 MS-PW connections come up correctly for the mspw routing instance.

Action

From operational mode, run the `show l2vpn connections instance __MSPW__ extensive` command.

```
user@S-PE2> show l2vpn connections instance __MSPW__ extensive
Layer-2 VPN connections:

Legend for connection status (St)
EI -- encapsulation invalid      NC -- interface encapsulation not CCC/TCC/VPLS
EM -- encapsulation mismatch     WE -- interface and instance encaps not same
VC-Dn -- Virtual circuit down    NP -- interface hardware not present
CM -- control-word mismatch      -> -- only outbound connection is up
CN -- circuit not provisioned    <- -- only inbound connection is up
OR -- out of range               Up -- operational
OL -- no outgoing label          Dn -- down
LD -- local site signaled down   CF -- call admission control failure
RD -- remote site signaled down  SC -- local and remote site ID collision
LN -- local site not designated  LM -- local site ID not minimum designated
RN -- remote site not designated RM -- remote site ID not minimum designated
XX -- unknown connection status  IL -- no incoming label
MM -- MTU mismatch               MI -- Mesh-Group ID not available
BK -- Backup connection          ST -- Standby connection
PF -- Profile parse failure      PB -- Profile busy
RS -- remote site standby        SN -- Static Neighbor
LB -- Local site not best-site   RB -- Remote site not best-site
VM -- VLAN ID mismatch

Legend for interface status
Up -- operational
Dn -- down

Instance: __MSPW__
  L2vpn-id: 100:15
  Local source-attachment-id: 700:0.0.2.188:700
    Target-attachment-id      Type  St     Time last up          # Up trans
```

```
    800:0.0.3.32:800          rmt   Up     Sep 18 00:58:55 2013           1
      Remote PE: 10.255.2.1, Negotiated control-word: Yes (Null), Encapsulation: ETHERNET
      Incoming label: 300064, Outgoing label: 300000
      Negotiated PW status TLV: Yes
      local PW status code: 0x00000000, Neighbor PW status code: 0x00000000
      Pseudowire Switching Points :
        Local address          Remote address        Status
        10.255.2.1             10.255.10.1           forwarding
  Local source-attachment-id: 800:0.0.3.32:800
    Target-attachment-id      Type  St     Time last up          # Up trans
    700:0.0.2.188:700         rmt   Up     Sep 18 00:58:55 2013           1
      Remote PE: 10.255.14.1, Negotiated control-word: Yes (Null), Encapsulation: ETHERNET
      Incoming label: 300048, Outgoing label: 300112
      Negotiated PW status TLV: Yes
      local PW status code: 0x00000000, Neighbor PW status code: 0x00000000
```

Check the following fields in the output to verify that MS-PW is established between the T-PE devices:

- **Target-attachment-id**—Check if the TAI value is the SAI value of T-PE1.
- **Remote PE**—Check if the S-PE1 and T-PE2 loopback addresses are listed.
- **Negotiated PW status TLV**—Ensure that the value is Yes.
- **Pseudowire Switching Points**—Check if the switching points are listed from S-PE1 to T-PE1.

Meaning

MS-PW is established between T-PE1 and T-PE2 in the reverse direction.

Checking the MS-PW Connections on T-PE2

Purpose

Make sure that all of the FEC 129 MS-PW connections come up correctly.

Action

From operational mode, run the show l2vpn connections extensive command.

```
user@T-PE2> show l2vpn connections extensive
Layer-2 VPN connections:
```

```
Legend for connection status (St)
EI -- encapsulation invalid      NC -- interface encapsulation not CCC/TCC/VPLS
EM -- encapsulation mismatch     WE -- interface and instance encaps not same
VC-Dn -- Virtual circuit down    NP -- interface hardware not present
CM -- control-word mismatch      -> -- only outbound connection is up
CN -- circuit not provisioned    <- -- only inbound connection is up
OR -- out of range               Up -- operational
OL -- no outgoing label          Dn -- down
LD -- local site signaled down   CF -- call admission control failure
RD -- remote site signaled down  SC -- local and remote site ID collision
LN -- local site not designated  LM -- local site ID not minimum designated
RN -- remote site not designated RM -- remote site ID not minimum designated
XX -- unknown connection status  IL -- no incoming label
MM -- MTU mismatch               MI -- Mesh-Group ID not available
BK -- Backup connection          ST -- Standby connection
PF -- Profile parse failure      PB -- Profile busy
RS -- remote site standby        SN -- Static Neighbor
LB -- Local site not best-site   RB -- Remote site not best-site
VM -- VLAN ID mismatch

Legend for interface status
Up -- operational
Dn -- down

Instance: ms-pw
  L2vpn-id: 100:15
    Number of local interfaces: 1
    Number of local interfaces up: 1
    ge-2/0/0.0
  Local source-attachment-id: 700:0.0.2.188:700 (CE2)
    Target-attachment-id      Type  St     Time last up          # Up trans
    800:0.0.3.32:800          rmt   Up     Sep 18 01:35:21 2013           1
      Remote PE: 10.255.3.1, Negotiated control-word: Yes (Null)
      Incoming label: 300112, Outgoing label: 300048
      Negotiated PW status TLV: Yes
      local PW status code: 0x00000000, Neighbor PW status code: 0x00000000
      Local interface: ge-2/0/0.0, Status: Up, Encapsulation: ETHERNET
      Pseudowire Switching Points :
        Local address          Remote address        Status
        10.255.3.1             10.255.2.1            forwarding
        10.255.2.1             10.255.10.1           forwarding
    Connection History:
        Sep 18 01:35:21 2013  status update timer
```

```
        Sep 18 01:35:21 2013  PE route changed
        Sep 18 01:35:21 2013  Out lbl Update                 300048
        Sep 18 01:35:21 2013  In lbl Update                  300112
        Sep 18 01:35:21 2013  loc intf up                 ge-2/0/0.0
```

Check the following fields in the output to verify that MS-PW is established between the T-PE devices:

- **Target-attachment-id**—Check if the TAI value is the SAI value of T-PE1.
- **Remote PE**—Check if the T-PE1 loopback address is listed.
- **Negotiated PW status TLV**—Ensure that the value is Yes.
- **Pseudowire Switching Points**—Check if the switching points are listed from S-PE2 to S-PE1 and from S-PE1 to T-PE1.

Meaning

MS-PW is established between T-PE1 and T-PE2 in the reverse direction.

Troubleshooting

IN THIS SECTION

- Ping | **1921**
- Bidirectional Forwarding Detection | **1922**
- Traceroute | **1923**

To troubleshoot the MS-PW connection, see:

Ping

Problem

How to check the connectivity between the T-PE devices and between a T-PE device and an intermediary device.

Solution

Verify that T-PE1 can ping T-PE2. The **ping mpls l2vpn fec129** command accepts SAIs and TAIs as integers or IP addresses and also allows you to use the CE-facing interface instead of the other parameters (**instance**, **local-id**, **remote-id**, **remote-pe-address**).

Checking Connectivity Between T-PE1 and T-PE2

```
user@T-PE1> ping mpls l2vpn fec129 instance FEC129-VPWS local-id 800:800:800 remote-pe-address
10.255.14.1 remote-id 700:700:700
!!!!!
--- lsping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
user@T-PE1> ping mpls l2vpn fec129 interface ge-3/1/2
!!!!!
--- lsping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
```

Checking Connectivity Between T-PE1 and S-PE2

```
user@T-PE1> ping mpls l2vpn fec129 interface ge-3/1/2 bottom-label-ttl 2
!!!!!
--- lsping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
```

Bidirectional Forwarding Detection

Problem

How to use BFD to troubleshoot the MS-PW connection from the T-PE device.

Solution

From operational mode, verify the `show bfd session extensive` command output.

```
user@T-PE1> show bfd session extensive

                                             Detect   Transmit
Address                 State     Interface      Time     Interval  Multiplier
198.51.100.7               Up        ge-3/1/0.0     0.900     0.300        3
```

```
 Client FEC129-OAM, TX interval 0.300, RX interval 0.300
 Session up time 03:12:42
 Local diagnostic None, remote diagnostic None
 Remote state Up, version 1
 Replicated
 Session type: VCCV BFD
 Min async interval 0.300, min slow interval 1.000
 Adaptive async TX interval 0.300, RX interval 0.300
 Local min TX interval 0.300, minimum RX interval 0.300, multiplier 3
 Remote min TX interval 0.300, min RX interval 0.300, multiplier 3
 Local discriminator 19, remote discriminator 19
 Echo mode disabled/inactive
 Remote is control-plane independent
 L2vpn-id 100:15, Local-id 800:0.0.3.32:800, Remote-id 700:0.0.2.188:700
  Session ID: 0x103

1 sessions, 1 clients
Cumulative transmit rate 3.3 pps, cumulative receive rate 3.3 pps
```

Traceroute

Problem

How to verify that MS-PW was established.

Solution

From operational mode, verify `traceroute` output.

```
user@T-PE1> traceroute mpls l2vpn fec129 interface interface
Probe options: ttl 64, retries 3, exp 7

  ttl    Label  Protocol    Address          Previous Hop     Probe Status
    1           FEC129      10.255.10.1         (null)           Success

    2           FEC129      10.255.2.1       10.255.10.1         Success

    3           FEC129      10.255.3.1       10.255.2.1          Success

    4           FEC129      10.255.14.1      10.255.2.1          Egress
```

```
Path 1 via ge-3/1/2 destination 198.51.100.0
```

MPLS Stitching For Virtual Machine Connection

IN THIS SECTION

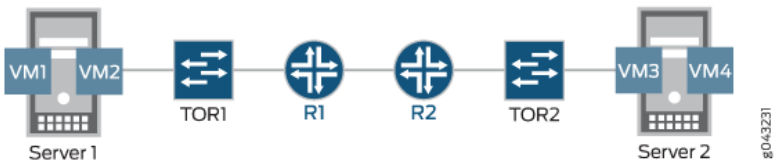
- When Would I Use Stitching? | **1924**
- How Does MPLS Stitching Work? | **1925**
- How Do I Configure Stitching? | **1925**
- Which Switches Support Stitching? | **1926**
- Q&A | **1926**

By using MPLS, the stitching feature of Junos OS provides connectivity between virtual machines that reside either on opposite sides of data center routers or in different data centers. An external controller, programmed in the data-plane, assigns MPLS labels to both virtual machines and servers. Then, the signaled MPLS labels are used between the data center routers, generating static link switched paths (LSPs), resolved over either BGP labeled unicast, RSVP or LDP, to provide the routes dictated by the labels.

When Would I Use Stitching?

There are several ways to connect virtual machines. One option when you have virtual machines on opposite sides of a router (or different data centers) is to use MPLS stitching. A typical topology for using MPLS stitching is shown in Figure 121 on page 1924.

Figure 121: Virtual Machines on Either Side of Routers



The above topology consists of the following MPLS layers: VMs | Servers | ToRs | Router ..... Router | ToRs | Servers | VMs

NOTE: The label on the left is the top of the label stack.

How Does MPLS Stitching Work?

With stitching, the MPLS static allocation of labels demultiplexes incoming traffic onto any device/entity in the next layer in the direction of traffic flow. Essentially, there is a label hierarchy that picks up labels for the correct top-of-rack switch, server, and virtual machine that receives traffic. Static label assignments are done between the top-of-rack switches and the virtual machines.

For example, imagine that traffic is sent from VM1 to VM3 in Figure 121 on page 1924. When traffic exits Server1, its label stack is L1 | L2 | L3 where:

- L1 represents the egress top-of-rack switch ToR1.
- L2 represents the physical server, Server2, towards which the egress-side ToR will forward the traffic.
- L3: represents the virtual machine on Server2 to which the Server2 should deliver the traffic.

Traffic arriving at ToR1 needs to be sent to ToR2. Since ToR1 and ToR2 are not directly connected, traffic must flow from ToR1 to ToR2 using label-switching starting on the outermost (top) label. Stitching has been added to static-LSP functionality to SWAP L1 to a l-BGP label that ToR2 advertises to ToR1. The label stack now must contain another label at the top to enable forwarding of the labeled packets between ToR1 and ToR2. An L-Top label is added if L-BGP is resolved over RSVP/LDP. If static LSP is resolved over L-BGP, then the top label is swapped with the L-BGP label and there is no L-Top label. When the traffic exits ToR1, the stack is: L-top | L-BGP | L2 | L3.

Traffic from ToR1 to ToR2 is then label switched over any signaled LSP.

When traffic arrives at ToR2, the top label is removed with PHP (popped) and the label stack becomes L-BGP | L2 | L3. Since L-BGP is a implicit null label, ToR2 pops the static LSP label L2 that corresponds to the egress server and then forwards the packet to the egress server using the static-LSP configuration on ToR2, which corresponds to a single-hop implicit-NULL LSP.

The outgoing stack becomes L3 and the next-hop is the egress server Server2.

When traffic arrives at the egress server Server2, Server2 pops L3 and delivers the packet to VM3.

How Do I Configure Stitching?

The new keyword `stitch` has been added under `transit` to resolve the remote next-hop. For example, instead of `set protocols mpls static-label-switched-path static-to-ToR2 transit 1000000 next-hop 10.9.82.47`, a top-of-rack switch redirects packets to another top-of-rack switch with `set protocols mpls static-label-switched-path static-to-ToR2 transit 1000000 stitch`. The *show mpls static-lsp* command has been extended

to show the LSP state as 'InProgress' whenever the LSP is waiting for protocol next-hop resolution by resolver.

See the complete example for stitching at Using MPLS Stitching with BGP to Connect Virtual Machines for more information.

Which Switches Support Stitching?

See Feature Explorer for the list of switches that support the MPLS Stitching For Virtual Machine Connections feature.

Q&A

Q: Is link and node protection for the next-hop provided by MPLS stitching?
A: Link and node protection for the next-hop of transit LSP stitched to L-BGP LSP are not needed. That is provided by L-BGP LSP.

TDM Pseudowires Overview

A TDM pseudowire acts as Layer 2 circuit or service for T1 and E1 circuit signals across an MPLS packet-switched network. On ACX Series routers, you configure a TDM *pseudowire* with Structure-Agnostic Time Division Multiplexing (TDM) over Packet (SAToP) on the ACX Series built-in channelized T1 and E1 interfaces. When you configure a TDM pseudowire, the network between the customer edge (CE) routers appears transparent to the CE routers, making it seem that the CE routers are directly connected. With the SAToP configuration on the provider edge (PE) router's T1 and E1 interfaces, the interworking function (IWF) forms a payload (frame) that contains the CE router's T1 and E1 Layer 1 data and control word. This data is transported to the remote PE over the pseudowire. The remote PE removes all the Layer 2 and MPLS headers added in the network cloud and forwards the control word and the Layer 1 data to the remote IWF, which in turn forwards the data to the remote CE router.

Example: TDM Pseudowire Base Configuration

IN THIS SECTION

- Requirements | **1927**
- Overview of a TDM Pseudowire Base Configuration | **1927**
- Configuring an TDM Pseudowire | **1927**

Requirements

The following is a list of the hardware and software requirements for this configuration.

- One ACX Series router
- Junos OS Release 12.2 or later

Overview of a TDM Pseudowire Base Configuration

The configuration shown here is the base configuration of an TDM pseudowire with T1 framing on an ACX Series router. This configuration is for one provider edge router. To complete the TDM pseudowire configuration, you need to repeat this configuration on an other provider edge router in the Multiprotocol Label Switched (MPLS) network.

Configuring an TDM Pseudowire

IN THIS SECTION

- Procedure | **1927**
- Results | **1930**

Procedure

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them in a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level:

```
set chassis fpc 0 pic 0 framing t1
set interfaces ct1-0/0/0 no-partition interface-type t1
set interfaces t1-0/0/0 encapsulation satop
set interfaces t1-0/0/0 unit 0
set interfaces ge-0/2/0 unit 0 family inet address 20.1.1.2/24
set interfaces ge-0/2/0 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 70.1.1.1/32
set protocols rsvp interface ge-0/2/0.0
set protocols mpls no-cspf
```

```
set protocols mpls label-switched-path PE1-to-PE2 to 40.1.1.1
set protocols mpls interface ge-0/2/0.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/2/0.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ldp interface ge-0/2/0.0
set protocols ldp interface lo0.0
set protocols l2circuit neighbor 40.1.1.1 interface t1-0/0/0.0 virtual-circuit-id 1
```

NOTE: To configure a TDM pseudowire with E1 framing, include the `e1` statement at the [`edit chassis fpc 0 pic 0 framing`] hierarchy level instead of the `t1` statement shown in this example.

Step-by-Step Procedure

1. Configure the framing format:

```
[edit]
user@host# edit chassis
[edit chassis]
user@host# set fpc 0 pic 0 framing t1
```

2. Create a T1 interface on a channelized T1 interface (`ct1`) and enable full channelization with the `no-partition` statement. On the logical T1 interface, set the Structure-Agnostic TDM over Packet (SAToP) encapsulation mode.

```
[edit]
user@host# edit interfaces
[edit interfaces]
user@host# set ct1-0/0/0 no-partition interface-type t1
user@host# set t1-0/0/0 encapsulation satop
user@host# set t1-0/0/0 unit 0
```

3. Create a Gigabit Ethernet interface and enable MPLS on that interface. Create the loopback (`lo0`) interface:

```
[edit interfaces]
user@host# set ge-0/2/0 unit 0 family inet address 20.1.1.2/24
user@host# set ge-0/2/0 unit 0 family mpls
user@host# set lo0 unit 0 family inet address 70.1.1.1/32
```

4. Enable the MPLS and RSVP protocols on the MPLS interface—`ge-0/2/0.0`:

```
[edit]
user@host# edit protocols
[edit protocols]
user@host# set rsvp interface ge-0/2/0.0
user@host# set mpls interface ge-0/2/0.0
```

5. Configure LDP. If you configure RSVP for a pseudowire, you must also configure LDP:

```
[edit protocols]
user@host# set ldp interface ge-0/2/0.0
user@host# set ldp interface lo0.0
```

6. Configure a point-to-point label-switched path (LSP) and disable constrained-path LSP computation:

```
[edit protocols]
user@host# set mpls label-switched-path PE1-to-PE2 to 40.1.1.1
user@host# set mpls no-cspf
```

7. Configure OSPF and enable traffic engineering on the MPLS interface—`ge-0/2/0.0`, and on the loopback (`lo0`) interface:

```
[edit protocols]
user@host# set ospf traffic-engineering
user@host# set ospf area 0.0.0.0 interface ge-0/2/0.0
user@host# set ospf area 0.0.0.0 interface lo0.0 passive
```

8. Uniquely identify a Layer 2 circuit for the TDM pseudowire:

```
[edit protocols]
user@host#  set l2circuit neighbor 40.1.1.1 interface t1-0/0/0.0 virtual-circuit-id 1
```

Results

```
[edit]
user@host# show
chassis {
    fpc 0 {
        pic 0 {
            framing t1;
        }
    }
}
interfaces {
    ct1-0/0/0 {
        no-partition interface-type t1;
    }
    t1-0/0/0 {
        encapsulation satop;
        unit 0;
    }
    ge-0/2/0 {
        unit 0 {
            family inet {
                address 20.1.1.2/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 70.1.1.1/32;
            }
        }
    }
}
protocols {
```

```
    rsvp {
        interface ge-0/2/0.0;
    }
    mpls {
        no-cspf;
        label-switched-path PE1-to-PE2 {
            to 40.1.1.1;
        }
        interface ge-0/2/0.0;
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface ge-0/2/0.0;
            interface lo0.0 {
                passive;
            }
        }
    }
    ldp {
        interface ge-0/2/0.0;
        interface lo0.0;
    }
    l2circuit {
        neighbor 40.1.1.1 {
            interface t1-0/0/0.0 {
                virtual-circuit-id 1;
            }
        }
    }
}
```

Configuring Load Balancing for Ethernet Pseudowires

You can configure load balancing for IPv4 traffic over Layer 2 Ethernet pseudowires. You can also configure load balancing for Ethernet pseudowires based on IP information. The option to include IP information in the hash key provides support for Ethernet circuit cross-connect (CCC) connections.

NOTE: This feature is supported only on M120, M320, MX Series, and T Series routers.

To configure load balancing for IPv4 traffic over Layer 2 Ethernet pseudowires, include the `ether-pseudowire` statement at the `[edit forwarding-options hash-key family mpls payload]` hierarchy level:

```
[edit forwarding-options]
hash-key {
    family mpls {
        (label-1 | no-labels);
        payload {
            ether-pseudowire;
        }
    }
}
```

NOTE: You must also configure either the `label-1` or the `no-labels` statement at the `[edit forwarding-options hash-key family mpls]` hierarchy level.

You can also configure load balancing for Ethernet pseudowires based on IP information. This functionality provides support for load balancing for Ethernet cross-circuit connect (CCC) connections. To include IP information in the hash key, include the `ip` statement at the `[edit forwarding-options hash-key family mpls payload]` hierarchy level:

```
[edit forwarding-options]
hash-key {
    family mpls {
        (label-1 | no-labels);
        payload {
            ip;
        }
    }
}
```

NOTE: You must also configure either the `label-1` or `no-labels` statement at the `[edit forwarding-options hash-key family mpls]` hierarchy level.

You can configure load balancing for IPv4 traffic over Ethernet pseudowires to include only Layer 3 IP information in the hash key. To include only Layer 3 IP information, include the `layer-3-only` option at the `[edit forwarding-options family mpls hash-key payload ip]` hierarchy level:

```
[edit forwarding-options]
hash-key {
    family mpls {
        (label-1 | no-labels);
        payload {
            ip {
                layer-3-only;
            }
        }
    }
}
```

NOTE: You must also configure either the `label-1` or `no-labels` statement at the `[edit forwarding-options hash-key family mpls]` hierarchy level.

Configuring Load Balancing Based on MAC Addresses

The hash key mechanism for load-balancing uses Layer 2 media access control (MAC) information such as frame source and destination address. To load-balance traffic based on Layer 2 MAC information, include the `family multiservice` statement at the `[edit forwarding-options hash-key]` hierarchy level:

```
family multiservice {
    destination-mac;
    source-mac;
}
```

To include the destination-address MAC information in the hash key, include the **destination-mac** option. To include the source-address MAC information in the hash key, include the **source-mac** option.

NOTE: Any packets that have the same source and destination address will be sent over the same path.

NOTE: You can configure per-packet load balancing to optimize VPLS traffic flows across multiple paths.

NOTE: Aggregated Ethernet member links will now use the physical MAC address as the source MAC address in 802.3ah OAM packets.

NOTE: ACX Series routers do not support VPLS.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.1X53	Starting in Junos OS Release 14.1X53 and Junos OS Release 16.1, an Ethernet pseudowire is used to carry Ethernet or 802.3 Protocol Data Units (PDUs) over an MPLS network enabling service providers to offer emulated Ethernet services over existing MPLS networks.

RELATED DOCUMENTATION

Basic MPLS Configuration | 46

Pseudowire Headend Termination (PWHT) Configuration

IN THIS SECTION

- PWHT Overview | 1935
- PWHT RLT Configuration Modes | 1936
- Configuring PWHT Active-Backup Mode | 1937
- Configuring PWHT Active-Active Mode without Targeting | 1940
- Configuring PWHT Active-Active Mode with Targeting | 1941

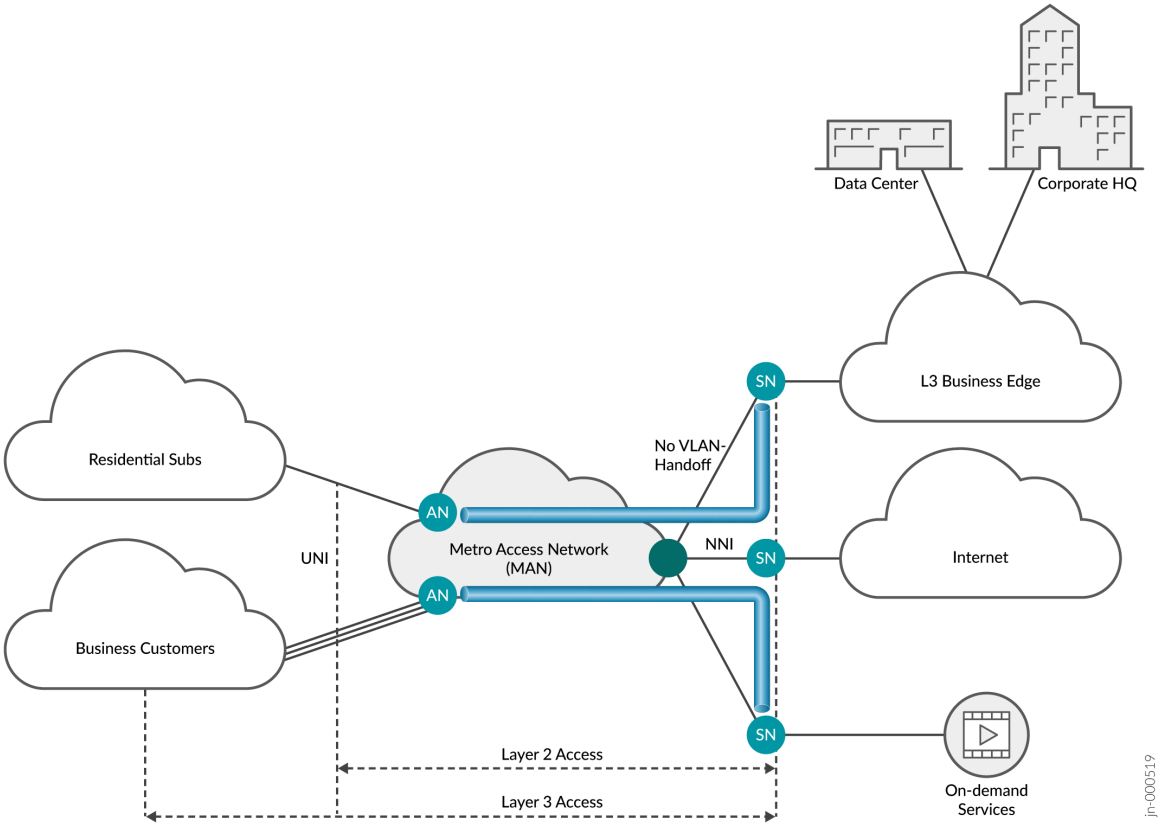
PWHT Overview

IN THIS SECTION

- Benefits of PWHT | **1936**

Pseudowire headend termination (PWHT) connects an L2 circuit from an access node directly to an L3 service at the service node.

Figure 122: PWHT Network



Benefits of PWHT

- PWHT allows you to connect an L2 dedicated circuit directly into an L3 service - such as an L3VPN or EVPN - at the Provider Edge (PE). Traditional pseudowire can only connect at the metro edge, which requires a VPN handoff between the PE router and the metro edge router.

PWHT RLT Configuration Modes

IN THIS SECTION

- Active-Backup Mode | **1936**
- Active-Active Mode without Targeting | **1937**
- Active-Active Mode with Targeting | **1937**

PWHT anchors a pseudowire service interface (ps) onto either a Logical Tunnel (*lt*) or a Redundant Logical Tunnel (RLT). Load balancing requires the ps interface to be anchored on an RLT.

For an RLT to function, you must have at least two *lt* interfaces that are members of the RLT. Each *lt* interface anchors onto a different Packet Forwarding Engine.

When you add more than two logical tunnel interfaces as members of an RLT, all *lt* members default to active mode. You can add up to 32 logical tunnels as members of an RLT.

When you add just two logical tunnels to the RLT, you can configure the members in one of two ways:

- One member in active mode and the other in backup mode
- Both members in active mode (with or without targeting)

We support three possible PWHT RLT configurations:

Active-Backup Mode

Active-Backup Mode is where you have at least one logical tunnel interface (*lt*) in active mode while the other *lt* interface is in backup mode. If the active *lt* fails, the backup *lt* becomes active. You gain redundancy in case of network or hardware failure, but you are not able to use the bandwidth of the backup *lt*.

Active-Active Mode without Targeting

In Active-Active Mode without Targeting, all RLT members are actively forwarding traffic. This allows you to use all of the bandwidth on your PWHT. Traffic is load-balanced across member *lt* interfaces. Flow-aware Transport (FAT) labels must be configured for load balancing to work properly in this mode.

Active-Active Mode with Targeting

Starting in Junos OS Release 23.1R1, we support Active-Active Mode with Targeting for Business Edge use cases.

When you use Active-Active Mode with Targeting, all *lt* interfaces that are members of the anchor RLT are active. Active-Active Mode with Targeting provides several benefits in addition to maximum bandwidth usage. You configure Distribution Lists - each containing at least one *lt* - to target traffic toward specific interfaces. Distribution Lists also guarantee accurate traffic shaping and policing if only one *lt* is in each list.

SEE ALSO

- Connecting Logical Systems Using Logical Tunnel Interfaces
- Redundant Logical Tunnel Overview
- Support for Redundant Logical Tunnel
- *redundant-logical-tunnel*

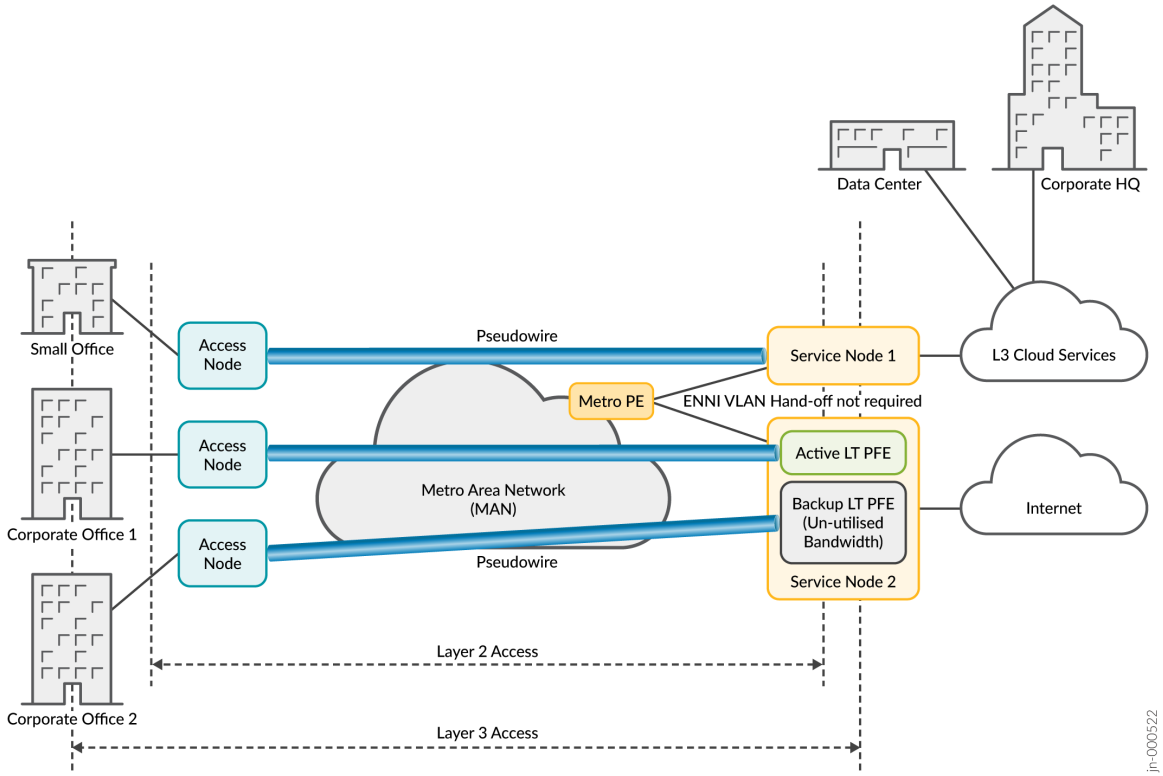
Configuring PWHT Active-Backup Mode

IN THIS SECTION

- Benefits of Active-Backup Mode | **1938**
- Configuration | **1938**
- Operational Verification | **1939**
- Configuration Verification | **1939**

Active-Backup Mode gives you redundancy in case of a network or equipment failure on your PWHT. One logical tunnel interface (*lt*) actively passes traffic through the PWHT, while the other *lt* waits in backup mode. You will not make full use of the bandwidth available to the service interface because one of the *lt* interfaces is in backup mode.

Figure 123: PWHT Network in Active-Backup Mode



Benefits of Active-Backup Mode

- Network Redundancy
- Easy configuration

Configuration

To configure Active-Backup Mode, first configure your pseudowire service interface and redundant logical tunnel (RLT) with two logical tunnel interfaces.

For direction on configuring logical tunnel interfaces and redundant logical tunnels, see Connecting Logical Systems Using Logical Tunnel Interfaces.

For direction on configuring pseudowire interfaces, see "MPLS Pseudowires Configuration" on page 1863.

1. Configure one logical tunnel interface (*lt*) as the active interface.

```
[edit interfaces rlt-name]
user@host# set redundancy-group member-interface lt-interface-name active
```

2. Configure the second *lt* interface as the backup interface.

```
[edit interfaces rlt-name]
user@host# set redundancy-group member-interface lt-interface-name backup
```

Operational Verification

Use the `show interfaces redundancy rlt-number` to verify the status of the *lt* interfaces within the RLT. The example below shows the RLT using the primary *lt* interface to pass traffic with both *lt* interfaces online.

```
[edit]
user@host# run show interfaces redundancy rlt0
Interface   State        Last change     Primary     Secondary     Current status
rlt0        On Primary   00:01:24        lt-0/0/10   lt-1/0/10     both up
```

Configuration Verification

Use the `show interfaces` command to confirm your active-backup configuration. Each member *lt* interface within the RLT is displayed, along with its status as active or backup. To change an *lt* to active or backup status, repeat the one of the steps above as needed.

```
[edit]
user@host# show interfaces rlt0
rlt0 {
    redundancy-group {
        member-interface lt-0/0/10 {
            active;
        }
        member-interface lt-1/0/10 {
            backup;
        }
    }
}
```

See *redundancy-group (Redundant Tunnel)* for additional details.

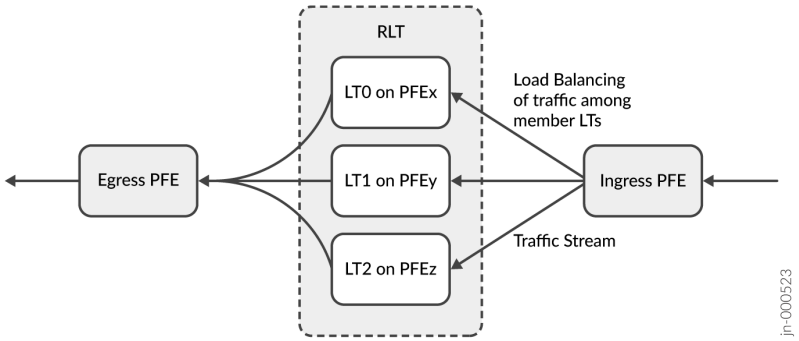
Configuring PWHT Active-Active Mode without Targeting

IN THIS SECTION

- Benefits of Active-Active Mode without Targeting | **1940**
- Configuration | **1940**
- Step-by-step Procedure | **1941**

Active-Active Mode without Targeting places all member logical tunnel interfaces (*lt*) of the anchor redundant logical tunnel (RLT) into an active status. Traffic is automatically managed across the active *lt* interfaces. This maximizes your usage of the reserved bandwidth for a PWHT connection. shows how traffic is directed in this mode.

Figure 124: Active-Active Mode without Targeting



Benefits of Active-Active Mode without Targeting

- Bandwidth is not wasted in standby mode.
- Automatic load-balancing. Traffic is evenly balanced among all active *lt* interfaces.

Configuration

To configure Active-Active Mode without Targeting, first configure your pseudowire service interface and redundant logical tunnel (RLT) with at least two logical tunnel interfaces.

For direction on configuring pseudowire interfaces, see "MPLS Pseudowires Configuration" on page 1863.

For direction on configuring logical tunnel interfaces and redundant logical tunnels, see Connecting Logical Systems Using Logical Tunnel Interfaces.

Step-by-step Procedure

1. Configure all member *lt* interfaces for active mode. If you have more than two member *lt* interfaces, the interfaces are active by default.

```
[edit interfaces rlt-name]
user@host# set redundancy-group member-interface lt-interface-name
```

2. Enable FAT flow labels on your L2 circuit. FAT flow labels allow you to load-balance upstream traffic. If FAT labels are not enabled, traffic will not be load balanced across the active *lt* interfaces.

```
[edit protocols]
user@host# set protocols l2circuit neighbor neighbor-id interface ps-service-interface
virtual-circuit-id circuit-id flow-label-receive flow-label transmit
```

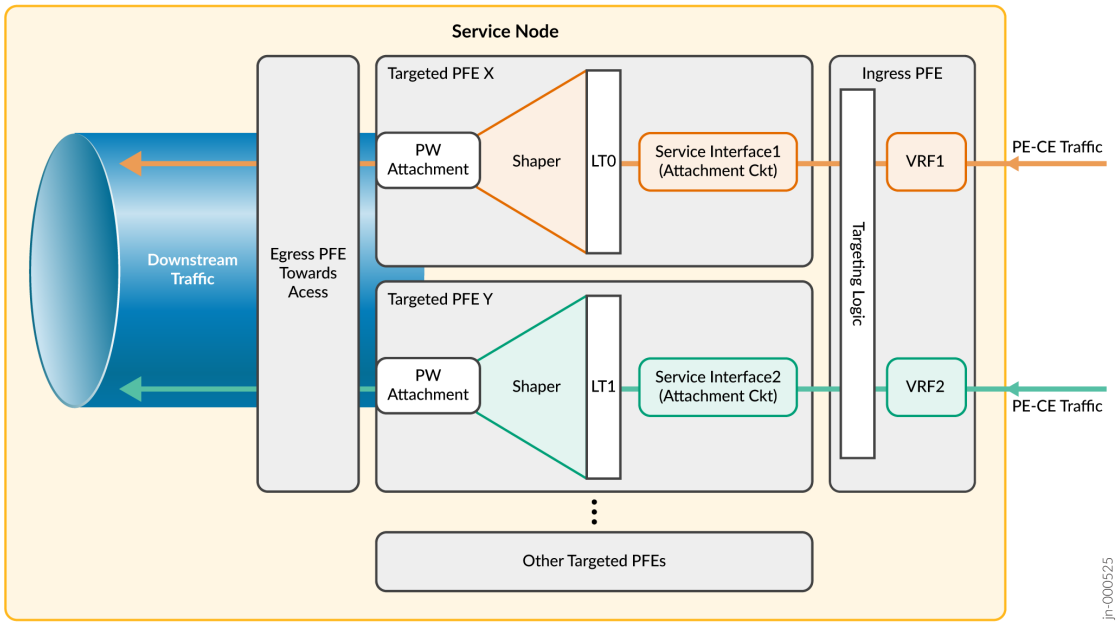
Configuring PWHT Active-Active Mode with Targeting

IN THIS SECTION

- Benefits of Active-Active Mode with Targeting | **1942**
- Configuration | **1942**
- Step-by-step Procedure | **1943**
- Sample Configuration | **1944**

Active-Active Mode with Targeting places all logical tunnel interfaces *(lt)* into an active state. The *lt* interfaces are members of the anchor redundant logical tunnel (RLT). Accurate traffic shaping and policing can be guaranteed in this mode. Figure 125 on page 1942 shows how Active-Active Mode with Targeting works within the RLT.

Figure 125: PWHT Active-Active Mode with Targeting



Benefits of Active-Active Mode with Targeting

- Target traffic to specific interfaces.
- Take full advantage of the available pseudowire bandwidth.
- Traffic Shaping and Policing with the use of Distribution Lists.

Configuration

To configure Active-Active Mode with Targeting, first configure your pseudowire service interface and RLT. You must have at least two *lt* interfaces as members of the RLT.

For information on configuring pseudowire interfaces, see "MPLS Pseudowires Configuration" on page 1863.

For information on configuring logical tunnel interfaces and redundant logical tunnels, see Connecting Logical Systems Using Logical Tunnel Interfaces. .

Step-by-step Procedure

1. Set all member lt interfaces to active mode. If more than two lt interfaces are members of the RLT, then all *lt* interfaces are active by default.

```
[edit]
user@host# edit interfaces rlt-name
[edit interfaces rlt-name]
user@host# set redundancy-group member-interface lt-interface-name
```

2. Set the pseudowire interface (ps) to manual targeting.

```
[edit interfaces]
user@host# set interfaces ps-interface-name targeted-options type manual
```

3. Assign your logical tunnel interfaces (*lt*) to distribution lists (dl).

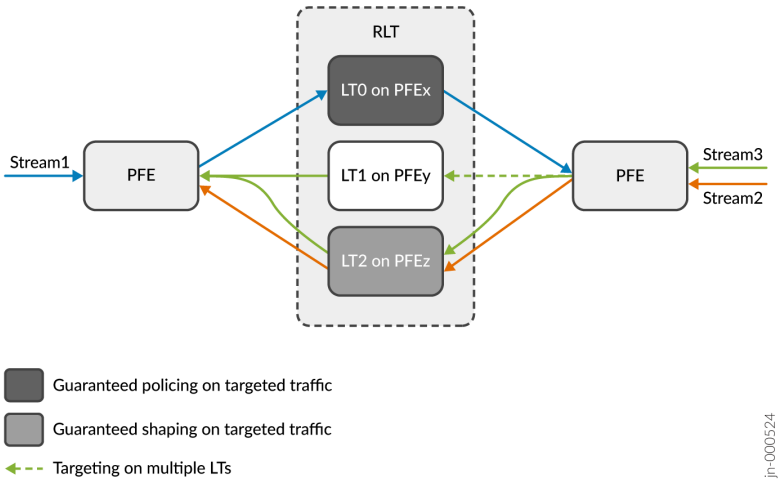
> **NOTE**: For accurate traffic shaping and policing, each distribution list must contain only one *lt* interface. If more than one *lt* interface is included in a distribution list, shaping and policing features are not guaranteed.

```
[edit interfaces]
user@host# set interfaces lt-name logical-tunnel-options distribution-list dl-name
```

4. Assign your distribution lists to a pseudowire interface and configure the distribution lists as primary and backup. Figure 126 on page 1944 shows the traffic flow for guaranteed shaping and policing.

```
[edit interfaces]
user@host# set interfaces ps-interface-name unit unit-number targeted-distribution primary-
list dl-name
user@host# set interfaces ps-interface-name unit unit-number targeted distribution backup-
list dl-name
```

Figure 126: Active-Active Mode with Guaranteed Shaping and Policing



Sample Configuration

From configuration mode, confirm your configuration by entering the `show interfaces` command.

```
[edit]
user@host# show interfaces
rlt0 {
    redundancy-group {
        member-interface lt-0/0/0;
        member-interface lt-1/0/0;
    }
}
lt-0/0/0 {
    logical-tunnel-options {
        distribution-list L0;
    }
}
lt-1/0/0 {
    logical-tunnel-options {
        distribution-list L1;
    }
}
ps1 {
    anchor-point {
        rlt0;
```

```
    }
    targeted-options {
        type manual;
    }
    unit 0 {
        encapsulation ethernet-ccc;
    }
    unit 1 {
        family inet {
            address 192.168.1.2/24;
        }
        targeted-distribution {
            primary-list L0;
            backup-list L1;
         }
    }
}
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
23.4R1	Starting with Junos OS Release 23.4R1, we support replicate mode (CoS) for PWHT All-Active configurations.
23.1R1	Starting with Junos OS Release 23.1R1, PWHT supports Active-Active Mode with Targeting for business edge cases.
23.1R1	Starting with Junos OS Release 23.1R1, support for PWHT Active-Active Mode without Targeting is extended to business edge cases.

RELATED DOCUMENTATION

MPLS Pseudowires Configuration | **1863**

Hierarchical CoS on MPLS Pseudowire Subscriber Interfaces Overview

CoS Two-Level Hierarchical Scheduling on MPLS Pseudowire Subscriber Interfaces

CoS Three-Level Hierarchical Scheduling on MPLS Pseudowire Subscriber Interfaces

Tunnel Interface Configuration on MX Series Routers Overview

Pseudowire over Aggregated Ethernet Overview

SUMMARY

Pseudowire over Aggregated Ethernet (PSoverAE) is used to eliminate the loopback penalty that exists with configurations using PSoLT and PSoRLT .

IN THIS SECTION

- Overview | **1946**
- Benefits of Pseudowire over Aggregated Ethernet | **1946**
- Configuring PSoverAE | **1947**
- CoS for PSoverAE | **1947**
- PSoverAE Targeting Mode | **1947**
- PSoverAE Load Balancing Mode | **1948**

Overview

The Pseudowire over Aggregated Ethernet (PSoverAE) feature enhances network efficiency by anchoring MPLS Pseudowires directly to an Aggregated Ethernet (ae) interface.

A PWHT using PSoverAE eliminates the loopback penalties associated with PSoRLT (pseudowire over redundant logical tunnel)_and PSoLT (pseudowire over logical tunnel) configurations. While PSoLT and PSoRLT use a double-pass operation to process traffic moving downstream, PSoverAE uses a single-pass operation for downstream direction traffic. Single-pass operations reduce resource consumption, improve traffic handling and bandwidth utilization, and simplify network configuration.

PSoverAE supports both targeting and load-balancing modes, allowing for precise traffic control or distributed load management across multiple links.

PSoverAE introduces a simplified CoS hierarchy, which provides accurate traffic shaping and policing

PSoverAE integrates seamlessly with existing layer 2 and layer 3 services.

Benefits of Pseudowire over Aggregated Ethernet

- Eliminates loopback penalties by anchoring Pseudowire services directly to an AE access interface, optimizing bandwidth usage and reducing redundant processing.

- Simplifies network management. Single-pass operation decreases your hardware resource usage and enhances traffic handling.
- Flexibility in traffic management with targeting and load-balancing modes, providing precise control over traffic distribution.
- Precise traffic management with accurate CoS and effective traffic policing.
- Traffic analysis and compliance through integrated port mirroring and monitoring services.
- Enhanced reliability through dynamic path monitoring adjustments with CFM, improving diagnostics and fault detection.
- Seamless integration with existing L2 and L3 service configurations, maintaining compatibility with current network infrastructures.

Configuring PSoverAE

Implementing PSoverAE does not require any new CLI commands. When you anchor the pseudowire, you anchor it to an access AE interface instead of an LT or RLT.

You can use this command to anchor the pseudowire to the Aggregated Ethernet interface:

- `set interfaces ps` *`ps-interface-number`* `anchor-point ae` *`ae-interface-number`*

CoS for PSoverAE

PSoverAE uses a simplified CoS hierarchy. This hierarchy allows for accurate traffic shaping and policing with full CoS support. You use existing commands to configure CoS.

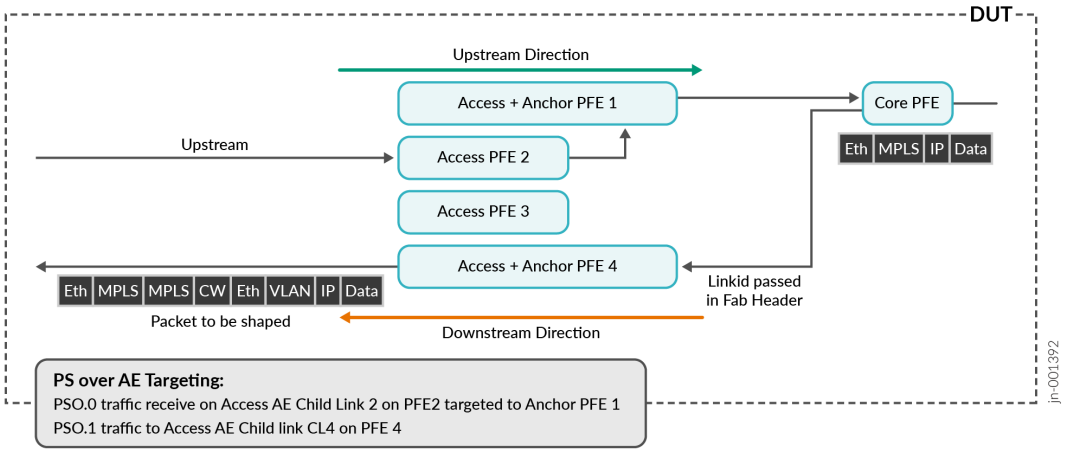
Available CoS features are similar to what is available in PSoRLT/LT configurations. These features are configured on the anchor interface - in this case, the ae IFD.

PSoverAE Targeting Mode

In targeted mode, PS over aggregated Ethernet sends traffic through specified child links in the aggregated Ethernet bundle. Targeted mode relies on the use of primary and backup distribution lists. To configure Targeting Mode, you use the `No Link Title` command to specify which primary and backup distribution lists the interface will use for traffic distribution.

Figure 1 on page 1948 shows an example of targeting mode.

Figure 127: PSoverAE Targeting Mode

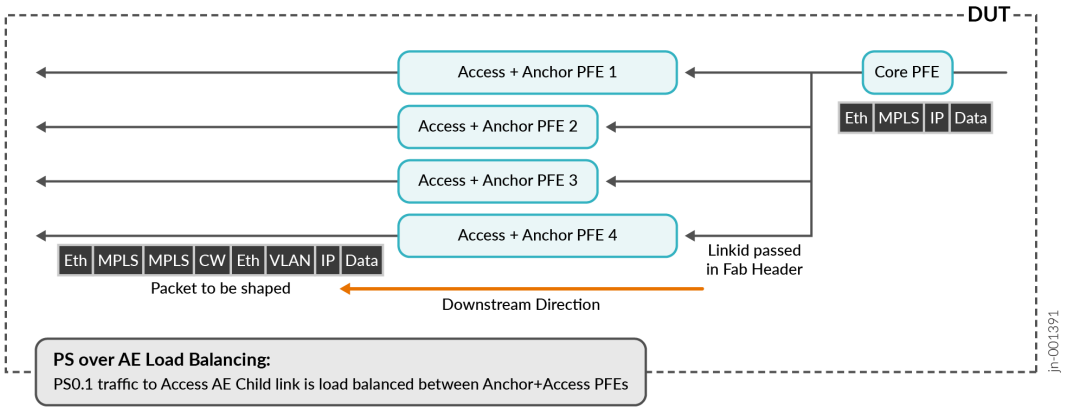


PSoverAE Load Balancing Mode

Pseudowire over Aggregated Ethernet supports load balancing. In load balancing mode, traffic proceeds through the Aggregated Ethernet bundle and balanced across the child links. Load Balancing mode uses the `No Link Title` command.

"Figure 1" on page 1948 shows load balancing for Pseudowire over aggregated Ethernet.

Figure 128: PSoverAE Load Balancing Mode



RELATED DOCUMENTATION

No Link Title

No Link Title

MPLS Pseudowires Configuration | **1863**

Pseudowire Headend Termination (PWHT) Configuration | **1934**

No Link Title

No Link Title

No Link Title

CHAPTER 15

Class-of-Service (CoS) for MPLS

IN THIS CHAPTER

- MPLS Class-of-Service Configuration | **1950**

MPLS Class-of-Service Configuration

IN THIS SECTION

- Configure CoS for MPLS LSPs | **1951**
- Configuring MPLS Rewrite Rules | **1954**
- Configuring CoS Bits for an MPLS Network | **1956**
- Configuring CoS on an MPLS Provider Edge Switch Using IP Over MPLS | **1957**
- Configuring CoS on an MPLS Provider Edge Switch Using Circuit Cross-Connect | **1959**
- Configuring CoS on Provider Switches of an MPLS Network | **1961**
- Understanding Using CoS with MPLS Networks on EX Series Switches | **1962**
- Example: Combining CoS with MPLS on EX Series Switches | **1966**
- Understanding CoS MPLS EXP Classifiers and Rewrite Rules | **1984**
- Configuring Rewrite Rules for MPLS EXP Classifiers | **1988**
- Configuring CoS Bits for an MPLS Network | **1990**
- Configuring a Global MPLS EXP Classifier | **1990**

Configure CoS for MPLS LSPs

IN THIS SECTION

- CoS for MPLS Overview | **1951**
- Configure the MPLS CoS Values | **1951**
- Rewrite IEEE 802.1p Packet Headers with the MPLS CoS Value | **1954**

The following sections provide an overview of MPLS CoS and describe how to configure the MPLS CoS value:

CoS for MPLS Overview

When IP traffic enters an LSP tunnel, the ingress router marks all packets with a CoS value, which is used to place the traffic into a transmission priority queue. Junos encodes the CoS value as part of the MPLS header and remains in the packets until the MPLS header is removed when the packets exit the egress router. The routers within the LSP use the CoS value set at the ingress. The CoS value is encoded by means of the CoS bits (also known as the EXP bits). For more information, see MPLS Label Allocation.

MPLS CoS works with the router's general CoS functionality. If you do not configure any CoS features, the default general CoS settings are used. For MPLS CoS, you might want to configure WRR to prioritize how the transmit queues are serviced. Configure congestion avoidance using RED.

Configure the MPLS CoS Values

When traffic enters an LSP tunnel, the CoS value in the MPLS header is set in one of three ways:

- The output queue number into which the packet was buffered and the PLP are written into the MPLS header and are used as the packet's CoS value. This is default behavior and requires no configuration. *Default MPLS EXP Classifier* explains the default MPLS CoS values and how these values are treated.
- You set a fixed CoS value on all packets entering the LSP tunnel. A fixed CoS value means that all packets entering the LSP receive the same class of service.
- You set an MPLS EXP rewrite rule to override the default behavior.

To set a fixed CoS value on all packets entering the LSP, include the `class-of-service` statement:

```
class-of-service cos-value;
```

You can include this statement at the following hierarchy levels:

- [edit protocols mpls]
- [edit protocols mpls label-switched-path *path-name*]
- [edit protocols mpls label-switched-path *path-name* primary *path-name*]
- [edit protocols mpls label-switched-path *path-name* secondary *path-name*]
- [edit protocols rsvp interface *interface-name* link-protection]
- [edit protocols rsvp interface *interface-name* link-protection bypass *destination*]
- [edit logical-systems *logical-system-name* protocols mpls]
- [edit logical-systems *logical-system-name* protocols mpls label-switched-path *path-name*]
- [edit logical-systems *logical-system-name* protocols mpls label-switched-path *path-name* primary *path-name*]
- [edit logical-systems *logical-system-name* protocols mpls label-switched-path *path-name* secondary *path-name*]
- [edit logical-systems *logical-system-name* protocols rsvp interface *interface-name* link-protection]
- [edit logical-systems *logical-system-name* protocols rsvp interface *interface-name* link-protection bypass *destination*]

The CoS value set using the class-of-service statement at the [edit protocols mpls] hierarchy level supersedes the CoS value set at the [edit class-of-service] hierarchy level for an interface. Effectively, the CoS value configured for an LSP overrides the CoS value set for an interface.

The class-of-service statement at the [edit protocols mpls label-switched-path] hierarchy level assigns an initial EXP value for the MPLS shim header of packets in the LSP. This value is initialized at the ingress routing device only and overrides the rewrite configuration established for that forwarding class. However, the CoS processing (weighted round robin [WRR] and RED) of packets entering the ingress routing device is not changed by the class-of-service statement on an MPLS LSP. Classification is still based on the behavior aggregate (BA) classifier at the [edit class-of-service] hierarchy level or the multifield classifier at the [edit firewall] hierarchy level.

BEST PRACTICE: We recommend configuring all routing devices along the LSP to have the same input classifier for EXP, and, if a rewrite rule is configured, all routing devices should have the same rewrite configuration. Otherwise, traffic at the next LSR might be classified into a different forwarding class, resulting in a different EXP value being written to the EXP header.

The CoS value can be a decimal number from 0 through 7. This number corresponds to a 3-bit binary number. The high-order 2 bits of the CoS value select which transmit queue to use on the outbound interface card.

The low-order bit of the CoS value is treated as the PLP bit and is used to select the RED drop profile to use on the output queue. If the low-order bit is 0, the non-PLP drop profile is used, and if the low-order bit is 1, the PLP drop profile is used. It is generally expected that RED will more aggressively drop packets that have the PLP bit set. For more information about RED and drop profiles, see *RED Drop Profiles for Congestion Management*.

NOTE: Configuring the PLP drop profile to drop packets more aggressively (for example, setting the CoS value from 6 to 7) decreases the likelihood of traffic getting through.

Table 40 on page 1953 summarizes how MPLS CoS values correspond to the transmit queue and PLP bit. Note that in MPLS, the mapping between the CoS bit value and the output queue is hard-coded. You cannot configure the mapping for MPLS; you can configure it only for IPv4 traffic flows, as described in *Understanding How Forwarding Classes Assign Classes to Output Queues*.

Table 40: MPLS CoS Values

MPLS CoS Value	Bits	Transmit Queue	PLP Bit
0	000	0	Not set
1	001	0	Set
2	010	1	Not set
3	011	1	Set
4	100	2	Not set
5	101	2	Set
6	110	3	Not set
7	111	3	Set

Because the CoS value is part of the MPLS header, the value is associated with the packets only as they travel through the LSP tunnel. The value is not copied back to the IP header when the packets exit from the LSP tunnel.

To configure class of service (CoS) for Multiprotocol Label Switching (MPLS) packets in a label-switched path (LSP):

1. Specify the CoS value

   If you do not specify a CoS value, the IP precedence bits from the packet's IP header are used as the packet's CoS value.

Rewrite IEEE 802.1p Packet Headers with the MPLS CoS Value

You can rewrite both MPLS and IEEE 802.1p CoS values to a configured value. Rewriting these values allows you to pass the configured value to the L2 VLAN path. To rewrite both the MPLS and IEEE 802.1p CoS values, you must include the EXP and IEEE 802.1p rewrite rules in the CoS interface configuration. The EXP rewrite table is applied when you configure the IEEE 802.1p and EXP rewrite rules.

For information about how to configure the EXP and IEEE 802.1p rewrite rules, see *Rewriting Packet Headers to Ensure Forwarding Behavior*.

Configuring MPLS Rewrite Rules

IN THIS SECTION

- Rewriting the EXP Bits of All Three Labels of an Outgoing Packet | **1954**
- Rewriting MPLS and IPv4 Packet Headers | **1955**

You can apply a number of different rewrite rules to MPLS packets.

For more information about how to configure statements at the `[edit class-of-service]` hierarchy level, see the Junos OS Class of Service User Guide for Routing Devices.

The following sections describe how you can apply rewrite rules to MPLS packets:

Rewriting the EXP Bits of All Three Labels of an Outgoing Packet

In interprovider, carrier-of-carrier, and complex traffic engineering scenarios, it is sometimes necessary to push three labels on the next hop.

By default, on M Series routers except the M320, the top MPLS EXP label of an outgoing packet is not rewritten when you configure swap-push-push and triple-push operations. You can rewrite the EXP bits of all three labels of an outgoing packet, thereby maintaining the class of service (CoS) of an incoming MPLS or non-MPLS packet.

To push three labels on incoming MPLS packets, include the `exp-swap-push-push default` statement at the `[edit class-of-service interfaces` *`interface-name`* `unit` *`logical-unit-number`* `rewrite-rules]` hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number rewrite-rules]
exp-swap-push-push default;
```

To push three labels on incoming non-MPLS packets, include the `exp-push-push-push default` statement at the `[edit class-of-service interfaces` *`interface-name`* `unit` *`logical-unit-number`* `rewrite-rules]` hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number rewrite-rules]
exp-push-push-push default;
```

For more information about how to configure statements at the `[edit class-of-service]` hierarchy level, see the Junos OS Class of Service User Guide for Routing Devices.

Rewriting MPLS and IPv4 Packet Headers

You can apply a rewrite rule to MPLS and IPv4 packet headers simultaneously. This allows you to initialize MPLS EXP and IP precedence bits at LSP ingress. You can configure different rewrite rules depending on whether the traffic is VPN or non-VPN.

To rewrite MPLS and IPv4 packet headers, include the `protocol` statement at the `[edit class-of-service interfaces` *`interface-name`* `unit` *`logical-unit-number`* `rewrite-rules exp` *`rewrite-rule-name`*`]` hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number rewrite-rules exp
rewrite-rule-name]
protocol types;
```

Use the `protocol` statement to specify the types of MPLS packets and packet headers to which to apply the rewrite rule. The MPLS packet can be a standard MPLS packet or an MPLS packet with an IPv4 payload. Specify the type of MPLS packet by using the following options:

- `mpls-any`—Applies the rewrite rule to MPLS packets and writes the code point value to MPLS headers.
- `mpls-inet-both`—Applies the rewrite rule to VPN MPLS packets with IPv4 payloads. Writes the code point value to the MPLS and IPv4 headers in T Series (except T4000 routers) and M320 routers. On

M Series routers, except the M320, the `mpls-inet-both` option causes all ingress MPLS LSP packets with IPv4 payloads to be initialized with 000 code points for IP precedence and MPLS EXP values.

- `mpls-inet-both-non-vpn`—Applies the rewrite rule to any non-VPN MPLS packets with IPv4 payloads. Writes the code point value to the MPLS and IPv4 headers in T Series and M320 routers. On M Series routers, except the M320, the `mpls-inet-both-non-vpn` option causes all ingress MPLS LSP packets with IPv4 payloads to be initialized with 000 code points for IP precedence and MPLS EXP values.

For a detailed example on how to configure rewrite rules for MPLS and IPv4 packets and for more information about how to configure class of service, see the Junos OS Class of Service User Guide for Routing Devices.

Configuring CoS Bits for an MPLS Network

When traffic enters a labeled-switch path (LSP) tunnel, the CoS bits in the MPLS header are set in one of two ways:

- The number of the output queue into which the packet was buffered and the packet loss priority (PLP) bit are written into the MPLS header and are used as the packet's CoS value. This behavior is the default, and no configuration is required. The *Junos OS Class of Service Configuration Guide* explains the IP CoS values, and summarizes how the CoS bits are treated.
- You set a fixed CoS value on all packets entering the LSP tunnel. A fixed CoS value means that all packets entering the LSP receive the same class of service.

To set a fixed CoS value on all packets entering the LSP:

1. Specify a class of service value for the LSP:

NOTE: The CoS value set using the `class-of-service` statement at the `[edit protocols mpls]` hierarchy level supersedes the CoS value set at the `[edit class-of-service]` hierarchy level for an interface. Effectively, the CoS value configured for an LSP overrides the CoS value set for an interface.

```
[edit protocols mpls]
user@switch# set class-of-service cos-value
```

Configuring CoS on an MPLS Provider Edge Switch Using IP Over MPLS

IN THIS SECTION

- Configuring CoS | **1957**
- Configuring an LSP Policer | **1958**

You can use class of service (CoS) within MPLS networks to prioritize certain types of traffic during periods of congestion. This topic describes configuring CoS components on a provider edge (PE) switch that is using IP Over MPLS.

This task describes how to create a custom DSCP classifier and a custom EXP rewrite rule on the ingress PE switch. It includes configuring a policer firewall filter and applying it to the customer-edge interface of the ingress PE switch. The policer firewall filter ensures that the amount of traffic forwarded through the MPLS tunnel never exceeds the requested bandwidth allocation.

Before you begin, configure the basic components for an MPLS network:

- Configure two PE switches. See "Basic MPLS Configuration" on page 46.
- Configure one or more provider switches.

Configuring CoS

To configure CoS on a provider edge switch:

1. Import the default DSCP classifier classes to the custom DSCP classifier that you are creating:

```
[edit class-of-service]
user@switch# set classifiers dscp classifier-name import default
```

2. Add a forwarding class to this custom DSCP classifier and specify a loss priority and code point:

```
[edit class-of-service]
user@switch# set classifiers dscp classifier-name forwarding-class forwarding-class loss-
priority loss-priority code-points code-point
```

3. Specify the values for the custom EXP rewrite rule, e1:

```
[edit class-of-service]
user@switch# set rewrite-rules exp e1 forwarding-class forwarding-class loss-priority loss-priority code-points code-point
```

Configuring an LSP Policer

To configure an LSP policer:

1. Specify the number of bits per second permitted, on average, for the firewall policer, which will later be applied to the customer-edge-interface:

```
[edit firewall]
user@switch# set policer mypolicer if-exceeding bandwidth-limit 500m
```

2. Specify the maximum size permitted for bursts of data that exceed the given bandwidth limit for this policer:

```
[edit firewall policer]
user@switch# set mypolicer if-exceeding burst-size-limit 33553920
```

3. Discard traffic that exceeds the rate limits for this policer:

```
[edit firewall policer]
user@switch# set mypolicer then discard
```

4. To reference the policer, configure a filter term that includes the policer action:

```
[edit firewall]
user@switch# set family inet filter myfilter term t1 then policer mypolicer
```

5. Apply the filter to the customer-edge interface:

```
[edit interfaces]
user@switch# set ge-2/0/3 unit 0 family inet address 192.168.121.1/16 policing filter myfilter
```

NOTE: You can also configure schedulers and shapers as needed. See *Defining CoS Schedulers and Scheduler Maps (CLI Procedure)*.

Configuring CoS on an MPLS Provider Edge Switch Using Circuit Cross-Connect

IN THIS SECTION

- Configuring CoS | **1959**
- Configuring an LSP Policer | **1960**

You can use class of service (CoS) within MPLS networks to prioritize certain types of traffic during periods of congestion. This topic describes configuring CoS components on a provider edge (PE) switch that is using MPLS over circuit-cross connect (CCC).

NOTE: If you are using MPLS over CCC, you can use only one DSCP or IP precedence classifier and only one IEEE 802.1p classifier on the CCC interfaces.

This procedure is for creating a custom DSCP classifier and a custom EXP rewrite rule on the ingress PE. It also includes enabling a policer on the label-switched path (LSP) of the ingress PE to ensure that the amount of traffic forwarded through the LSP never exceeds the requested bandwidth allocation.

This topic includes:

Configuring CoS

To configure CoS on a provider edge switch:

1. Import the default DSCP classifier classes to the custom DSCP classifier that you are creating:

```
[edit class-of-service]
user@switch# set classifiers dscp classifier-nameimport default
```

2. Add the expedited-forwarding class to this custom DSCP classifier, specifying a loss priority and code point:

```
[edit class-of-service]
user@switch# set classifiers dscp classifier-name forwarding-class forwarding-class loss-
priority loss-priority code-points code-point
```

3. Specify the values for the custom EXP rewrite rule, `e1`:

```
                              [edit class-of-service]
user@switch# set rewrite-rules exp e1 forwarding-class forwarding-class loss-priority loss-
priority code-point code-point
```

4. Bind the DSCP classifier to the CCC interface:

```
[edit ]
user@switch# set class-of-service interfaces interface unit unit classifier classifier-
name
```

Configuring an LSP Policer

To configure an LSP policer:

1. Specify the number of bits per second permitted, on average, for the policer, which will later be applied to the LSP:

```
[edit firewall]
set policer mypolicer if-exceeding bandwidth-limit 500m
```

2. Specify the maximum size permitted for bursts of data that exceed the given bandwidth limit for this policer:

```
[edit firewall policer]
set mypolicer if-exceeding burst-size-limit 33553920
```

3. Discard traffic that exceeds the rate limits for this policer:

```
[edit firewall policer]
set mypolicer then discard
```

4. To reference the policer, configure a filter term that includes the policer action:

```
                              [edit firewall]
user@switch# set family any filter myfilter term t1 then policer
mypolicer
```

5. Apply the filter to the LSP:

```
[edit protocols mpls]
set label-switched-path lsp_to_pe2_ge1 policing filter myfilter
```

NOTE: You can also configure schedulers and shapers as needed. See *Defining CoS Schedulers and Scheduler Maps (CLI Procedure)*.

Configuring CoS on Provider Switches of an MPLS Network

You can add class-of-service (CoS) components to your MPLS networks on EX Series switches to achieve end-to-end Differentiated Services to match your specific business requirements. The configuration of CoS components on the provider switches is the same regardless of whether the provider edge (PE) switches are using MPLS over CCC or IP over MPLS.

This task shows how to configure a custom EXP classifier and custom EXP rewrite rule on the provider switch.

1. Import the default EXP classifier classes to the custom EXP classifier that you are creating:

```
[edit class-of-service]
user@switch# set classifiers exp exp1 import default
```

2. Add the expedited-forwarding class to this custom EXP classifier, specifying a loss priority and code point:

```
[edit class-of-service]
user@switch# set classifiers exp exp1 forwarding-class expedited-forwarding loss-priority low
code-points 010
```

3. Specify the values for the custom EXP rewrite rule, **e1**:

```
[edit class-of-service]
user@switch# set rewrite-rules exp e1 forwarding-class expedited-forwarding loss-priority low
code-point 111
```

NOTE: You can also configure schedulers and shapers as needed. See *Defining CoS Schedulers and Scheduler Maps (CLI Procedure)*.

Understanding Using CoS with MPLS Networks on EX Series Switches

IN THIS SECTION

- EXP Classifiers and EXP rewrite Rules | **1963**
- Guidelines for Using CoS Classifiers on CCCs | **1963**
- Using CoS Classifiers with IP over MPLS | **1964**
- Setting CoS Bits in an MPLS Header | **1964**
- EXP Rewrite Rules | **1965**
- Policer | **1966**
- Schedulers | **1966**

You can use *class of service* (CoS) within MPLS networks to prioritize certain types of traffic during periods of congestion. See EX Series Switch Software Features Overview for a complete list of the Junos OS MPLS features that are supported on specific EX Series switches.

Juniper Networks EX Series Ethernet Switches support Differentiated Service Code Point (DSCP) or IP precedence and IEEE 802.1p CoS classifiers on the customer-edge interfaces of the ingress provider edge (PE) switch. DSCP or IP precedence classifiers are used for Layer 3 packets. IEEE 802.1p is used for Layer 2 packets.

When a packet enters a customer-edge interface of the ingress PE switch, the switch associates the packet with a particular CoS servicing level before putting the packet onto the label-switched path (LSP). The switches within the LSP utilize the CoS value set at the ingress PE switch. The CoS value that was embedded in the classifier is translated and encoded in the MPLS header by means of the EXP or experimental bits. EX Series switches enable a default EXP classifier and a default EXP rewrite rule. For more information about EXP classifiers and EXP *rewrite rules*, see EXP Classifiers and EXP rewrite Rules.

This topic includes:

EXP Classifiers and EXP rewrite Rules

EX Series switches enable a default EXP classifier and a default EXP rewrite rule. You can configure a custom EXP classifier and a custom EXP rewrite rule if you prefer. However, the switch supports only one type of EXP classifier (default or custom) and only one EXP rewrite rule (default or custom).

You do not bind the EXP classifier or the EXP rewrite rule to individual interfaces. The switch automatically and implicitly applies the default or the custom EXP classifier and the default or the custom EXP rewrite rule to the appropriate MPLS-enabled interfaces. Because rewrite rules affect only egress interfaces, the switch applies the EXP rewrite rule only to those MPLS interfaces that are transmitting MPLS packets (not to the MPLS interfaces that are receiving the packets).

After traversing the MPLS tunnel, the traffic flows out from the egress provider edge (PE) switch. Before the traffic leaves the egress interface, the egress PE switch copies the EXP bits from the MPLS header to the most significant bits in the original IP packet--- that is, to the IP precedence bits.

Guidelines for Using CoS Classifiers on CCCs

When you are configuring CoS for MPLS over circuit cross-connect (CCC), there are some additional guidelines, as follows:

- You *must* explicitly bind a CoS classifier to the CCC interface on the ingress PE switch.
- You *must* use the same DSCP, IP precedence, or IEEE 802.1p classifier on CCC interfaces. However, if the CCC interfaces are on the same switch, you cannot configure both a DSCP and an IP precedence classifier on these interfaces. Thus, if you configure one CCC interface to use a DSCP classifier DSCP1, you cannot configure another CCC interface to use another DSCP classifier DSCP2. All the CCC interfaces on the switch must use the same DSCP (or IP precedence) classifier and the same IEEE 802.1p classifier.
- You *cannot* configure one CCC interface to use a DSCP classifier and another CCC interface to use an IP precedence classifier, because these classifier types overlap.
- You *can* configure one CCC interface to use a DSCP classifier and another CCC interface to use IEEE 802.1p classifier.

- You *can* configure one CCC interface to use both a DSCP and an IEEE 802.1p classifier. If you configure a CCC interface to use both these classifiers, the DSCP classifier is used for routing Layer 3 packets and the IEEE 802.1p classifier is used for routing Layer 2 packets.
- You *can* configure one CCC interface to use both an IP precedence and an IEEE 802.1p classifier. If you configure a CCC interface to use both these classifiers, the IP precedence classifier is used for routing Layer 3 packets and the IEEE 802.1p classifier is used for routing Layer 2 packets.

You can define multiple DSCP, IP precedence, and IEEE 802.1p classifiers for the non-CCC interfaces on a switch.

Using CoS Classifiers with IP over MPLS

When you are configuring CoS for IP over MPLS, the customer-edge interface uses the CoS configuration for the switch as the default. You do not have to bind a classifier to the customer-edge interface in this case. There are no restrictions on using multiple DSCP, IP precedence, and IEEE 802.1p classifiers on the same switch.

- You can modify the CoS classifier for a particular interface, but it is not required.
- You can configure a DSCP classifier, DSCP1 on the first interface, another DSCP classifier, DSCP2 on the second interface, and an IP precedence classifier on a third interface, and so forth.

Setting CoS Bits in an MPLS Header

When traffic enters an LSP tunnel, the CoS bits in the MPLS header are set in one of two ways:

- The number of the output queue into which the packet was buffered and the packet loss priority (PLP) bit are written into the MPLS header and are used as the packet's CoS value. This behavior is the default, and no configuration is required. The Junos OS Class of Service User Guide for Routing Devices explains the IP CoS values, and summarizes how the CoS bits are treated.
- You set a fixed CoS value on all packets entering the LSP tunnel. A fixed CoS value means that all packets entering the LSP receive the same class of service.

The CoS value can be a decimal number from 0 through 7. This number corresponds to a 3-bit binary number. The high-order 2 bits of the CoS value select which transmit queue to use on the outbound interface card.

The low-order bit of the CoS value is treated as the PLP bit and is used to select the RED drop profile to use on the output queue. If the low-order bit is 0, the non-PLP drop profile is used, and if the low-order bit is 1, the PLP drop profile is used. It is generally expected that random early detection (RED) will more aggressively drop packets that have the PLP bit set. For more information about RED and drop profiles, see the Junos OS Class of Service User Guide for Routing Devices.

NOTE: Configuring the PLP drop profile to drop packets more aggressively (for example, setting the CoS value from 6 to 7) decreases the likelihood of traffic getting through.

Table 41 on page 1965 summarizes how MPLS CoS values correspond to the transmit queue and PLP bit. Note that in MPLS, the mapping between the CoS bit value and the output queue is hard-coded. You cannot configure the mapping for MPLS; you can configure it only for IPv4 traffic flows, as described in the Junos OS Class of Service User Guide for Routing Devices.

Table 41: MPLS CoS Values

MPLS CoS Value	Bits	Transmit Queue	PLP Bit
0	000	0	Not set
1	001	0	Set
2	010	1	Not set
3	011	1	Set
4	100	2	Not set
5	101	2	Set
6	110	3	Not set
7	111	3	Set

Because the CoS value is part of the MPLS header, the value is associated with the packets only while they travel through the LSP tunnel. The value is not copied back to the IP header when the packets exit from the LSP tunnel.

EXP Rewrite Rules

When traffic passes from the customer-edge interface to an MPLS interface, the DSCP, IP precedence, or IEEE 802.1p CoS classifier is translated into the EXP bits within the MPLS header. You cannot disable the default EXP rewrite rule, but you can configure your own custom EXP classifier and a custom EXP

rewrite rule. You cannot bind the EXP classifier to individual MPLS interfaces; the switch applies it globally to all the MPLS-enabled interfaces on the switch.

Only one EXP rewrite rule (either default or custom) is supported on a switch. The switch applies it to all the egress interfaces on which MPLS is enabled.

Policer

Policing helps to ensure that the amount of traffic forwarded through an LSP never exceeds the requested bandwidth allocation. During periods of congestion (when the total rate of queuing packets exceeds the rate of transmission), any new packets being sent to an interface can be dropped because there is no place to store them. You can configure a policer on the ingress PE switch to prevent this:

- If you are using MPLS over CCC, you bind the policer to the LSP. You cannot bind a policer to a CCC interface.
- If you are using IP over MPLS, you bind the policer to the **inet-family** customer-edge interface. You cannot bind a policer to the LSP when you are using IP over MPLS.

Schedulers

The schedulers for using CoS with MPLS are the same as for the other CoS configurations on EX Series switches. Default schedulers are provided for best-effort and network-control forwarding classes. If you are using assured-forwarding, expedited-forwarding, or any custom forwarding class, we recommend that you configure a scheduler to support that forwarding class. See *Understanding CoS Schedulers*.

Example: Combining CoS with MPLS on EX Series Switches

IN THIS SECTION

- Requirements | **1967**
- Overview and Topology | **1967**
- Configuring the Local PE Switch | **1971**
- Configuring the Remote PE Switch | **1974**
- Configuring the Provider Switch | **1976**
- Verification | **1978**

You can use class of service (CoS) within MPLS networks to prioritize certain types of traffic during periods of congestion. The CoS value is included within the MPLS label, which is passed through the network, enabling end-to-end CoS across the network.

MPLS services are often used to ensure better performance for low-latency applications such as VoIP and other business-critical functions. These applications place specific demands on a network for successful transmission. CoS gives you the ability to control the mix of bandwidth, delay, jitter, and packet loss while taking advantage of the MPLS labeling mechanism.

This example shows how to configure CoS on an MPLS network that is using a unidirectional circuit cross-connect (CCC) from the ingress provider edge (PE) switch to the egress PE switch. The configuration is for the customer-edge interface of the ingress PE switch. It describes adding the configuration of CoS components to the ingress PE switch, the egress PE switch, and the core provider switches of the existing MPLS network. Because of the unidirectional configuration, the DSCP classifier needs to be configured only on the ingress PE switch.

Requirements

This example uses the following hardware and software components:

- Junos OS Release 10.1 or later for EX Series switches
- Three EX Series switches

Before you configure CoS with MPLS, be sure you have:

Configured an MPLS network with two PE switches and one provider switch. See "Basic MPLS Configuration" on page 46. This example assumes that an MPLS network has been configured using a cross circuit-connect (CCC).

Overview and Topology

IN THIS SECTION

- Topology | **1971**

This example describes adding custom classifiers and custom rewrite rules to switches in an MPLS network that is using MPLS over CCC.

It is a unidirectional configuration. Therefore, you need to configure custom classifiers and custom rewrite rules as follows:

- On the ingress PE switch: custom DSCP classifier and custom EXP rewrite rule
- On the egress PE switch: custom EXP classifier
- On the provider switch: customer EXP classifier and custom EXP rewrite rule

> **NOTE**: You can also configure schedulers and shapers as needed. If you are using **assured-forwarding**, **expedited-forwarding**, or other custom forwarding classes, we recommend that you configure a scheduler to support that forwarding class. See *Defining CoS Schedulers and Scheduler Maps (CLI Procedure)*.

The example creates a custom DSCP classifier (**dscp1**) on the ingress PE switch and binds this classifier to the CCC interface. It includes configuration of a policer on the ingress PE switch. The policer is applied as a filter on the label-switched path (LSP) **lsp_to_pe2_ge1** to ensure that the amount of traffic forwarded through the LSP never exceeds the requested bandwidth allocation.

This example creates a custom EXP rewrite rule (**exp1**) on the ingress PE switch, specifying a loss-priority and code point to be used for the expedited-forwarding class as the packet travels through the LSP. The switch applies this custom rewrite rule on the core interfaces **ge-0/0/5.0** and **ge-0/0/6.0**, which are the egress interfaces for this switch.

Table 42 on page 1968 shows the CoS configuration components added to the ingress PE switch.

Table 42: CoS Configuration Components on the Ingress PE Switch

Property	Settings	Description
Local PE switch hardware	EX Series switch	PE-1
Policing filter configured and applied to the LSP.	**policing filter mypolicer** **filter myfilter**	Name of the rate-limiting policer. Name of the filter, which refers to the policer
Custom DSCP classifier	**dscp1**	Specifies the name of the custom DSCP classifier
Custom EXP rewrite rule	**e1**	Name of the custom EXP rewrite rule.

Table 42: CoS Configuration Components on the Ingress PE Switch *(Continued)*

Property	Settings	Description
Customer-edge interface	**ge-0/0/1.0**	Interface that receives packets from devices outside the network. The custom DSCP classifier must be specified on this CCC interface.
Core interfaces	**ge-0/0/5.0** and **ge-0/0/6.0**	Interfaces that transmit MPLS packets to other switches within the MPLS network. The EXP rewrite rule is applied implicitly to these interfaces.

Table 43 on page 1969 shows the CoS configuration components added to the egress PE switch in this example.

Table 43: CoS Configuration Components of the Egress PE Switch

Property	Settings	Description
Remote provider edge switch hardware	EX Series switch	PE-2
Custom EXP classifier	**exp1**	Name of custom EXP classifier
Customer-edge interface	**ge-0/0/1.0**	Interface that transmits packets from this network to devices outside the network. No CoS classifier is specified for this interface. A scheduler can be specified.

Table 43: CoS Configuration Components of the Egress PE Switch *(Continued)*

Property	Settings	Description
Core interfaces	**ge-0/0/7.0** and **ge-0/0/8.0**	Core interfaces on PE-2 that receive MPLS packets from the provider switch. The EXP classifier is enabled by default on the switch and applied implicitly to these interfaces.

Table 44 on page 1970 shows the MPLS configuration components used for the provider switch in this example.

Table 44: CoS Configuration Components of the Provider Switch

Property	Settings	Description
Provider switch hardware	EX Series switch	Transit switch within the MPLS network configuration.
Custom EXP classifier	**exp1**	Name of the custom EXP classifier.
Custom EXP rewrite rule	**e1**	Name of the custom EXP rewrite rule.
Core interfaces receiving packets from other MPLS switches.	**ge-0/0/5.0** and **ge-0/0/6.0**	Interfaces that connect the provider switch to the ingress PE switch (PE-1). The EXP classifier is enabled by default on the switch and applied implicitly to these interfaces.
Core interfaces transmitting packets to other switches within the MPLS network.	**ge-0/0/7.0** and **ge-0/0/8.0**	Interfaces that transmit packets to the egress PE (PE-2). The EXP rewrite rule is applied implicitly on these interfaces. Schedulers can also be specified and will be applied to these interfaces.

Topology

Configuring the Local PE Switch

IN THIS SECTION

- Procedure | 1971

Procedure

CLI Quick Configuration

To quickly configure a custom DSCP classifier, custom EXP rewrite rule, and a policer on the local PE switch, copy the following commands and paste them into the switch terminal window of PE-1:

```
[edit]

set class-of-service classifiers dscpset class-of-service classifiers dscp dscp1 import default
set class-of-service classifiers dscp dscp1 forwarding-class expedited-forwarding loss-priority
low code-points 000111
set class-of-service rewrite-rules exp e1 forwarding-class expedited-forwarding loss-priority
low code-point 111
set class-of-service interfaces ge-0/0/1 unit 0 classifier dscp1
set firewall policer mypolicer if-exceeding bandwidth-limit 500m
set firewall policer mypolicer if-exceeding burst-size-limit 33553920
set firewall policer mypolicer then discard
set firewall family any filter myfilter term t1 then policer mypolicer
set protocols mpls label-switched-path lsp_to_pe2_ge1 to 127.1.1.3 policing filter myfilter
```

Step-by-Step Procedure

To configure a custom DSCP classifier, custom EXP rewrite rule, and a policer on the ingress PE switch:

1. Import the default DSCP classifier classes to the custom DSCP classifier that you are creating:

```
[edit class-of-service]
user@switch# set classifiers dscp dscp1 import default
```

2. Add the expedited-forwarding class to this custom DSCP classifier, specifying a loss priority and code point:

```
[edit class-of-service]
user@switch# set classifiers dscp dscp1 forwarding-class expedited-forwarding loss-priority
low code-points 000111
```

3. Specify the values for the custom EXP rewrite rule, **e1**:

```
[edit class-of-service]
user@switch# set rewrite-rules exp e1 forwarding-class expedited-forwarding loss-priority low
code-point 111
```

4. Bind the DSCP classifier to the CCC interface:

```
[edit class-of-service]
user@switch# set class-of-service interfaces ge-0/0/1 unit 0 classifier dscp1
```

5. Specify the number of bits per second permitted, on average, for the firewall policer, which will later be applied to the LSP:

```
[edit firewall]
set policer mypolicer if-exceeding bandwidth-limit 500m
```

6. Specify the maximum size permitted for bursts of data that exceed the given bandwidth limit for this policer:

```
[edit firewall policer]
set mypolicer if-exceeding burst-size-limit 33553920
```

7. Discard traffic that exceeds the rate limits for this policer:

```
[edit firewall policer]
set mypolicer then discard
```

8. To reference the policer, configure a filter term that includes the policer action:

```
[edit firewall]
user@switch# set family any filter myfilter term t1 then policer
mypolicer
```

9. Apply the filter to the LSP:

```
[edit protocols mpls]
set label-switched-path lsp_to_pe2_ge1 policing filter myfilter
```

Results

Display the results of the configuration:

```
[edit]
user@switch# show
class-of-service {
    classifiers {
        dscp dscp1 {
            import default;
            forwarding-class expedited-forwarding {
                loss-priority low code-points 000111;
            }
        }
    }
    interfaces {
        ge-0/0/1 {
            unit 0 {
                classifiers {
                    dscp dscp1;
                }
            }
        }
```

```
    }
    rewrite-rules {
        exp e1 {
            forwarding-class expedited-forwarding {
                loss-priority low code-point 111;
            }
        }
    }
}
firewall {
    family any {
        filter myfilter {
            term t1 {
                then policer mypolicer;
            }
        }
    }
    policer mypolicer {
        if-exceeding {
            bandwidth-limit 500m;
            burst-size-limit 33553920;
        }
        then discard;
    }
}
```

Configuring the Remote PE Switch

IN THIS SECTION

- Procedure | **1975**

Procedure

CLI Quick Configuration

To quickly configure a custom EXP classifier on the remote PE switch, copy the following commands and paste them into the switch terminal window of PE-2:

```
[edit]
set class-of-service classifiers exp exp1 import default
set class-of-service classifiers exp exp1 forwarding-class expedited-forwarding loss-priority
low code-points 010
```

Step-by-Step Procedure

To configure a custom EXP classifier on the egress PE switch:

1. Import the default EXP classifier classes to the custom EXP classifier that you are creating:

```
[edit class-of-service]
user@switch# set classifiers exp exp1 import default
```

2. Add the expedited-forwarding class to this custom EXP classifier, specifying a loss priority and code point:

```
[edit class-of-service]
user@switch# set classifiers exp exp1 forwarding-class expedited-forwarding loss-priority low
code-points 010
```

Results

Display the results of the configuration:

```
[edit]
 user@switch# show
class-of-service {
    classifiers {
        exp exp1 {
```

```
            import default;
            forwarding-class expedited-forwarding {
                loss-priority low code-points 010;
            }
        }
    }
}
```

Configuring the Provider Switch

IN THIS SECTION

- Procedure | **1976**

Procedure

CLI Quick Configuration

To quickly configure a custom EXP classifier and a custom EXP rewrite rule on the provider switch, copy the following commands and paste them into the switch terminal window of the provider switch:

```
[edit]
set class-of-service classifiers exp exp1 import default
set class-of-service classifiers exp exp1 forwarding-class expedited-forwarding loss-priority
low code-points 010
set class-of-service rewrite-rules exp e1 forwarding-class expedited-forwarding loss-priority
low code-point 111
```

Step-by-Step Procedure

To configure a custom EXP classifier and a custom EXP rewrite rule on the provider switch:

1. Import the default EXP classifier classes to the custom EXP classifier that you are creating:

```
[edit class-of-service]
user@switch# set classifiers exp exp1 import default
```

2. Add the expedited-forwarding class to this custom EXP classifier, specifying a loss priority and code point:

```
[edit class-of-service]
user@switch# set classifiers exp exp1 forwarding-class expedited-forwarding loss-priority low
code-points 010
```

3. Specify the values for the custom EXP rewrite rule, **e1**:

```
[edit class-of-service]
user@switch# set rewrite-rules exp e1 forwarding-class expedited-forwarding loss-priority low
code-point 111
```

Results

Display the results of the configuration:

```
[edit]
 user@switch# show
class-of-service {
    classifiers {
        exp exp1 {
            import default;
            forwarding-class expedited-forwarding {
                loss-priority low code-points 010;
            }
        }
    }
    rewrite-rules {
        exp e1 {
            forwarding-class expedited-forwarding {
                loss-priority low code-point 111;
            }
```

```
        }
    }
}
```

Verification

IN THIS SECTION

- Verifying That the Policer Firewall Filter Is Operational | **1978**
- Verifying That the CoS Classifiers Are Going to the Right Queue | **1979**
- Verifying the CoS Forwarding Table Mapping | **1983**
- Verifying the Rewrite Rules | **1983**

To confirm that the configuration is working properly, perform these tasks:

Verifying That the Policer Firewall Filter Is Operational

Purpose

Verify the operational state of the policer that is configured on the ingress PE switch.

Action

```
user@switch> show firewall

Filter: myfilter
Policers:
Name                                                Packets
mypolicer-t1                                              0
```

Meaning

This output shows that the firewall filter **mypolicer** has been created.

Verifying That the CoS Classifiers Are Going to the Right Queue

Purpose

Verify that the CoS classifiers are going to the right queue.

Action

```
user@switch> show class-of-service forwarding-table classifier

Classifier table index: 7, # entries: 64, Table type: DSCP
Entry #   Code point   Forwarding-class #   PLP
   0        000000        0       0
   1        000001        0       0
   2        000010        0       0
   3        000011        0       0
   4        000100        0       0
   5        000101        0       0
   6        000110        0       0
   7        000111        0       0
   8        001000        0       0
   9        001001        0       0
  10        001010        0       0
  11        001011        0       0
  12        001100        0       0
  13        001101        0       0
  14        001110        0       0
  15        001111        0       0
  16        010000        0       0
  17        010001        0       0
  18        010010        0       0
  19        010011        0       0
  20        010100        0       0
  21        010101        0       0
  22        010110        0       0
  23        010111        0       0
  24        011000        0       0
  25        011001        0       0
  26        011010        0       0
  27        011011        0       0
  28        011100        0       0
  29        011101        0       0
```

```
  30        011110        0       0
  31        011111        0       0
  32        100000        0       0
  33        100001        0       0
  34        100010        0       0
  35        100011        0       0
  36        100100        0       0
  37        100101        0       0
  38        100110        0       0
  39        100111        0       0
  40        101000        0       0
  41        101001        0       0
  42        101010        0       0
  43        101011        0       0
  44        101100        0       0
  45        101101        0       0
  46        101110        0       0
  47        101111        0       0
  48        110000        3       0
  49        110001        3       0
  50        110010        3       0
  51        110011        3       0
  52        110100        3       0
  53        110101        3       0
  54        110110        3       0
  55        110111        3       0
  56        111000        3       0
  57        111001        3       0
  58        111010        3       0
  59        111011        3       0
  60        111100        3       0
  61        111101        3       0
  62        111110        3       0
  63        111111        3       0

Classifier table index: 11, # entries: 8, Table type: IEEE 802.1
Entry #   Code point   Forwarding-class #   PLP
   0           000        0       0
   1           001        0       0
   2           010        0       0
   3           011        0       0
   4           100        0       0
   5           101        0       0
```

```
   6           110        3       0
   7           111        3       0

Classifier table index: 12, # entries: 8, Table type: IPv4 precedence
Entry #   Code point   Forwarding-class #   PLP
   0           000        0       0
   1           001        0       0
   2           010        0       0
   3           011        0       0
   4           100        0       0
   5           101        0       0
   6           110        3       0
   7           111        3       0

Classifier table index: 16, # entries: 8, Table type: Untrust
Entry #   Code point   Forwarding-class #   PLP
   0           000        0       0
   1           001        0       0
   2           010        0       0
   3           011        0       0
   4           100        0       0
   5           101        0       0
   6           110        0       0
   7           111        0       0

Classifier table index: 9346, # entries: 64, Table type: DSCP
Entry #   Code point   Forwarding-class #   PLP
   0        000000        0       0
   1        000001        0       0
   2        000010        0       0
   3        000011        0       0
   4        000100        0       0
   5        000101        0       0
   6        000110        0       0
   7        000111        1       0
   8        001000        0       0
   9        001001        0       0
  10        001010        0       0
  11        001011        0       0
  12        001100        0       0
  13        001101        0       0
  14        001110        0       0
  15        001111        0       0
```

```
16      010000      0      0
17      010001      0      0
18      010010      0      0
19      010011      0      0
20      010100      0      0
21      010101      0      0
22      010110      0      0
23      010111      0      0
24      011000      0      0
25      011001      0      0
26      011010      0      0
27      011011      0      0
28      011100      0      0
29      011101      0      0
30      011110      0      0
31      011111      0      0
32      100000      0      0
33      100001      0      0
34      100010      0      0
35      100011      0      0
36      100100      0      0
37      100101      0      0
38      100110      0      0
39      100111      0      0
40      101000      0      0
41      101001      0      0
42      101010      0      0
43      101011      0      0
44      101100      0      0
45      101101      0      0
46      101110      0      0
47      101111      0      0
48      110000      3      0
49      110001      3      0
50      110010      3      0
51      110011      3      0
52      110100      3      0
53      110101      3      0
54      110110      3      0
55      110111      3      0
56      111000      3      0
57      111001      3      0
58      111010      3      0
```

```
  59        111011        3       0
  60        111100        3       0
  61        111101        3       0
  62        111110        3       0
  63        111111        3       0
```

Meaning

This output shows that a new DSCP classifier has been created, index **9346**, on the ingress PE switch (PE-1).

Verifying the CoS Forwarding Table Mapping

Purpose

For each logical interface, display either the table index of the classifier for a given code point type or the queue number (if it is a fixed classification) in the forwarding table.

Action

```
user@switch> show class-of-service forwarding-table classifier mapping

                       Table Index/
Interface      Index      Q num      Table type
ge-0/0/1.0        92       9346      DSCP
```

Meaning

The results show that the new DSCP classifier, index number **9346**, is bound to interface **ge-0/0/1.0**.

Verifying the Rewrite Rules

Purpose

Display mapping of the queue number and loss priority to code point value for each rewrite rule as it exists in the forwarding table.

Action

```
user@switch>show class-of-service forwarding-table rewrite-rule

Rewrite table index: 31, # entries: 4, Table type: DSCP
FC#     Low bits  State      High bits  State
0       000000  Enabled       000000  Enabled
1       101110  Enabled       101110  Enabled
2       001010  Enabled       001100  Enabled
3       110000  Enabled       111000  Enabled

Rewrite table index: 34, # entries: 4, Table type: IEEE 802.1
FC#     Low bits  State      High bits  State
0          000  Enabled          001  Enabled
1          010  Enabled          011  Enabled
2          100  Enabled          101  Enabled
3          110  Enabled          111  Enabled

Rewrite table index: 35, # entries: 4, Table type: IPv4 precedence
FC#     Low bits  State      High bits  State
0          000  Enabled          000  Enabled
1          101  Enabled          101  Enabled
2          001  Enabled          001  Enabled
3          110  Enabled          111  Enabled

Rewrite table index: 9281, # entries: 1, Table type: EXP
FC#     Low bits  State      High bits  State
1          111  Enabled          000  Disabled
```

Meaning

This output shows that a new EXP classifier with the index number **9281** has been created.

Understanding CoS MPLS EXP Classifiers and Rewrite Rules

IN THIS SECTION

- EXP Classifiers | **1986**
- EXP Rewrite Rules | **1987**

- Schedulers | **1988**

You can use *class of service* (CoS) within MPLS networks to prioritize certain types of traffic during periods of congestion by applying packet classifiers and rewrite rules to the MPLS traffic. MPLS classifiers are global and apply to all interfaces configured as `family mpls` interfaces.

When a packet enters a customer-edge interface on the ingress provider edge (PE) switch, the switch associates the packet with a particular CoS servicing level before placing the packet onto the label-switched path (LSP). The switches within the LSP utilize the CoS value set at the ingress PE switch to determine the CoS service level. The CoS value embedded in the classifier is translated and encoded in the MPLS header by means of the experimental (EXP) bits.

EXP classifiers map incoming MPLS packets to a forwarding class and a loss priority, and assign MPLS packets to output queues based on the forwarding class mapping. EXP classifiers are behavior aggregate (BA) classifiers.

EXP rewrite rules change (rewrite) the CoS value of the EXP bits in outgoing packets on the egress queues of the switch so that the new (rewritten) value matches the policies of a targeted peer. Policy matching allows the downstream routing platform or switch in a neighboring network to classify each packet into the appropriate service group.

NOTE: On QFX5200, QFX5100, and EX4600 switches, there is no default EXP classifier. If you want to classify incoming MPLS packets using the EXP bits, you must configure a global EXP classifier. The global EXP classifier applies to all MPLS traffic on interfaces configured as `family mpls`.

On QFX10000 switches, there is a no default EXP classifier. If you want to classify incoming MPLS packets using the EXP bits, you must configure EXP classifiers and apply them to logical interfaces configured as `family mpls`. (You cannot apply classifiers to physical interfaces.). You can configure up to 64 EXP classifiers.

There is no default EXP rewrite rule. If you want to rewrite the EXP bit value at the egress interface, you must configure EXP rewrite rules and apply them to logical interfaces.

EXP classifiers and rewrite rules are applied only to interfaces that are configured as `family mpls` (for example, `set interfaces xe-0/0/35 unit 0 family mpls`.)

EXP Classifiers

On QFX5200, QFX5100, EX4600 switches, unlike DSCP and IEEE 802.1p BA classifiers, EXP classifiers are global to the switch and apply to all switch interfaces that are configured as `family mpls`. On QFX10000 switches, you apply EXP classifiers to individual logical interfaces, and different interfaces can use different EXP classifiers.

When you configure and apply an EXP classifier, MPLS traffic on all `family mpls` interfaces uses the EXP classifier, even on interfaces that also have a fixed classifier. If an interface has both an EXP classifier and a fixed classifier, the EXP classifier is applied to MPLS traffic and the fixed classifier is applied to all other traffic.

Also unlike DSCP and IEEE 802.1p BA classifiers, there is no default EXP classifier. If you want to classify MPLS traffic based on the EXP bits, you must explicitly configure an EXP classifier and apply it to the switch interfaces. Each EXP classifier has eight entries that correspond to the eight EXP CoS values (0 through 7, which correspond to CoS bits 000 through 111).

You can configure up to 64 EXP classifiers.

However, on QFX5200, QFX5100, EX4600, switches, the switch uses only one MPLS EXP classifier as a global classifier on all interfaces. After you configure an MPLS EXP classifier, you can configure that classifier as the global EXP classifier by including the EXP classifier in the `[edit class-of-service system-defaults classifiers exp]` hierarchy level. All switch interfaces configured as `family mpls` use the global EXP classifier to classify MPLS traffic.

On these switches, only one EXP classifier can be configured as the global EXP classifier at any time. If you want to change the global EXP classifier, delete the global EXP classifier configuration (use the **user@switch# delete class-of-service system-defaults classifiers exp** configuration statement), then configure the new global EXP classifier.

NOTE: QFX5130 switch does not support MPLS CoS.

QFX10000 switches do not support global EXP classifiers. You can configure one EXP classifier and apply it to multiple logical interfaces, or configure multiple EXP classifiers and apply different EXP classifiers to different logical interfaces.

If an EXP classifier is not configured, then if a fixed classifier is applied to the interface, the MPLS traffic uses the fixed classifier. (Switches that have a default EXP classifier use the default classifier.) If no EXP classifier and no fixed classifier are applied to the interface, MPLS traffic is treated as best-effort traffic using the 802.1 default untrusted classifier. DSCP classifiers are not applied to MPLS traffic.

On QFX5200, QFX5100, EX4600 switches, because the EXP classifier is global, you cannot configure some ports to use a fixed IEEE 802.1p classifier for MPLS traffic on some interfaces and the global EXP classifier for MPLS traffic on other interfaces. When you configure a global EXP classifier, all MPLS traffic on all interfaces uses the EXP classifier.

NOTE: The switch uses only the outermost label of incoming EXP packets for classification.

NOTE: MPLS packets with 802.1Q tags are not supported.

On QFX5220 switch, you can use class of service (CoS) within MPLS networks to prioritize certain types of traffic during periods of congestion by applying packet classifiers and rewrite rules to the MPLS traffic. We have also added the MPLS EXP rewrite support.

- Default CoS on the Provider (P) and Provider Edge (PE) routers for MPLS interfaces – The MPLS traffic uses the default EXP classifier. MPLS traffic is treated as best-effort traffic using the 802.1 default untrusted classifier. The default EXP classifier applies to all MPLS traffic on interfaces configured as `family mpls`. DSCP classifiers are not applied to MPLS traffic.
- Default CoS on PE routers for Layer 3 interfaces – By default, all L3VPN logical interfaces are bound to default Differentiated Services Code Point (DSCP) classifiers.

If you apply an EXP classifier on a penultimate hop popping (PHP) node, then by default, the IP header time-to-live (TTL) value is overwritten by the MPLS header TLL value, and the IP header DSCP bits are over written by a zero (0), which signifies uniform mode. On Junos OS Evolved, to use pipe mode, where IP header TTL and IP header DSCP bits are not overwritten, you should configure the following command:

```
set protocols mpls no-propagate-ttl
```

However, on Junos OS, you can configure MPLS CoS without the set protocols mpls no-propagate-ttl command.

NOTE: The DSCP of IP in MPLS packets can't be remarked either at PE or P routers.

EXP Rewrite Rules

As MPLS packets enter or exit a network, edge switches might be required to alter the class-of-service (CoS) settings of the packets. EXP *rewrite rules* set the value of the EXP CoS bits within the header of the outgoing MPLS packet on `family mpls` interfaces. Each rewrite rule reads the current forwarding class and loss priority associated with the packet, locates the chosen CoS value from a table, and writes that CoS value into the packet header, replacing the old CoS value. EXP rewrite rules apply only to MPLS traffic.

EXP rewrite rules apply only to logical interfaces. You cannot apply EXP rewrite rules to physical interfaces.

There are no default EXP rewrite rules. If you want to rewrite the EXP value in MPLS packets, you must configure EXP rewrite rules and apply them to logical interfaces. If no rewrite rules are applied, all MPLS labels that are pushed have a value of zero (0). The EXP value remains unchanged on MPLS labels that are swapped.

You can configure up to 64 EXP rewrite rules, but you can only apply 16 EXP rewrite rules at any time on the switch. On a given logical interface, all pushed MPLS labels have the same EXP rewrite rule applied to them. You can apply different EXP rewrite rules to different logical interfaces on the same physical interface.

You can apply an EXP rewrite rule to an interface that has a DSCP, DSCP IPv6, or IEEE 802.1p rewrite rule. Only MPLS traffic uses the EXP rewrite rule. MPLS traffic does not use DSCP or DSCP IPv6 rewrite rules.

If the switch is performing penultimate hop popping (PHP), EXP rewrite rules do not take effect. If both an EXP classifier and an EXP rewrite rule are configured on the switch, then the EXP value from the last popped label is copied into the inner label. If either an EXP classifier or an EXP rewrite rule (but not both) is configured on the switch, then the inner label EXP value is sent unchanged.

NOTE: On each physical interface, either all forwarding classes that are being used on the interface must have rewrite rules configured or no forwarding classes that are being used on the interface can have rewrite rules configured. On any physical port, do not mix forwarding classes with rewrite rules and forwarding classes without rewrite rules.

Schedulers

The schedulers for using CoS with MPLS are the same as for the other CoS configurations on the switch. Default schedulers are provided only for the best-effort, fcoe, no-loss, and network-control default forwarding classes. If you configure a custom forwarding class for MPLS traffic, you need to configure a scheduler to support that forwarding class and provide bandwidth to that forwarding class.

Configuring Rewrite Rules for MPLS EXP Classifiers

You configure EXP rewrite rules to alter CoS values in outgoing MPLS packets on the outbound `family mpls` interfaces of a switch to match the policies of a targeted peer. Policy matching allows the downstream routing platform or switch in a neighboring network to classify each packet into the appropriate service group.

To configure an EXP CoS rewrite rule, create the rule by giving it a name and associating it with a forwarding class, loss priority, and code point. This creates a rewrite table. After the rewrite rule is created, enable it on a logical `family mpls` interface. EXP rewrite rules can only be enabled on logical `family`

mpls interfaces, not on physical interfaces or on interfaces of other family types. You can also apply an existing EXP rewrite rule on a logical interface.

NOTE: There are no default rewrite rules.

You can configure up to 64 EXP rewrite rules, but you can only use 16 EXP rewrite rules at any time on the switch. On a given family mpls logical interface, all pushed MPLS labels have the same EXP rewrite rule applied to them. You can apply different EXP rewrite rules to different logical interfaces on the same physical interface.

NOTE: On each physical interface, either all forwarding classes that are being used on the interface must have rewrite rules configured, or no forwarding classes that are being used on the interface can have rewrite rules configured. On any physical port, do not mix forwarding classes with rewrite rules and forwarding classes without rewrite rules.

NOTE: To replace an existing rewrite rule on the interface with a new rewrite rule of the same type, first explicitly remove the existing rewrite rule and then apply the new rule.

To create an EXP rewrite rule for MPLS traffic and enable it on a logical interface:

1. Create an EXP rewrite rule:

```
user@switch# set class-of-service rewrite-rules exp rewrite-rule-name forwarding-class
forwarding-class-name loss-priority level code-points [aliases] [bit-patterns]
```

   For example, to configure an EXP rewrite rule named exp-rr-1 for a forwarding class named mpls-1 with a loss priority of low that rewrites the EXP code point value to 001:

```
user@switch# set class-of-service rewrite-rules exp exp-rr-1 forwarding-class mpls-1 loss-
priority low code-points 001
```

2. Apply the rewrite rule to a logical interface:

```
user@switch # set class-of-service interfaces interface-name unit logical-unit rewrite-rules
exp rewrite-rule-name
```

For example, to apply a rewrite rule named `exp-rr-1` to logical interface `xe-0/0/10.0`:

```
user@switch# set class-of-service interfaces xe-0/0/10 unit 0 rewrite-rules exp exp-rr-1
```

NOTE: In this example, all forwarding classes assigned to port xe-0/0/10 must have rewrite rules. Do not mix forwarding classes that have rewrite rules with forwarding classes that do not have rewrite rules on the same interface.

Configuring CoS Bits for an MPLS Network

When traffic enters a labeled-switch path (LSP) tunnel, the CoS bits in the MPLS header are set in one of two ways:

- The number of the output queue into which the packet was buffered and the packet loss priority (PLP) bit are written into the MPLS header and are used as the packet's CoS value. This behavior is the default, and no configuration is required. The Junos OS Class of Service User Guide for Routing Devices explains the IP CoS values, and summarizes how the CoS bits are treated.
- You set a fixed CoS value on all packets entering the LSP tunnel. A fixed CoS value means that all packets entering the LSP receive the same class of service.

To set a fixed CoS value on all packets entering the LSP:

1. Specify a class of service value for the LSP:

NOTE: The CoS value set using the `class-of-service` statement at the `[edit protocols mpls]` hierarchy level supersedes the CoS value set at the `[edit class-of-service]` hierarchy level for an interface. Effectively, the CoS value configured for an LSP overrides the CoS value set for an interface.

```
[edit protocols mpls]
user@switch# set class-of-service cos-value
```

Configuring a Global MPLS EXP Classifier

EXP packet classification associates incoming packets with a particular MPLS CoS servicing level. EXP behavior aggregate (BA) classifiers examine the MPLS EXP value in the packet header to determine the CoS settings applied to the packet. EXP BA classifiers allow you to set the forwarding class and loss priority of an MPLS packet based on the incoming CoS value.

You can configure up to 64 EXP classifiers, however, the switch uses only one MPLS EXP classifier as a global classifier, which is applied only on interfaces configured as `family mpls`. All `family mpls` switch interfaces use the global EXP classifier to classify MPLS traffic.

There is no default EXP classifier. If you want to classify incoming MPLS packets using the EXP bits, you must configure a global EXP classifier. The global classifier applies to all MPLS traffic on all `family mpls` interfaces.

If a global EXP classifier is configured, MPLS traffic on `family mpls` interfaces uses the EXP classifier. If a global EXP classifier is not configured, then if a fixed classifier is applied to the interface, the MPLS traffic uses the fixed classifier. If no EXP classifier and no fixed classifier is applied to the interface, MPLS traffic is treated as best-effort traffic. DSCP classifiers are not applied to MPLS traffic.

To configure an MPLS EXP classifier using the CLI:

1. Create an EXP classifier and associate it with a forwarding class, a loss priority, and a code point:

```
[edit class-of-service classifiers]
user@switch# set (dscp | ieee-802.1 | exp) classifier-name forwarding-class forwarding-class-name loss-priority level code-points [aliases] [bit-patterns]
```

2. Apply the EXP classifier to the switch interfaces:

```
[edit class-of-service]
user@switch# set system-defaults classifiers exp classifier-name
```

RELATED DOCUMENTATION

- Class of Service User Guide (Routers and EX9200 Switches)

CHAPTER 16

Generalized MPLS (GMPLS)

IN THIS CHAPTER

- GMPLS Configuration | **1992**

GMPLS Configuration

IN THIS SECTION

- Introduction to GMPLS | **1992**
- GMPLS Terms and Acronyms | **1994**
- GMPLS Operation | **1995**
- GMPLS and OSPF | **1995**
- GMPLS and CSPF | **1996**
- GMPLS Features | **1996**
- Configuring MPLS Paths for GMPLS | **1997**
- Tracing LMP Traffic | **1997**
- Configuring MPLS LSPs for GMPLS | **1998**
- Gracefully Tearing Down GMPLS LSPs | **2002**
- GMPLS RSVP-TE VLAN LSP Signaling Overview | **2004**
- Example: Configuring GMPLS RSVP-TE VLAN LSP Signaling | **2011**

Introduction to GMPLS

Traditional MPLS is designed to carry Layer 3 IP traffic using established IP-based paths and associating these paths with arbitrarily assigned labels. These labels can be configured explicitly by a network administrator, or can be dynamically assigned by means of a protocol such as LDP or RSVP.

GMPLS generalizes MPLS in that it defines labels for switching varying types of Layer 1, Layer 2, or Layer 3 traffic. GMPLS nodes can have links with one or more of the following switching capabilities:

- Fiber-switched capable (FSC)
- Lambda-switched capable (LSC)
- Time-division multiplexing (TDM) switched-capable (TSC)
- Packet-switched capable (PSC)

Label-switched paths (LSPs) must start and end on links with the same switching capability. For example, routers can establish packet-switched LSPs with other routers. The LSPs might be carried over a TDM-switched LSP between SONET add/drop multiplexers (ADMs), which in turn might be carried over a lambda-switched LSP.

The result of this extension of the MPLS protocol is an expansion in the number of devices that can participate in label switching. Lower-layer devices, such as OXCs and SONET ADMs, can now participate in GMPLS signaling and set up paths to transfer data. A router can participate in signaling optical paths across a transport network.

Two service models determine the visibility that a client node (a router, for example) has into the optical core or transport network. The first is through a user-to-network interface (UNI), which is often referred to as the overlay model. The second is known as the peer model. Juniper Networks supports both models.

NOTE: There is not necessarily a one-to-one correspondence between a physical interface and a GMPLS interface. If a GMPLS connection uses a nonchannelized physical connector, the GMPLS label can use the physical port ID. However, the label for channelized interfaces often is based on a channel or time slot. Consequently, it is best to refer to GMPLS labels as identifiers for a resource on a traffic engineering link.

To establish LSPs, GMPLS uses the following mechanisms:

- An out-of-band control channel and a data channel—RSVP messages for LSP setup are sent over an out-of-band control network. Once the LSP setup is complete and the path is provisioned, the data channel is up and can be used to carry traffic. The Link Management Protocol (LMP) is used to define and manage the data channels between a pair of nodes. You can optionally use LMP to establish and maintain LMP control channels between peers running the same Junos OS Release.
- RSVP-TE extensions for GMPLS—RSVP-TE is already designed to signal the setup of packet LSPs. This has been extended for GMPLS to be able to request path setup for various kinds of LSPs (nonpacket) and request labels like wavelengths, time slots, and fibers as label objects.
- Bidirectional LSPs—Data can travel both ways between GMPLS devices over a single path, so nonpacket LSPs are signaled to be bidirectional.

GMPLS Terms and Acronyms

IN THIS SECTION

- Generalized MPLS (GMPLS) | **1994**
- Forwarding adjacency | **1994**
- GMPLS label | **1994**
- GMPLS LSP types | **1994**
- Link Management Protocol | **1995**
- Traffic engineering link | **1995**

Generalized MPLS (GMPLS)

An extension to MPLS that allows data from multiple layers to be switched over label-switched paths (LSPs). GMPLS LSP connections are possible between similar Layer 1, Layer 2, and Layer 3 devices.

Forwarding adjacency

A forwarding path for sending data between GMPLS-enabled devices.

GMPLS label

Layer 3 identifiers, fiber port, time-division multiplexing (TDM) time slot, or dense wavelength-division multiplexing (DWDM) wavelength of a GMPLS-enabled device used as a next-hop identifier.

GMPLS LSP types

The four types of GMPLS LSPs are:

- Fiber-switched capable (FSC)—LSPs are switched between two fiber-based devices, such optical cross-connects (OXCs) that operate at the level of individual fibers.
- Lambda-switched capable (LSC)—LSPs are switched between two DWDM devices, such as such as OXCs that operate at the level of individual wavelengths.
- TDM-switched capable (TDM)—LSPs are switched between two TDM devices, such as SONET ADMs.
- Packet-switched capable (PSC)—LSPs are switched between two packet-based devices, such as routers or ATM switches.

Link Management Protocol

A protocol used to define a forwarding adjacency between peers and to maintain and allocate resources on the traffic engineering links.

Traffic engineering link

A logical connection between GMPLS-enabled devices. Traffic engineering links can have addresses or IDs and are associated with certain resources or interfaces. They also have certain attributes (encoding-type, switching capability, bandwidth, and so on). The logical addresses can be routable, although this is not required because they are acting as link identifiers. Each traffic engineering link represents a forwarding adjacency between a pair of devices.

GMPLS Operation

The basic functionality of GMPLS requires close interaction between RSVP and LMP. It works in the following sequence:

1. LMP notifies RSVP of the new entities:
    - Traffic engineering link (forwarding adjacency)
    - Resources available for the traffic engineering link
    - Control peer
2. GMPLS extracts the LSP attributes from the configuration and requests RSVP to signal one or more specific paths, which are specified by the traffic engineering link addresses.
3. RSVP determines the local traffic engineering link, corresponding control adjacency and active control channel, and transmission parameters (such as IP destination). It requests that LMP allocate a resource from the traffic engineering link with the specified attributes. If LMP finds a resource matching the attributes, label allocation succeeds. RSVP sends a PathMsg hop by hop until it reaches the target router.
4. When the target router receives the PathMsg, RSVP again requests that LMP allocate a resource based on the signaled parameters. If label allocation succeeds, the router sends back a ResvMsg.
5. If the signaling is successful, a bidirectional optical path is provisioned.

GMPLS and OSPF

You can configure OSPF for GMPLS. OSPF is an interior gateway protocol (IGP) that routes packets within a single autonomous system (AS). OSPF uses link-state information to make routing decisions.

GMPLS and CSPF

GMPLS introduces extra constraints for computing paths for GMPLS LSPs that use CSPF. These additional constraints affect the following link attributes:

- Signal type (minimum LSP bandwidth)
- Encoding type
- Switching type

These new constraints are populated in the traffic engineering database with the exchange of an interface-switching capability descriptor type, length, value (TLV) through an IGP.

The ignored constraints that are exchanged through the interface switching capability descriptor include:

- Maximum LSP bandwidth
- Maximum transmission unit (MTU)

The CSPF path computation is the same as in non-GMPLS environments, except that the links are also limited by GMPLS constraints.

Each link can have multiple interface-switching capability descriptors. All the descriptors are checked before a link is rejected.

The constraints are checked in the following order:

1. The signal type configured for the GMPLS LSP signifies the amount of bandwidth requested. If the desired bandwidth is less than the minimum LSP bandwidth, the interface-switching descriptor is rejected.
2. The encoding type of the link for the ingress and the egress interfaces should match. The encoding type is selected and stored at the ingress node after all the constraints are satisfied by the link and is used to select the link on the egress node.
3. The switching type of the links of the intermediate switches should match that of the GMPLS LSP specified in the configuration.

GMPLS Features

The Junos OS includes the following GMPLS functionality:

- An out-of-band control plane makes it possible to signal LSP path setup.
- RSVP-TE extensions support additional objects beyond Layer 3 packets, such as ports, time slots, and wavelengths.

- The LMP protocol creates and maintains a database of traffic engineering links and peer information. Only the static version of this protocol is supported in the Junos OS. You can optionally configure LMP to establish and maintain LMP control channels between peers running the same Junos OS Release.
- Bidirectional LSPs are required between devices.
- Several GMPLS label types that are defined in RFC 3471, *Generalized MPLS—Signaling Functional Description*, such as MPLS, Generalized, SONET/SDH, Suggested, and Upstream, are supported. Generalized labels do not contain a type field, because the nodes should know from the context of their connection what type of label to expect.
- Traffic parameters facilitate GMPLS bandwidth encoding and SONET/SDH formatting.
- Other supported attributes include interface identification and errored interface identification, user-to-network (UNI)-style signaling, and secondary LSP paths.

Configuring MPLS Paths for GMPLS

As part of the configuration for GMPLS, you need to establish an MPLS path for each unique device connected through GMPLS. Configure the traffic engineering link remote address as the address at the `[edit protocols mpls path path-name]` hierarchy level. Constrained Shortest Path First (CSPF) is supported so you can choose either the `strict` or `loose` option with the address.

See LMP Configuration Overview for information about how to obtain a traffic engineering link remote address.

To configure the MPLS path, include the `path` statement at the `[edit protocols mpls]` hierarchy level:

```
[edit protocols mpls]
path path-name {
    next-hop-address (strict | loose);
}
```

For information about how to configure MPLS paths, see "Creating Named Paths" on page 594.

Tracing LMP Traffic

To trace LMP protocol traffic, include the `traceoptions` statement at the `[edit protocols link-management]` hierarchy level:

```
[edit protocols link-management]
traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
```

```
    flag flag <flag-modifier> <disable>;
}
```

Use the `file` statement to specify the name of the file that receives the output of the tracing operation. All files are placed in the directory /var/log.

The following trace flags display the operations associated with the sending and receiving of various LMP messages:

- `all`—Trace all available operations
- `hello-packets`—Trace hello packets on any LMP control channel
- `init`—Output from the initialization messages
- `packets`—Trace all packets other than hello packets on any LMP control channel
- `parse`—Operation of the parser
- `process`—Operation of the general configuration
- `route-socket`—Operation of route socket events
- `routing`—Operation of the routing protocols
- `server`—Server processing operations
- `show`—Servicing operations for `show` commands
- `state`—Trace state transitions of the LMP control channels and traffic engineering links

Each flag can carry one or more of the following flag modifiers:

- `detail`—Provide detailed trace information
- `receive`—Packets being received
- `send`—Packets being transmitted

Configuring MPLS LSPs for GMPLS

IN THIS SECTION

- Configuring the Encoding Type | **1999**
- Configuring the GPID | **2000**
- Configuring the Signal Bandwidth Type | **2001**

Configuring GMPLS Bidirectional LSPs | 2001

Allowing Nonpacket GMPLS LSPs to Establish Paths Through Routers Running Junos OS | 2001

To enable the proper GMPLS switching parameters, configure the label-switched path (LSP) attributes that are appropriate for your network connection. The default value for `switching-type` is `psc-1`, which is also appropriate for standard MPLS.

To configure the LSP attributes, include the `lsp-attributes` statement at the `[edit protocols mpls label-switched-path` *`lsp-name`*`]` hierarchy level:

```
[edit protocols mpls label-switched-path lsp-name]
lsp-attributes {
    encoding-type type;
    gpid gpid;
    signal-bandwidth type;
    switching-type type;
}
```

If you include the `no-cspf` statement in the label-switched path configuration, you must also configure primary and secondary paths, or the configuration cannot be committed.

The following sections describe how to configure each of the LSP attributes for a GMPLS LSP:

Configuring the Encoding Type

You need to specify the encoding type of the payload carried by the LSP. It can be any of the following:

- `ethernet`—Ethernet
- `packet`—Packet
- `pdh`—Plesiochronous digital hierarchy (PDH)
- `sonet-sdh`—SONET/SDH

The default value is `packet`.

To configure the encoding type, include the `encoding-type` statement at the `[edit protocols mpls label-switched-path` *`lsp-name`* `lsp-attributes]` hierarchy level:

```
[edit protocols mpls label-switched-path lsp-name lsp-attributes]
encoding-type type;
```

Configuring the GPID

You need to specify the type of payload carried by the LSP. The payload is the type of packet underneath the MPLS label. The payload is specified by the generalized payload identifier (GPID).

You can specify the GPID with any of the following values:

- `hdlc`—High-Level Data Link Control (HDLC)
- `ethernet`—Ethernet
- `ipv4`—IP version 4 (default)
- `pos-scrambling-crc-16`—For interoperability with other vendors' equipment
- `pos-no-scrambling-crc-16`—For interoperability with other vendors' equipment
- `pos-scrambling-crc-32`—For interoperability with other vendors' equipment
- `pos-no-scrambling-crc-32`—For interoperability with other vendors' equipment
- `ppp`—Point-to-Point Protocol (PPP)

To configure the GPID, include the `gpid` statement at the `[edit protocols mpls label-switched-path` *`lsp-name`* `lsp-attributes]` hierarchy level:

```
[edit protocols mpls label-switched-path lsp-name lsp-attributes]
gpid gpid;
```

Configuring the Signal Bandwidth Type

The signal bandwidth type is the encoding used for path computation and admission control. To configure the signal bandwidth type, include the `signal-bandwidth` statement at the `[edit protocols mpls label-switched-path` *`lsp-name`* `lsp-attributes]` hierarchy level:

```
[edit protocols mpls label-switched-path lsp-name lsp-attributes]
signal-bandwidth type;
```

Configuring GMPLS Bidirectional LSPs

Because MPLS and GMPLS use the same configuration hierarchy for LSPs, it is helpful to know which LSP attributes control LSP functionality. Standard MPLS packet-switched LSPs are unidirectional, whereas GMPLS nonpacket LSPs are bidirectional.

If you use the default packet-switching type of `psc-1`, your LSP becomes unidirectional. To enable a GMPLS bidirectional LSP, you must select a non-packet-switching type option, such as `lambda`, `fiber`, or `ethernet`. Include the `switching-type` statement at the `[edit protocols mpls label-switched-path` *`lsp-name`* `lsp-attributes]` hierarchy level:

```
[edit protocols mpls label-switched-path lsp-name lsp-attributes]
switching-type (lambda | fiber | ethernet);
```

Allowing Nonpacket GMPLS LSPs to Establish Paths Through Routers Running Junos OS

By setting the A-bit in the Admin Status object. you can enable nonpacket GMPLS LSPs to establish paths through routers that run Junos. When an ingress router sends an RSVP PATH message with the Admin Status A-bit set, an external device (not a router running the Junos OS) can either perform a Layer 1 path setup test or help bring up an optical cross-connect.

When set, the A-bit in the Admin Status object indicates the administrative down status for a GMPLS LSP. This feature is used specifically by nonpacket GMPLS LSPs. It does not affect control path setup or data forwarding for packet LSPs.

Junos does not distinguish between the control path setup and data path setup. Other nodes along the network path use RSVP PATH signaling using the A-bit in a meaningful way.

To configure the Admin Status object for a GMPLS LSP, include the `admin-down` statement:

```
admin-down;
```

You can include this statement at the following hierarchy levels:

- [edit protocols mpls label-switched-path *lsp-name*]
- [edit logical-systems *logical-system-name* protocols mpls label-switched-path *lsp-name*]

Gracefully Tearing Down GMPLS LSPs

IN THIS SECTION

- Temporarily Deleting GMPLS LSPs | **2002**
- Permanently Deleting GMPLS LSPs | **2003**
- Configuring the Graceful Deletion Timeout Interval | **2003**

You can gracefully tear down nonpacket GMPLS LSPs. An LSP that is torn down abruptly, a common process in a packet-switched network, can cause stability problems in nonpacket-switched networks. To maintain the stability of nonpacket-switched networks, it might be necessary to tear down LSPs gracefully.

The following sections describe how to tear down GMPLS LSPs gracefully:

Temporarily Deleting GMPLS LSPs

You can gracefully tear down a GMPLS LSP using the `clear rsvp session gracefully` command.

This command gracefully tears down an RSVP session for a nonpacket LSP in two passes. In the first pass, the Admin Status object is signaled along the path to the endpoint of the LSP. During the second pass, the LSP is taken down. Using this command, the LSP is taken down temporarily. After the appropriate interval, the GMPLS LSP is resignaled and then reestablished.

The `clear rsvp session gracefully` command has the following properties:

- It only works on the ingress and egress routers of an RSVP session. If used on a transit router, it has the same behavior as the `clear rsvp session` command.
- It only works for nonpacket LSPs. If used with packet LSPs, it has the same behavior as the `clear rsvp session` command.

For more information, see the CLI Explorer.

Permanently Deleting GMPLS LSPs

When you disable an LSP in the configuration, the LSP is permanently deleted. By configuring the `disable` statement, you can disable a GMPLS LSP permanently. If the LSP being disabled is a nonpacket LSP, then the graceful LSP tear-down procedures that use the Admin Status object are used. If the LSP being disabled is a packet LSP, then the regular signaling procedures for LSP deletion are used.

To disable a GMPLS LSP, include the `disable` statement at any of the following hierarchy levels:

- `[edit protocols mpls label-switched-path` *`lsp-name`*`]`—Disable the LSP.
- `[edit protocols link-management te-link` *`te-link-name`*`]`—Disable a traffic engineering link.
- `[edit protocols link-management te-link` *`te-link-name`* `interface` *`interface-name`*`]`—Disable an interface used by a traffic engineering link.

Configuring the Graceful Deletion Timeout Interval

The router that initiates the graceful deletion procedure for an RSVP session waits for the graceful deletion timeout interval to ensure that all routers along the path (especially the ingress and egress routers) have prepared for the LSP to be taken down.

The ingress router initiates the graceful deletion procedure by sending the Admin Status object in the path message with the `D` bit set. The ingress router expects to receive an Resv message with the `D` bit set from the egress router. If the ingress router does not receive this message within the time specified by the graceful deletion timeout interval, it initiates a forced tear-down of the LSP by sending a PathTear message.

To configure the graceful deletion timeout interval, include the `graceful-deletion-timeout` statement at the `[edit protocols rsvp]` hierarchy level. You can configure a time between 1 through 300 seconds. The default value is 30 seconds.

```
graceful-deletion-timeout seconds;
```

You can configure this statement at the following hierarchy levels:

- `[edit protocols rsvp]`
- `[edit logical-systems` *`logical-system-name`* `protocols rsvp]`

You can use the `show rsvp version` command to determine the current value configured for the graceful deletion timeout.

GMPLS RSVP-TE VLAN LSP Signaling Overview

IN THIS SECTION

- Understanding GMPLS RSVP-TE Signaling | **2004**
- Need for GMPLS RSVP-TE VLAN LSP Signaling | **2004**
- GMPLS RSVP-TE VLAN LSP Signaling Functionality | **2006**
- LSP Hierarchy with GMPLS RSVP-TE VLAN LSP | **2007**
- Path Specification for GMPLS RSVP-TE VLAN LSP | **2007**
- GMPLS RSVP-TE VLAN LSP Configuration | **2007**
- Associated Bidirectional Packet LSP | **2009**
- Make-Before-Break for Associated Bidirectional Packet and GMPLS RSVP-TE VLAN LSP | **2009**
- Supported and Unsupported Features | **2010**

Understanding GMPLS RSVP-TE Signaling

Signaling is the process of exchanging messages within the control plane to set up, maintain, modify, and terminate data paths (label-switched paths (LSPs)) in the data plane. Generalized MPLS (GMPLS) is a protocol suite that extends the existing control plane of MPLS to manage further classes of interfaces and to support other forms of label switching, such as time-division multiplexing (TDM), fiber (port), Lambda, and so on.

GMPLS extends intelligent IP/MPLS connections from Layer 2 and Layer 3 all the way to Layer 1 optical devices. Unlike MPLS, which is supported mainly by routers and switches, GMPLS can also be supported by optical platforms, including SONET/SDH, optical cross-connects (OXCs), and dense wave division multiplexing (DWDM).

In addition to labels, which are primarily used to forward data in MPLS, other physical entries, such as wavelengths, time slots, and fibers can be used as label objects to forward data in GMPLS, thereby leveraging the existing control plane mechanisms to signal different kinds of LSPs. GMPLS uses RSVP-TE to be able to request the other label objects to signal the various kinds of LSPs (nonpacket). Bidirectional LSPs and an out-of-band control channel and a data channel using the Link Management Protocol (LMP) are the other mechanisms that are used by GMPLS to establish LSPs.

Need for GMPLS RSVP-TE VLAN LSP Signaling

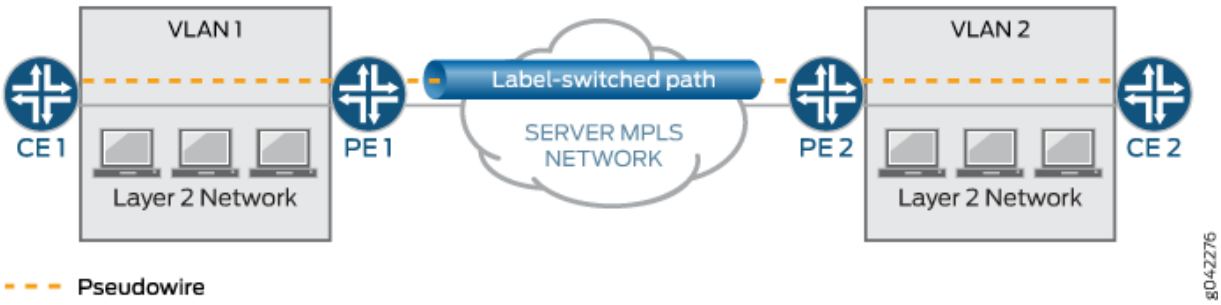
The traditional Layer 2 point-to-point services use Layer 2 circuits and Layer 2 VPN technologies that are based on LDP and BGP. In the traditional deployment, the customer edge (CE) devices do not

participate in the signaling of the Layer 2 service. The provider edge (PE) devices manage and provision the Layer 2 service to provide end-to-end connectivity between the CE devices.

One of the biggest challenges of having the PE devices provision the Layer 2 services for each Layer 2 circuit between a pair of CE devices is the network management burden on the provider network.

Figure 129 on page 2005 illustrates how the Layer 2 service is set up and used by the CE routers in a LDP/BGP-based Layer 2 VPN technology. Two CE routers CE1 and CE2 are connected to a provider MPLS network through the PE routers PE1 and PE2 respectively. The CE routers are connected to the PE routers by Ethernet links. Routers CE1 and CE2 are configured with VLAN1 and VLAN2 logical Layer 3 interfaces, so they appear to be directly connected. Routers PE1 and PE2 are configured with Layer 2 circuit (pseudowire) to carry the Layer 2 VLAN traffic between the CE routers. The PE routers use packet MPLS LSPs within the provider MPLS network to carry the Layer 2 VLAN traffic.

Figure 129: Traditional Layer 2 Point-to-Point Services



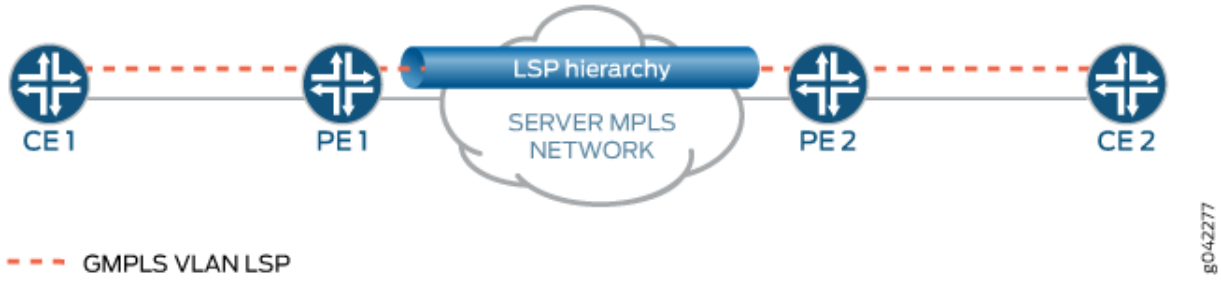
With the introduction of GMPLS-based VLAN LSP signaling, the need for the PE (also called server-layer) network to provision each individual Layer 2 connection between the CE (also called client) devices is minimized. The client router requests the server-layer router to which it is directly connected, for setting up the Layer 2 service to connect with a remote client router through GMPLS signaling.

The server-layer devices extend the signaling through the server-layer network to connect with the remote client routers. In the process, the server-layer device sets up the data plane for the Layer 2 service at the server-client border, and sets up the data plane for carrying the Layer 2 traffic within the server-layer network. With the Layer 2 service setup, the client routers can run IP/MPLS directly on top of the Layer 2 service and have IP/MPLS adjacency with each other.

In addition to reducing the provisioning activity needed on the server-layer devices, GMPLS signaling also provides the client routers with the flexibility of bringing up the Layer 2 circuits on an on-demand basis without depending on the server-layer administration for the provisioning of the Layer 2 service.

Using the same topology as in Figure 1, Figure 130 on page 2006 illustrates how the Layer 2 service is set up and used by the client routers in GMPL RSVP-TE-based Layer 2 VPN technology.

Figure 130: GMPLS RSVP-TE VLAN LSP



In Figure 130 on page 2006, instead of configuring a pseudowire to carry the Layer 2 VLAN traffic between the client routers, Routers PE1 and PE2 are configured with an IP-based communication channel and other GMPLS-specific configurations (identification of Ethernet links as TE-links) for allowing the exchange of GMPLS RSVP-TE signaling messages with the client routers. Routers CE1 and CE2 are also configured with an IP-based communication channel and relevant GMPLS configuration for exchanging the GMPLS RSVP-TE signaling messages with the server-layer routers. Routers CE1 and CE2 establish an IP/MPLS adjacency on top of this Layer 2 service.

GMPLS RSVP-TE VLAN LSP Signaling Functionality

Based on Figure 130 on page 2006, the client router establishes the Layer 2 service in the server-layer network as follows:

1. Router CE1 initiates GMPLS RSVP-TE signaling with Router PE1. In this signaling message, Router CE1 indicates the VLAN on the Ethernet link for which it needs the Layer 2 service and the remote CE router, Router CE2, with which the VLAN should be connected.

   Router CE1 also indicates the remote PE router, Router PE2, to which Router CE2 is connected, and the exact Ethernet link connecting Router CE2 to Router PE2 on which the Layer 2 service is required in the signaling message.

2. Router PE1 uses the information from Router CE1 in the signaling message and determines the remote PE router, Router PE2, with which Router CE2 is attached. Router PE1 then establishes a packet MPLS LSP (associated bidirectional) through the server-layer MPLS network for carrying the VLAN traffic and then passes the GMPLS RSVP-TE signaling message to Router PE2 using the LSP hierarchy mechanism.

3. Router PE2 propagates the GMPLS RSVP-TE signaling message to Router CE2 with the VLAN to be used on the PE2-CE2 Ethernet link.

4. Router CE2 responds with an acknowledgment to the GMPLS RSVP-TE signaling message to Router PE2. Router PE2 then propagates it to Router PE1, which in turn propagates it to Router CE1.

5. As part of this message propagation, Routers PE1 and PE2 set up the forwarding plane to enable bidirectional flow of VLAN Layer 2 traffic between Routers CE1 and CE2.

LSP Hierarchy with GMPLS RSVP-TE VLAN LSP

The Layer 2 service in GMPLS RSVP-TE VLAN LSP signaling is brought up using a hierarchy mechanism in which two different RSVP LSPs are created for the Layer 2 service:

- An end-to-end VLAN LSP that has state information at the client and server-layer routers.
- An associated bidirectional packet transport LSP that is present in the server-layer routers (PE and P) of the server-layer network.

The LSP hierarchy avoids sharing information about technology-specific LSP characteristics with the core nodes of the server-layer network. This solution cleanly separates the VLAN LSP state and the transport LSP state, and ensures that the VLAN LSP state is only present on the nodes (PE, CE) where it is needed.

Path Specification for GMPLS RSVP-TE VLAN LSP

The path for the GMPLS RSVP-TE LSP is configured as an Explicit Route Object (ERO) at the initiating client router. As this LSP traverses different network domains (initiating, terminating at client network, and traversing the server-layer network), the LSP setup falls under the category of an interdomain LSP setup. In an interdomain scenario, one network domain generally does not have full visibility into the topology of the other network domain. Hence, the ERO that gets configured at the initiating client router does not have full hop information for the server-layer portion. This feature requires that the ERO configured at the CE router has three hops, with the first hop being a strict hop identifying the CE1-PE1 Ethernet link, the second hop being a loose hop identifying the egress PE router (PE2), and the third hop being a strict hop identifying the CE2-PE2 Ethernet link.

GMPLS RSVP-TE VLAN LSP Configuration

The configuration required to set up a GMPLS VLAN LSP at the client and server routers uses the existing GMPLS configuration model with some extensions. The Junos OS GMPLS configuration model for nonpacket LSPs is targeted toward bringing the physical interfaces up and running through GMPLS RSVP-TE signaling, whereas signaling a GMPLS RSVP-TE VLAN LSP aims at bringing up individual VLANs on top of a physical interface. The `ethernet-vlan` configuration statement under the `[edit protocols link-management te-link]` hierarchy enables this.

The client router has physical interfaces connected to a server network, and the server network provides a point-to-point connection between two client routers over the attached physical interfaces. The physical interface is brought into an operational state by GMPLS RSVP-TE as follows:

1. The client router maintains a routing or signaling adjacency with the server network node to which the physical interface is connected, typically through a control channel different from the physical interface, because the physical interface itself is brought up and running only after the signaling.

2. The client router and the server network node identify the physical interfaces connecting them using the TE-link mechanism.

3. The client router and the server network node use the TE-link identifier (IP address) as the GMPLS RSVP hop and the physical interface identifier as the GMPLS label values in the GMPLS RSVP-TE signaling messages to bring the physical interface into an operational state.

In the existing GMPLS configuration, the server and client network nodes use the `protocols link-management peer` *`peer-name`* configuration statement to specify the adjacent peer node. Because a client router can have one or more physical interfaces connected to the server network node, these physical interfaces are grouped and identified by an IP address through the `protocols link-management te-link` *`link-name`* configuration statement. The TE-link is assigned a local IP address, a remote IP address, and a list of physical interfaces. The TE-link is then associated with the `protocols link-management peer` *`peer-name`* `te-link` *`te-link-list`* configuration statement.

The out-of-band control channel that is required for exchanging signaling messages is specified using the `protocols link-management peer` *`peer-name`* `control-channel` *`interface-name`* configuration statement. The existence of the server or client network node is made visible to the RSVP and IGP (OSPF) protocols through the `peer-interface` *`interface-name`* configuration statement under the `[edit protocols rsvp]` and `[edit protocols ospf]` hierarchy levels.

In the existing GMPLS configuration, the label (upstream label and resv label) that is carried in the signaling message is an integer identifier that identifies the physical interface that is required to be brought up. As the label is used to identify the physical interface, the existing GMPLS configuration allows multiple interfaces to be grouped under a single TE-link. In the existing GMPLS configuration, there is sufficient information in the GMPLS RSVP-TE signaling message, such as TE-link address and label value, to identify the physical interface that is required to be brought up. In contrast, for GMPLS RSVP-TE VLAN LSP configuration, the VLAN ID value is used as the label in the signaling message.

In the GMPLS RSVP-TE VLAN LSP configuration, if multiple interfaces are allowed to be configured under a single TE-link, using VLAN ID as the label value in the signaling message can cause ambiguity as to which physical interface on which the VLAN has to be provisioned. Therefore, the TE-link is configured with the `ethernet-vlan` configuration statement, if the number of physical interfaces that can be configured under the TE-link is restricted to only one.

In the existing GMPLS configuration, the bandwidth for a nonpacket LSP is a discrete quantity that corresponds to the bandwidth of the physical interface that needs to be brought up. So, the GMPLS LSP configuration does not allow any bandwidth to be specified, but allows the bandwidth to be specified only through the `signal-bandwidth` configuration statement under the `[protocols mpls label-switched-path` *`lsp-name`* `lsp-attributes]` hierarchy level. In the GMPLS VLAN LSP configuration, bandwidth is specified similar

to that of a packet LSP. In the GMPLS VLAN LSP configuration, the `bandwidth` option is supported and `signal-bandwidth` is not supported.

Associated Bidirectional Packet LSP

The GMPLS RSVP-TE VLAN LSP is carried on an associated bidirectional transport LSP within the server-layer network, which is a single-sided provisioned LSP. The transport LSP signaling is initiated as a unidirectional LSP from the source router to the destination router in the forward direction, and the destination router in turn initiates the signaling of the unidirectional LSP in the reverse direction back to the source router.

Make-Before-Break for Associated Bidirectional Packet and GMPLS RSVP-TE VLAN LSP

The make-before-break support for an associated bidirectional transport LSP follows a similar model, where the destination router for the forward direction of the bidirectional LSP does not perform any make-before-break operations on the reverse direction of the bidirectional LSP. It is the source router (initiator of the associated bidirectional LSP) that initiates the make-before-break newer instance of the associated bidirectional LSP, and the destination router in turn initiates the make-before-break newer instance in the other direction.

For instance, in Figure 130 on page 2006, the unidirectional transport LSP is initiated from Router PE1 to Router PE2 in the forwarding direction, and in turn Router PE2 initiates the transport LSP to Router PE1 in the reverse direction. When a make-before-break instance occurs, only Router PE1 as the initiating client router can establish a new instance of the associated bidirectional LSP. Router PE2 in turn initiates the make-before-break newer instance in the reverse direction.

The make-before-break support for the associated bidirectional transport LSP is used only in scenarios where the transport LSP gets into a state of being locally protected due to link or node failure on the path of the LSP. The GMPLS RSVP-TE VLAN LSP uses the make-before-break mechanism for adjusting seamless bandwidth changes.

NOTE: Periodic re-optimization is not enabled for the associated bidirectional transport LSPs.

The newer make-before-break instance of the GMPLS VLAN LSP is supported under the following constraints:

- It should originate from the same client router as the older instance and be destined to the same client router as the older instance.
- It should use the same server-client links at both the server-client ends as the older instance.
- It should use the same VLAN label at the server-client links as the older instance.

- The GMPLS VLAN LSP should be configured as `adaptive` when the bandwidth change is initiated from the CLI, or else the current instance of the VLAN LSP is torn down and a new VLAN LSP instance is established.

The make-before-break operation for the GMPLS VLAN LSP on the server-layer edge router is rejected if these constraints are not met.

On the server-layer edge routers, when a make-before-break instance of the GMPLS VLAN LSP is seen, a completely new, separate associated bidirectional transport LSP is created to support this make-before-break instance. The existing associated bidirectional LSP (supporting the older instance) is not triggered to initiate a make-before-break instance at the transport LSP level. An implication of this choice (of initiating a new transport LSP) is that at the server-layer resource/bandwidth sharing does not happen when a make-before-break operation is performed for the GMPLS VLAN LSP.

Supported and Unsupported Features

Junos OS supports the following features with the GMPLS RSVP-TE VLAN LSP:

- Request for specific bandwidth and local protection for the VLAN LSP on the client router to the server-layer router.
- Nonstop active routing (NSR) support for the GMPLS VLAN LSP at the client routers, server-layer edge routers, and associated bidirectional transport LSP at the server-layer edge routers.
- Multichassis support.

Junos OS does **not** support the following GMPLS RSVP-TE VLAN LSP functionality:

- Graceful restart support for associated bidirectional packet LSP and GMPLS VLAN LSP.
- End-to-end path computation for GMPLS VLAN LSP using CSPF algorithm at the client router.
- Non-CSPF routing-based discovery of next-hop routers by the different client, server-layer edge routers.
- Automatic provisioning of the client Layer 3 VLAN interfaces upon the successful setup of the VLAN LSP at the client routers.
- MPLS OAM (LSP-ping, BFD).
- Packet MPLS applications, such as next-hop in static route and in IGP shortcuts.
- Local cross connect mechanism, where a client router connects to a remote client router which is connected to the same server router.
- Junos OS Services Framework.
- IPv6 support.

- Logical systems.
- Aggregated Ethernet/SONET/IRB interfaces at the server-client link.

Example: Configuring GMPLS RSVP-TE VLAN LSP Signaling

IN THIS SECTION

- Requirements | **2011**
- Overview | **2012**
- Configuration | **2018**
- Verification | **2034**

This example shows how to configure GMPLS RSVP-TE VLAN LSP signaling on the client routers to enable one client router to connect with a remote client router through a server-layer network using the LSP hierarchy. This enables the client routers to establish, maintain, and provision the Layer 2 services, without depending on the server-layer administration, thereby reducing the burden on the operational expenses of the provider network.

Requirements

This example uses the following hardware and software components:

- Six routers that can be a combination of M Series Multiservice Edge Routers, MX Series 5G Universal Routing Platforms, T Series Core Routers, and PTX Series Packet Transport Routers
- Junos OS Release 14.2 or later running on the client routers and server-layer edge routers

Before you begin:

1. Configure the device interfaces.
2. Configure the interface-associated VLANs.
3. Configure the following routing protocols:
   - RSVP
   - MPLS
   - LMP

Overview

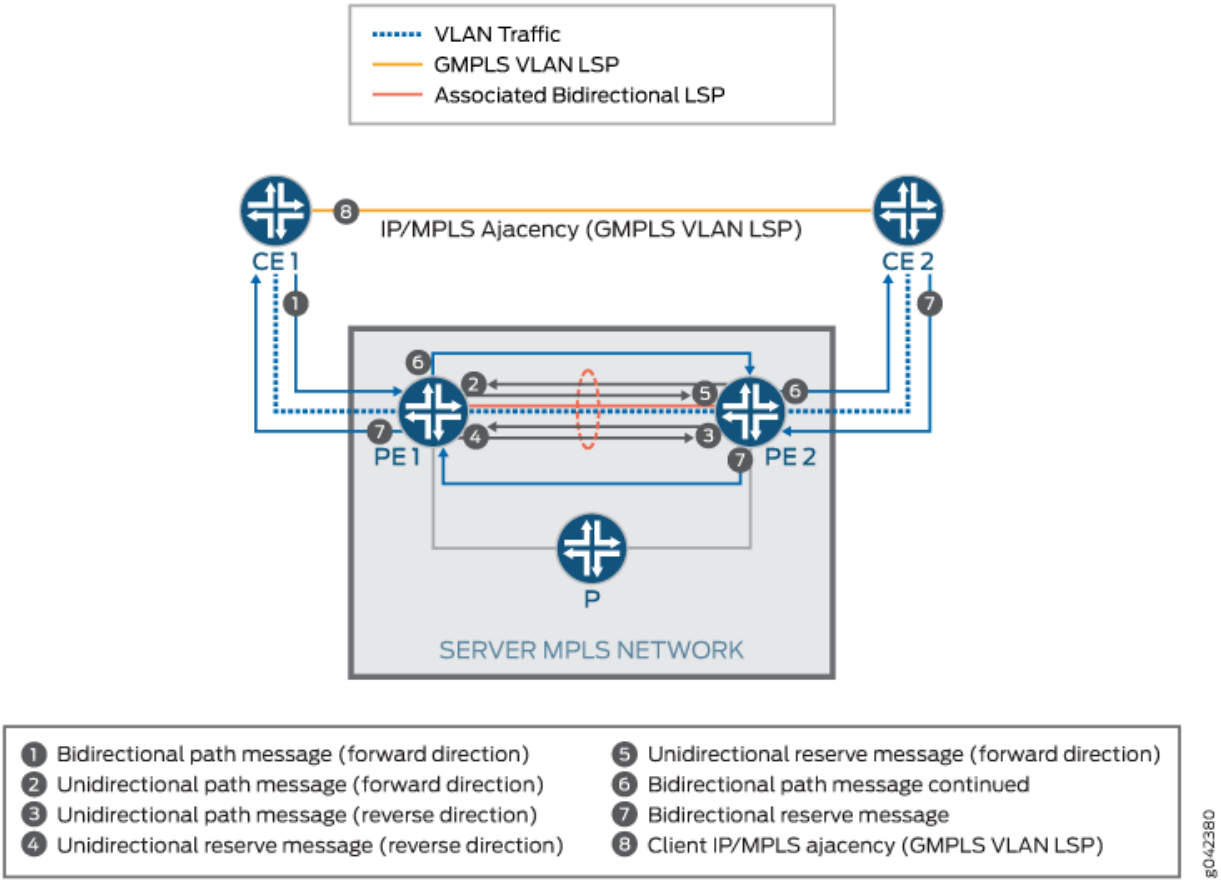
IN THIS SECTION

- Topology | **2018**

Starting with Junos OS Release 14.2, the Layer 2 services between two client routers across an external/third-party server-layer network are set up by the client routers on an on-demand basis through GMPLS RSVP-TE signaling. This feature provides the client routers the flexibility to establish, maintain, and provision the Layer 2 services, without depending on the server-layer administration, thereby reducing the burden on the operational expenses of the provider network. In traditional Layer 2 VPN technology based on LDP and BGP, the provider network handled the provisioning activity for each Layer 2 circuit established between two client routers.

Figure 131 on page 2013 illustrates the setting up and signaling of the GMPLS VLAN LSP between two client routers, CE1 and CE2, across a server-layer network with two server-layer edge routers, PE1 and PE2, and one server-layer core router, P.

Figure 131: Setting Up a GMPLS VLAN LSP



The signaling of GMPLS VLAN LSP is executed as follows:

1. `Initiating GMPLS VLAN LSP at CE1`

   Router CE1 initiates the GMPLS VLAN LSP setup by sending the GMPLS RSVP-TE path message to Router PE1. The signaling between CE1 and PE1 is over an out-of-band control channel, which is a separate control VLAN configured on the Ethernet link connecting the two routers.

   The GMPLS RSVP-TE path message initiated by Router CE1 is used to perform the following:

   a. Identify the Ethernet link on which the VLAN is active.

   b. Abstract the Ethernet link as a TE-link and assign an IP address to identify the Ethernet link.

   c. Allocate a VLAN ID from the pool of free VLANs managed by Router CE1 for every Ethernet link connecting Router PE1 to the identified Ethernet link.

      This VLAN ID can also be used for the GMPLS VLAN LSP at the CE2-PE2 Ethernet link.

d. Identify the VLAN for which the Layer 2 service is required to be set up using the allocated VLAN ID as the upstream label object and the upstream direction label value.

e. Include an ERO object that helps Router PE1 in establishing the VLAN LSP through the server-layer network to the remote client router, CE2. The ERO object in the path message includes three hops:

- First hop—Strict hop identifying the initiating client-server Ethernet link, PE1-CE1.
- Second hop—Loose hop identifying the remote server-layer router, PE2.
- Third hop—Strict hop identifying the remote clinet-server Ethernet link, PE2-CE2.

f. Include the bandwidth required for the GMPLS VLAN LSP.

g. Include any local-protection required within the server-layer network for the VLAN LSP.

2. **Initiating Associated Bidirectional Transport LSP at PE1**

After Router PE1 receives the path message from Router CE1, the message is validated to check the availability of the Ethernet link and VLAN ID. In the server-layer network, the Layer 2 services between the server-layer routers, PE1 and PE2, are provided at the data plane in a manner similar to Layer 2 circuits. Router PE1 brings up a transport LSP to Router PE2 and then extends the GMPLS VLAN LSP as a hierarchical LSP running on top of the PE1-PE2 transport LSP. The PE1-PE2 transport LSP is a packet LSP and is bidirectional in nature. This is because the GMPLS VLAN LSP is bidirectional and each server-layer router needs to be able to do the following:

- Receive traffic from the server-client Ethernet link (for example, the PE1-CE1 link) and send it to the remote server-layer router, PE2.
- Receive traffic from remote Router PE2 and send it on the PE1-CE1 Ethernet link.

For each GMPLS VLAN LSP, a packet transport LSP is set up within the server-layer network. The transport LSP is exclusively used to carry traffic of the GMPLS VLAN LSP for which it was created. The transport LSP is dynamically created at the time of receiving the GMPLS VLAN LSP; thus, no configuration is required to trigger its creation. The transport LSP established for the VLAN LSP inherits the bandwidth and the local-protection attributes from the VLAN LSP.

Router PE1 signals the PE1-PE2 transport LSP to Router PE2. Router PE1 determines the destination for the transport LSP from the loose hop specified in the ERO object of the GMPLS RSVP-TE path message from Router CE1 and then signals the VLAN LSP. However, if the PE1-PE2 transport LSP fails to establish, Router PE1 sends back a path error message to Router CE1, and the GMPLS VLAN LSP is not established as well.

3. **Setting Up the Associated Bidirectional Transport LSP Between the Server-Layer Routers**

The associated bidirectional LSP between routers PE1 and PE2 consists of two unidirectional packet LSPs:

- PE1-to-PE2
- PE2-to-PE1

Router PE1 initiates signaling of a unidirectional packet LSP to Router PE2. This unidirectional packet LSP constitutes the forward direction (PE1-to-PE2) of the associated bidirectional LSP, and the path message carries the Extended Association Object indicating this is a single-sided provisioning model. On receiving the path message for the LSP, Router PE2 responds with a Resv message and triggers the signaling of a unidirectional packet LSP to Router PE1 with the same path as (PE1-to-PE2) in the reverse direction. This unidirectional packet LSP uses the PE2-to-PE1 direction of the associated bidirectional LSP, and this path message carries the same Extended Association Object seen in the PE1-to-PE2 path message.

When Router PE1 receives the Resv message for the PE1-to-PE2 unidirectional LSP and the path message for the PE2-to-PE1 unidirectional LSP, PE1 binds the PE1-to-PE2 and PE2-to-PE1 unidirectional LSPs by matching the Extended Association Objects carried in the respective path messages. For the path message for the PE2-to-PE1 unidirectional LSP, Router PE1 responds with the Resv Message. On receiving the Resv message for the PE1-to-PE2 LSP and the path message for the PE2-to-PE1 LSP, Router PE1 has established the associated bidirectional packet transport LSP.

4. **Setting Up the GMPLS VLAN LSP at Router PE1**

After successfully establishing the transport LSP, Router PE1 triggers the signaling of the GMPLS VLAN LSP. Router PE1 sends the GMPLS RSVP-TE path message corresponding to the VLAN LSP directly to Router PE2, which is bidirectional in nature and includes the upstream label object.

Router PE2 is not aware of the association between the transport LSP and the VLAN LSP. This association is indicated to Router PE2 by Router PE1.

5. **Setting Up the GMPLS VLAN LSP at Router PE2**

On receiving the VLAN LSP path message from Router PE1, Router PE2 verifies the availability of the transport LSP. If the transport LSP is not available or the LSP setup is in progress, the VLAN LSP processing is put on hold. When the transport LSP is available, Router PE2 processes the VLAN LSP path message. The ERO object in this path message indicates that the next hop is a strict hop identifying the PE2-to-CE2 Ethernet link. The ERO object can indicate the VLAN ID to be used on the PE2-to-CE2 Ethernet link by Router PE2.

Router PE2 appropriately allocates the VLAN ID to be sent as the upstream label in the VLAN LSP path message to Router CE2, and sends it through an out-of-band control channel.

6. **Processing the GMPLS VLAN LSP at Router CE2**

On receiving the GMPLS RSVP-TE LSP from Router PE2, Router CE2 validates the availability of VLAN ID for allocation on the PE2-to-CE2 link. Router CE2 then allocates the VLAN ID for this

VLAN LSP and sends back a Resv message to Router PE2 with the VLAN ID as the label object in the Resv message.

7. **Processing the GMPLS VLAN LSP at Router PE2**

   On receiving the Resv message from Router CE2, Router PE2 validates that the label object in the Resv message has the same VLAN ID as in the path message. Router PE2 then allocates a 20-bit MPLS label, which is included in the Resv message sent to Router PE1.

   Router PE2 then programs the forwarding plane with the entries to provide the Layer 2 service functionality.

   NOTE: For all the VLAN IDs that can be allocated as labels on the PE1-to-CE1 and PE2-CE2 Ethernet links, you must manually configure logical interfaces for circuit cross-connect (CCC) purposes on the server-layer edge routers and not for other families, such as IPv4, IPv6, or MPLS.

8. **Processing the GMPLS VLAN LSP at Router PE1**

   On receiving the Resv message for the VLAN LSP from Router PE2, Router PE1 sends a Resv message to Router CE1 with the same VLAN ID it received as the upstream label from Router CE1. Router PE1 programs the forwarding plane with the entries to provide the Layer 2 service functionality as Router PE2.

9. **Processing the GMPLS VLAN LSP at Router CE1**

   On receiving the Resv message from Router PE1, Router CE1 validates that the VLAN ID received in the Resv message matches the VLAN ID in the upstream label in the path message it sent. This completes the setup of the GMPLS VLAN LSP from Router CE1 to Router CE2.

   NOTE:

   - The GMPLS VLAN LSP setup does not result in the addition of any forwarding plane entries at the client routers, CE1 and CE2. Only the server-layer routers, PE1 and PE2, add the forwarding plane entries for the GMPLS VLAN LSP.
   - There is no routing information exchange between the client and the server-layer routers. The client and server-layer routers do not exchange their network topology information with each other.

10. **Accounting for Bandwidth of the GMPLS VLAN LSP**

    On successfully setting up the GMPLS VLAN LSP, both the client and server-layer routers reduce the amount of available bandwidth on the server-client Ethernet links by the bandwidth amount

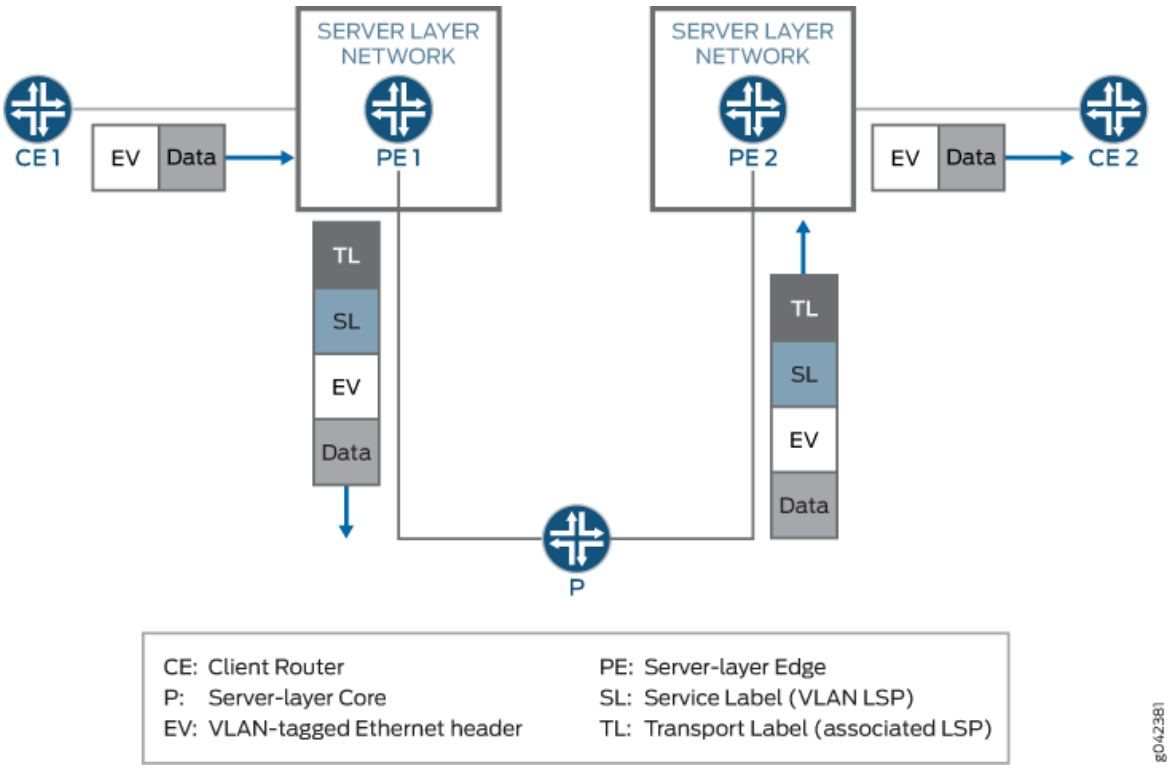
allocated for the GMPLS VLAN LSP. This bandwidth accounting information is used for admission control purposes when additional GMPLS VLAN LSPs are brought up on the server-client Ethernet links.

11. **Using GMPLS VLAN LSP by the Client Routers**

After successfully setting up the GMPLS VLAN LSP, the client routers – CE1 and CE2 – need to be manually configured with the VLAN logical interface on top of the server-client Ethernet links with the signaled VLAN ID. This logical interface needs to be configured with the IP address and needs to be included in the IGP protocol. As a result of this configuration, Routers CE1 and CE2 establish IGP adjacency and exchange data traffic over the Layer 2 service established through the GMPLS signaling.

Figure 132 on page 2017 illustrates the data traffic flow of the GMPLS VLAN LSP from Router CE1 to Router CE2 after the LSP setup is complete and the necessary CE1-to-CE2 IGP/MPLS adjacency has been established. The server-layer transport LSP originates from Router PE1, traverses a single server-layer core router, Router P, and reaches Router PE2. The server-layer transport LSP is shown as a penultimate-hop pop LSP, where Router P pops off the transport LSP label, and only the service label is present on the P-to-PE2 link.

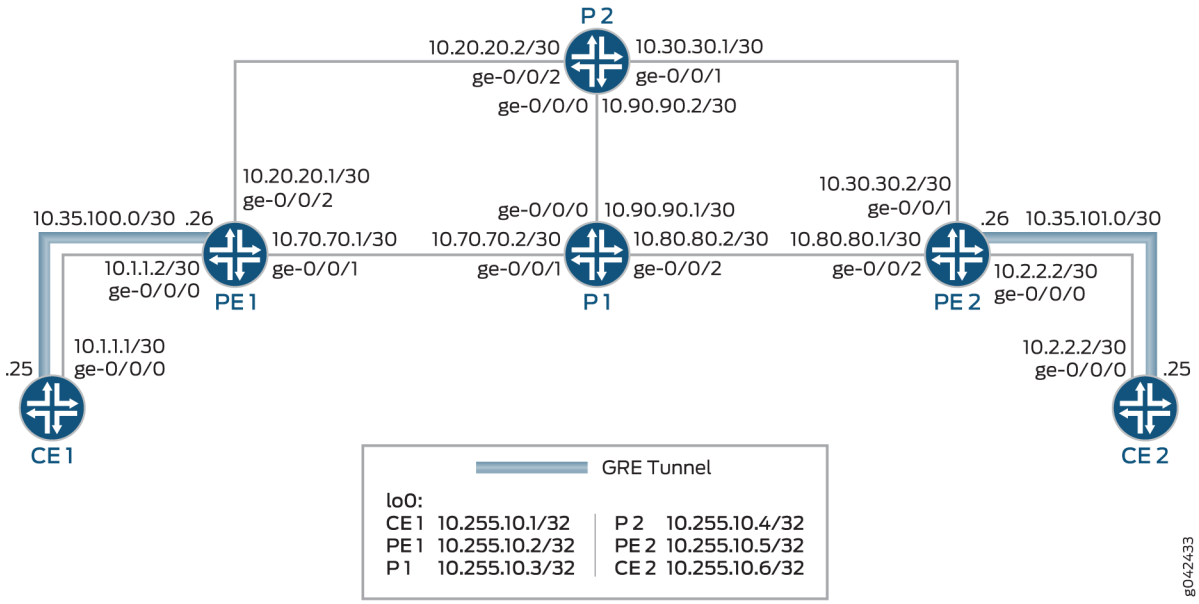
Figure 132: Data Traffic Flow of GMPLS VLAN LSP



Topology

In Figure 133 on page 2018, GMPLS RSVP-TE VLAN LSP signaling is used to establish the Layer 2 services between the client routers, Router CE1 and Router CE2. The server routers, Router PE1 and Router PE2, have a GRE tunnel established with each of the directly connected client routers. Routers P1 and P2 are also server routers in the server-layer network.

Figure 133: Configuring GMPLS RSVP-TE VLAN LSP Signaling



Configuration

IN THIS SECTION

- CLI Quick Configuration | **2019**
- Configuring the Client Router | **2023**
- Configuring the Server Router | **2028**

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

CE1

```
set interfaces ge-0/0/0 vlan-tagging
set interfaces ge-0/0/0 unit 1 vlan-id 1
set interfaces ge-0/0/0 unit 1 family inet address 10.1.1.1/30
set interfaces ge-0/0/0 unit 1 family mpls
set interfaces ge-0/0/0 unit 10 vlan-id 10
set interfaces ge-0/0/0 unit 10 family inet address 10.10.10.1/24
set interfaces ge-0/0/0 unit 10 family mpls
set interfaces gre unit 0 tunnel source 10.1.1.1
set interfaces gre unit 0 tunnel destination 10.1.1.2
set interfaces gre unit 0 family inet address 10.35.100.25/30
set interfaces lo0 unit 0 family inet address 10.255.10.1/32
set routing-options router-id 10.255.10.1
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols rsvp peer-interface PE1
set protocols mpls no-cspf
set protocols mpls label-switched-path CE1-to-CE2 from 10.255.10.1
set protocols mpls label-switched-path CE1-to-CE2 to 10.255.10.6
set protocols mpls label-switched-path CE1-to-CE2 lsp-attributes switching-type ethernet-vlan
set protocols mpls label-switched-path CE1-to-CE2 lsp-attributes upstream-label vlan-id 10
set protocols mpls label-switched-path CE1-to-CE2 bandwidth 100m
set protocols mpls label-switched-path CE1-to-CE2 primary path1
set protocols mpls path path1 10.35.1.2 strict
set protocols mpls path path1 10.255.10.5 loose
set protocols mpls path path1 10.36.1.1 strict
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols link-management te-link link10 local-address 10.35.1.1
set protocols link-management te-link link10 remote-address 10.35.1.2
set protocols link-management te-link link10 ethernet-vlan
set protocols link-management te-link link10 interface ge-0/0/0
set protocols link-management peer PE1 address 10.255.10.2
set protocols link-management peer PE1 control-channel gre.0
set protocols link-management peer PE1 te-link link10
```

PE1

```
set interfaces ge-0/0/0 vlan-tagging
set interfaces ge-0/0/0 encapsulation flexible-ethernet-services
set interfaces ge-0/0/0 unit 1 vlan-id 1
set interfaces ge-0/0/0 unit 1 family inet address 10.1.1.2/30
set interfaces ge-0/0/0 unit 1 family mpls
set interfaces ge-0/0/0 unit 10 encapsulation vlan-ccc
set interfaces ge-0/0/0 unit 10 vlan-id 10
set interfaces ge-0/0/1 unit 0 family inet address 10.70.70.1/30
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 10.20.20.1/30
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces gre unit 0 tunnel source 10.1.1.2
set interfaces gre unit 0 tunnel destination 10.1.1.1
set interfaces gre unit 0 family inet address 10.35.100.26/30
set interfaces lo0 unit 0 family inet address 10.255.10.2/32
set routing-options router-id 10.255.10.2
set protocols rsvp associated-bidirectional-lsp single-sided-provisioning
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols rsvp peer-interface CE1 dynamic-bidirectional-transport
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols link-management te-link link1 local-address 10.35.1.2
set protocols link-management te-link link1 remote-address 10.35.1.1
set protocols link-management te-link link1 ethernet-vlan vlan-id-range 1-1000
set protocols link-management te-link link1 interface ge-0/0/0
set protocols link-management peer CE1 address 10.255.10.1
set protocols link-management peer CE1 control-channel gre.0
set protocols link-management peer CE1 te-link link1
```

P1

```
set interfaces ge-0/0/0 unit 0 family inet address 10.90.90.1/24
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 10.70.70.2/24
set interfaces ge-0/0/1 unit 0 family mpls
```

```
set interfaces ge-0/0/2 unit 0 family inet address 10.80.80.2/24
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.10.3/32
set routing-options router-id 10.255.10.3
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
```

P2

```
set interfaces ge-0/0/0 unit 0 family inet address 10.90.90.2/30
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 10.30.30.1/30
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 10.20.20.2/30
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.10.4/32
set routing-options router-id 10.255.10.4
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
```

PE2

```
set interfaces ge-0/0/0 vlan-tagging
set interfaces ge-0/0/0 encapsulation flexible-ethernet-services
set interfaces ge-0/0/0 unit 0 vlan-id 1
set interfaces ge-0/0/0 unit 0 family inet address 10.2.2.2/30
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/0 unit 10 encapsulation vlan-ccc
set interfaces ge-0/0/0 unit 10 vlan-id 10
set interfaces ge-0/0/1 unit 0 family inet address 10.30.30.2/30
set interfaces ge-0/0/1 unit 0 family mpls
```

```
set interfaces ge-0/0/2 unit 0 family inet address 10.80.80.1/30
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces gre unit 0 tunnel source 10.2.2.2
set interfaces gre unit 0 tunnel destination 10.2.2.1
set interfaces gre unit 0 family inet address 10.35.101.26/30
set interfaces lo0 unit 0 family inet address 10.255.10.5/32
set routing-options router-id 10.255.10.5
set protocols rsvp associated-bidirectional-lsp single-sided-provisioning
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols rsvp peer-interface CE2 dynamic-bidirectional-transport
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols link-management te-link link1 local-address 10.36.1.2
set protocols link-management te-link link1 remote-address 10.36.1.1
set protocols link-management te-link link1 ethernet-vlan vlan-id-range 1-1000
set protocols link-management te-link link1 interface ge-0/0/0
set protocols link-management peer CE2 address 10.255.10.6
set protocols link-management peer CE2 control-channel gre.0
set protocols link-management peer CE2 te-link link1
```

CE2

```
set interfaces ge-0/0/0 vlan-tagging
set interfaces ge-0/0/0 unit 1 vlan-id 1
set interfaces ge-0/0/0 unit 1 family inet address 10.2.2.1/24
set interfaces ge-0/0/0 unit 1 family mpls
set interfaces ge-0/0/0 unit 10 vlan-id 10
set interfaces ge-0/0/0 unit 10 family inet address 10.10.10.2/24
set interfaces ge-0/0/0 unit 10 family mpls
set interfaces gre unit 0 tunnel source 10.2.2.1
set interfaces gre unit 0 tunnel destination 10.2.2.2
set interfaces gre unit 0 family inet address 10.35.101.25/30
set interfaces lo0 unit 0 family inet address 10.255.10.6/32
set routing-options router-id 10.255.10.6
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols rsvp peer-interface PE2
set protocols mpls interface all
```

```
set protocols mpls interface fxp0.0 disable
set protocols link-management te-link link10 local-address 10.36.1.1
set protocols link-management te-link link10 remote-address 10.36.1.2
set protocols link-management te-link link10 ethernet-vlan vlan-id-range 1-1000
set protocols link-management te-link link10 interface ge-0/0/0
set protocols link-management peer PE2 address 10.255.10.5
set protocols link-management peer PE2 control-channel gre.0
set protocols link-management peer PE2 te-link link10
```

Configuring the Client Router

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure Router CE1:

NOTE: Repeat this procedure for Router CE2 in the server-layer network, after modifying the appropriate interface names, addresses, and any other parameters for the router.

1. Configure the interface connecting Router CE1 to Router PE1.

```
[edit interfaces]
user@CE1# set ge-0/0/0 vlan-tagging
```

2. Configure the control VLAN for the ge-0/0/0 interface.

```
[edit interfaces]
user@CE1# set ge-0/0/0 unit 1 vlan-id 1
user@CE1# set ge-0/0/0 unit 1 family inet address 10.1.1.1/30
user@CE1# set ge-0/0/0 unit 1 family mpls
```

3. Configure the LSP VLAN on the ge-0/0/0 interface.

```
[edit interfaces]
user@CE1# set ge-0/0/0 unit 10 vlan-id 10
```

```
user@CE1# set ge-0/0/0 unit 10 family inet address 10.10.10.1/24
user@CE1# set ge-0/0/0 unit 10 family mpls
```

4. Configure the GRE tunnel as the controlling interface for Router CE1.

```
[edit interfaces]
user@CE1# set gre unit 0 tunnel source 10.1.1.1
user@CE1# set gre unit 0 tunnel destination 10.1.1.2
user@CE1# set gre unit 0 family inet address 10.35.100.25/30
```

5. Configure the loopback interface of Router CE1.

```
[edit interfaces]
user@CE1# set lo0 unit 0 family inet address 10.255.10.1/32
```

6. Configure the loopback address of Router CE1 as its router ID.

```
[edit routing-options]
user@CE1# set router-id 10.255.10.1
```

7. Enable RSVP on all the interfaces of Router CE1, excluding the management interface.

```
[edit protocols]
user@CE1# set rsvp interface all
user@CE1# set rsvp interface fxp0.0 disable
```

8. Configure the RSVP peer interface for Router CE1.

```
[edit protocols]
user@CE1# set rsvp peer-interface PE1
```

9. Disable automatic path computation for label-switched paths (LSPs).

```
[edit protocols]
user@CE1# set mpls no-cspf
```

10. Configure the LSP to connect Router CE1 to Router CE2.

```
[edit protocols]
user@CE1# set mpls label-switched-path CE1-to-CE2 from 10.255.10.1
user@CE1# set mpls label-switched-path CE1-to-CE2 to 10.255.10.6
```

11. Configure the CE1-to-CE2 LSP attributes.

```
[edit protocols]
user@CE1# set mpls label-switched-path CE1-to-CE2 lsp-attributes switching-type ethernet-
vlan
user@CE1# set mpls label-switched-path CE1-to-CE2 lsp-attributes upstream-label vlan-id 10
user@CE1# set mpls label-switched-path CE1-to-CE2 bandwidth 100m
```

12. Configure the CE1-to-CE2 LSP path and path parameters.

```
[edit protocols]
user@CE1# set mpls label-switched-path CE1-to-CE2 primary path1
user@CE1# set mpls path path1 10.35.1.2 strict
user@CE1# set mpls path path1 10.255.10.5 loose
user@CE1# set mpls path path1 10.36.1.1 strict
```

13. Enable MPLS on all the interfaces of Router CE1, excluding the management interface.

```
[edit protocols]
user@CE1# set mpls interface all
user@CE1# set mpls interface fxp0.0 disable
```

14. Configure a traffic engineering link, and assign addresses for the local and remote end of the link.

```
[edit protocols]
user@CE1# set link-management te-link link10 local-address 10.35.1.1
user@CE1# set link-management te-link link10 remote-address 10.35.1.2
```

15. Enable setting up of Layer 2 VLAN LSP on the link10 traffic engineering link.

```
[edit protocols]
user@CE1# set link-management te-link link10 ethernet-vlan
```

16. Configure the Router CE1 interface as the member interface of the link10 traffic engineering link.

```
[edit protocols]
user@CE1# set link-management te-link link10 interface ge-0/0/0
```

17. Configure Router PE1 as the Link Management Protocol (LMP) peer for Router CE1, and configure the peer attributes.

```
[edit protocols]
user@CE1# set link-management peer PE1 address 10.255.10.2
user@CE1# set link-management peer PE1 control-channel gre.0
user@CE1# set link-management peer PE1 te-link link10
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@CE1# show interfaces
ge-0/0/0 {
    vlan-tagging;
    unit 1 {
        vlan-id 1;
        family inet {
            address 10.1.1.1/30;
        }
        family mpls;
    }
    unit 10 {
        vlan-id 10;
        family inet {
            address 10.10.10.1/24;
```

```
        }
        family mpls;
    }
}
gre {
    unit 0 {
        tunnel {
            source 10.1.1.1;
            destination 10.1.1.2;
        }
        family inet {
            address 10.35.100.25/30;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.255.10.1/32;
        }
    }
}
```

```
user@CE1# show routing-options
router-id 10.255.10.1;
```

```
user@CE1# show protocols
rsvp {
    interface all;
    interface fxp0.0 {
        disable;
    }
    peer-interface PE1;
}
mpls {
    no-cspf;
    label-switched-path CE1-to-CE2 {
        from 10.255.10.1;
        to 10.255.10.6;
        lsp-attributes {
```

```
            switching-type ethernet-vlan;
            upstream-label {
                vlan-id 10;
            }
        }
        bandwidth 100m;
        primary path1;
    }
    path path1 {
        10.35.1.2 strict;
        10.255.10.5 loose;
        10.36.1.1 strict;
    }
    interface all;
    interface fxp0.0 {
        disable;
    }
}
link-management {
    te-link link10 {
        local-address 10.35.1.1;
        remote-address 10.35.1.2;
        ethernet-vlan;
        interface ge-0/0/0;
    }
    peer PE1 {
        address 10.255.10.2;
        control-channel gre.0;
        te-link link10;
    }
}
```

Configuring the Server Router

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure Router PE1:

NOTE: Repeat this procedure for Router PE2 in the server-layer network, after modifying the appropriate interface names, addresses, and any other parameters for the router.

1. Configure the interface connecting Router PE1 to Router CE1.

```
[edit interfaces]
user@PE1# set ge-0/0/0 vlan-tagging
user@PE1# set ge-0/0/0 encapsulation flexible-ethernet-services
```

2. Configure the control VLAN for the ge-0/0/0 interface.

```
[edit interfaces]
user@PE1# set ge-0/0/0 unit 1 vlan-id 1
user@PE1# ge-0/0/0 unit 1 family inet address 10.1.1.2/30
user@PE1# set ge-0/0/0 unit 1 family mpls
```

3. Configure the LSP VLAN on the ge-0/0/0 interface.

```
[edit interfaces]
user@PE1# set ge-0/0/0 unit 10 encapsulation vlan-ccc
user@PE1# set ge-0/0/0 unit 10 vlan-id 10
```

4. Configure the interface connecting Router PE1 to the core routers (Router P1 and Router P2).

```
[edit interfaces]
user@PE1# set ge-0/0/1 unit 0 family inet address 10.70.70.1/30
user@PE1# set ge-0/0/1 unit 0 family mpls
user@PE1# set ge-0/0/2 unit 0 family inet address 10.20.20.1/30
user@PE1# set ge-0/0/2 unit 0 family mpls
```

5. Configure the GRE tunnel as the controlling interface for Router PE1.

```
[edit interfaces]
user@PE1# set gre unit 0 tunnel source 10.1.1.2
```

```
user@PE1# set gre unit 0 tunnel destination 10.1.1.1
user@PE1# set gre unit 0 family inet address 10.35.100.26/30
```

6. Configure the loopback interface of Router PE1.

```
[edit interfaces]
user@PE1# set lo0 unit 0 family inet address 10.255.10.2/32
```

7. Configure the loopback address of Router PE1 as its router ID.

```
[edit routing-options]
user@PE1# set router-id 10.255.10.2
```

8. Configure an associated bidirectional LSP, and enable unidirectional reverse LSP setup for single-sided provisioned forward LSP.

```
[edit protocols]
user@PE1# set rsvp associated-bidirectional-lsp single-sided-provisioning
```

9. Enable RSVP on all the interfaces of Router PE1, excluding the management interface.

```
[edit protocols]
user@PE1# set rsvp interface all
user@PE1# set rsvp interface fxp0.0 disable
```

10. Configure the RSVP peer interface for Router PE1, and enable dynamic setup of bidirectional packet LSP for transporting nonpacket GMPLS LSP.

```
[edit protocols]
user@PE1# set rsvp peer-interface CE1 dynamic-bidirectional-transport
```

11. Enable MPLS on all the interfaces of Router PE1, excluding the management interface.

```
[edit protocols]
user@PE1# set mpls interface all
user@PE1# set mpls interface fxp0.0 disable
```

12. Configure OSPF with traffic engineering capabilities.

```
[edit protocols]
user@PE1# set ospf traffic-engineering
```

13. Enable OSPF area 0 on all the interfaces of Router PE1, excluding the management interface.

```
[edit protocols]
user@PE1# set ospf area 0.0.0.0 interface all
user@PE1# set ospf area 0.0.0.0 interface fxp0.0 disable
```

14. Configure a traffic engineering link, and assign addresses for the local and remote end of the link.

```
[edit protocols]
user@PE1# set link-management te-link link1 local-address 10.35.1.2
user@PE1# set link-management te-link link1 remote-address 10.35.1.1
```

15. Enable setting up of a Layer 2 VLAN LSP for a specific range of VLANs on the link1 traffic engineering link.

```
[edit protocols]
user@PE1# set link-management te-link link1 ethernet-vlan vlan-id-range 1-1000
```

16. Configure the Router PE1 interface as the member interface of the link1 traffic engineering link.

```
[edit protocols]
user@CE1# set link-management te-link link1 interface ge-0/0/0
```

17. Configure Router CE1 as the LMP peer for Router PE1, and configure the peer attributes.

```
[edit protocols]
user@CE1# set link-management peer CE1 address 10.255.10.1
user@CE1# set link-management peer CE1 control-channel gre.0
user@CE1# set link-management peer CE1 te-link link1
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@PE1# show interfaces
ge-0/0/0 {
    vlan-tagging;
    encapsulation flexible-ethernet-services;
    unit 1 {
        vlan-id 1;
        family inet {
            address 10.1.1.2/30;
        }
        family mpls;
    }
    unit 10 {
        encapsulation vlan-ccc;
        vlan-id 10;
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.70.70.1/30;
        }
        family mpls;
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 10.20.20.1/30;
        }
        family mpls;
    }
}
gre {
    unit 0 {
        tunnel {
            source 10.1.1.2;
```

```
            destination 10.1.1.1;
        }
        family inet {
            address 10.35.100.26/30;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.255.10.2/32;
        }
    }
}
```

```
user@PE1# show routing-options
router-id 10.255.10.2;
```

```
user@PE1# show protocols
rsvp {
    associated-bidirectional-lsp single-sided-provisioning;
    interface all;
    interface fxp0.0 {
        disable;
    }
    peer-interface CE1 {
        dynamic-bidirectional-transport;
    }
}
mpls {
    interface all;
    interface fxp0.0 {
        disable;
    }
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface all;
        interface fxp0.0 {
```

```
            disable;
        }
    }
}
link-management {
    te-link link1 {
        local-address 10.35.1.2;
        remote-address 10.35.1.1;
        ethernet-vlan {
            vlan-id-range 1-1000;
        }
        interface ge-0/0/0;
    }
    peer CE1 {
        address 10.255.10.1;
        control-channel gre.0;
        te-link link1;
    }
}
```

Verification

IN THIS SECTION

- Verifying the Traffic Engineering Link Status on the Client Routers | **2034**
- Verifying the RSVP Session Status on the Client Routers | **2036**
- Verifying the LSP Status on the Server Router | **2037**
- Verifying the CCC Entries in the MPLS Routing Table of the Server Routers | **2038**
- Verifying End-to-End Connectivity | **2039**

Confirm that the configuration is working properly.

Verifying the Traffic Engineering Link Status on the Client Routers

Purpose

Verify the status of the traffic engineering link configured between Router CE1 and Router CE2.

Action

From operational mode, run the **show link-management** and the **show link-management te-link detail** commands.

```
user@CE1> show link-management
Peer name: PE1, System identifier: 50740
 State: Up, Control address: 10.255.10.2
 Hello interval: 150, Hello dead interval: 500
   Control-channel                  State
   gre.0                            Active
  TE links:
   link10

 TE link name: link10, State: Up
  Local identifier: 65075, Remote identifier: 0, Local address: 10.35.1.1, Remote address:
10.35.1.2, Encoding: Ethernet, Switching: EVPL, Minimum bandwidth: 0bps,
  Maximum bandwidth: 1000Mbps, Total bandwidth: 1000Mbps, Available bandwidth: 900Mbps
    Name                             State Local ID  Remote ID      Bandwidth Used  LSP-name
    ge-0/0/0                         Up       54183          0       1000Mbps  Yes  CE1-to-CE2
```

```
user@CE1> show link-management te-link detail
TE link name: link10, State: Up
  Local identifier: 65075, Remote identifier: 0, Local address: 10.35.1.1, Remote address:
10.35.1.2, Encoding: Ethernet, Switching: EVPL, Minimum bandwidth: 0bps,
  Maximum bandwidth: 1000Mbps, Total bandwidth: 1000Mbps, Available bandwidth: 900Mbps
    Resource: ge-0/0/0, Type: IFD, System identifier: 137, State: Up, Local identifier: 54183,
Remote identifier: 0
    Total bandwidth: 1000Mbps, Unallocated bandwidth: 900Mbps
    Traffic parameters: Encoding: Ethernet, Switching: EVPL, Granularity: Unknown
    Maximum allocations: 4094, Number of allocations: 1, Unique allocations: 1, In use: Yes
      LSP name: CE1-to-CE2, Local label: 10, Remote label: 10, Allocated bandwidth: 100Mbps
```

```
user@CE2> show link-management
Peer name: PE2, System identifier: 50743
 State: Up, Control address: 10.255.10.5
 Hello interval: 150, Hello dead interval: 500
   Control-channel                  State
   gre.0                            Active
```

```
  TE links:
   link10

 TE link name: link10, State: Up
  Local identifier: 65075, Remote identifier: 0, Local address: 10.36.1.1, Remote address:
10.36.1.2, Encoding: Ethernet, Switching: EVPL, Minimum bandwidth: 0bps,
  Maximum bandwidth: 1000Mbps, Total bandwidth: 1000Mbps, Available bandwidth: 900Mbps
    Name                            State Local ID  Remote ID      Bandwidth Used  LSP-name
    ge-0/0/0                        Up       54183          0       1000Mbps  Yes  CE1-to-CE2
```

Meaning

The Link Management Protocol (LMP) peering has been established between the client routers, and the traffic engineering link is up on both Routers CE1 and CE2.

Verifying the RSVP Session Status on the Client Routers

Purpose

Verify the status of the RSVP sessions between Router CE1 and Router CE2.

Action

From operational mode, run the **show rsvp session** command.

```
user@CE1> show rsvp session
Ingress RSVP: 1 sessions
To              From            State   Rt Style Labelin Labelout LSPname
10.255.10.6     10.255.10.1     Up       0  1 FF       -       10 CE1-to-CE2 Bidir
Total 1 displayed, Up 1, Down 0

Egress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

```
user@CE2> show rsvp session
Ingress RSVP: 0 sessions
```

```
Total 0 displayed, Up 0, Down 0

Egress RSVP: 1 sessions
To              From            State   Rt Style Labelin Labelout LSPname
10.255.10.6     10.255.10.1     Up       0  1 FF      10        - CE1-to-CE2 Bidir
Total 1 displayed, Up 1, Down 0

Transit RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Meaning

The RSVP sessions are established between the ingress router, Router CE1, and the egress router, Router CE2.

Verifying the LSP Status on the Server Router

Purpose

Verify the status of the MPLS LSP on Router PE1.

Action

From operational mode, run the **show mpls lsp** command.

```
user@PE1> show mpls lsp
Ingress LSP: 1 sessions
To              From            State Rt P     ActivePath       LSPname
10.255.10.5     10.255.10.2     Up     0 *                      vlan:0:10:8176:10.255.10.2-
>10.255.10.5 Assoc-Bidir
Total 1 displayed, Up 1, Down 0

Egress LSP: 1 sessions
To              From            State   Rt Style Labelin Labelout LSPname
10.255.10.2     10.255.10.5     Up       0  1 FF       3        - vlan:0:10:8176:10.255.10.2-
>10.255.10.5:rev
Total 1 displayed, Up 1, Down 0

Transit LSP: 1 sessions
To              From            State   Rt Style Labelin Labelout LSPname
```

```
10.255.10.6     10.255.10.1     Up       0  1 FF      10   299808 CE1-to-CE2 Bidir
Total 1 displayed, Up 1, Down 0
```

Meaning

The CE1-to-CE2 LSP is established, and the output displays the LSP attributes.

Verifying the CCC Entries in the MPLS Routing Table of the Server Routers

Purpose

Verify the circuit cross-connect (CCC) interface entries in the MPLS routing table.

Action

From operational mode, run the **show route table mpls.0** and the **show route forwarding-table ccc *ccc-interface*** commands.

```
user@PE1> show route table mpls.0
mpls.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 1d 22:14:51, metric 1
                      Receive
1                  *[MPLS/0] 1d 22:14:51, metric 1
                      Receive
2                  *[MPLS/0] 1d 22:14:51, metric 1
                      Receive
13                 *[MPLS/0] 1d 22:14:51, metric 1
                      Receive
299824             *[RSVP/7/1] 17:32:07, metric 1
                    > via ge-0/0/0.10, Pop
ge-0/0/0.10        *[RSVP/7/1] 17:32:07, metric 1
                    > to 10.20.20.2 via ge-0/0/2.0, label-switched-path CE1-to-CE2
```

```
user@PE1> show route forwarding-table ccc ge-0/0/0.10
Routing table: default.mpls
MPLS:
Destination        Type RtRef Next hop           Type Index    NhRef Netif
```

```
ge-0/0/0.10  (CCC) user     0 10.20.20.2       Push 299808, Push 299872(top)      581     2
ge-0/0/2.0

Routing table: __mpls-oam__.mpls
MPLS:
Destination        Type RtRef Next hop           Type Index    NhRef Netif
default            perm     0                    dscd      534     1
```

Meaning

The output displays the CCC interface that is the client-router-facing interface and the next-hop details for that interface.

Verifying End-to-End Connectivity

Purpose

Verify the connectivity between Router CE1 and the remote client router, Router CE2.

Action

From operational mode, run the **ping** command.

```
user@CE1> ping 10.10.10.2
PING 10.10.10.2 (10.10.10.2): 56 data bytes
64 bytes from 10.10.10.2: icmp_seq=0 ttl=64 time=15.113 ms
64 bytes from 10.10.10.2: icmp_seq=1 ttl=64 time=13.353 ms
64 bytes from 10.10.10.2: icmp_seq=2 ttl=64 time=13.769 ms
64 bytes from 10.10.10.2: icmp_seq=3 ttl=64 time=10.341 ms
64 bytes from 10.10.10.2: icmp_seq=4 ttl=64 time=12.597 ms
^C
--- 10.10.10.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 10.341/13.035/15.113/1.575 ms
```

Meaning

The ping from Router CE1 to Router CE2 is successful.

RELATED DOCUMENTATION

Basic MPLS Configuration | 46

PART 8

MPLS VPNs and Circuits

- Ethernet over MPLS (L2 Circuit) | **2042**
- CCC, TCC, and Layer 2.5 Switching | **2050**

CHAPTER 17

Ethernet over MPLS (L2 Circuit)

IN THIS CHAPTER

- Understanding Ethernet-over-MPLS (L2 Circuit) | 2042
- Configuring Ethernet over MPLS (Layer 2 Circuit) | 2043

Understanding Ethernet-over-MPLS (L2 Circuit)

IN THIS SECTION

- Ethernet-over-MPLS in Data Centers | 2042

Ethernet-over-MPLS allows sending Layer 2 (L2) Ethernet frames transparently over MPLS. Ethernet-over-MPLS uses a tunneling mechanism for Ethernet traffic through an MPLS-enabled Layer 3 core. It encapsulates Ethernet protocol data units (PDUs) inside MPLS packets and forwards the packets, using label stacking, across the MPLS network This technology has applications in service provider, enterprise and data center environments. For disaster recovery purposes, data centers are hosted in multiple sites that are geographically distant and interconnected using a WAN network.

NOTE: A Layer 2 circuit is similar to a circuit cross-connect (CCC), except that multiple Layer 2 circuits can be transported over a single label-switched path (LSP) tunnel between two provider edge (PE) routers. In contrast, each CCC requires a dedicated LSP.

Ethernet-over-MPLS in Data Centers

For disaster recovery purposes, data centers are hosted in multiple sites that are geographically distant and interconnected using a WAN network. These data centers require L2 connectivity between them for the following reasons:

- To replicate the storage over Fiber Channel IP (FCIP). FCIP works only on the same broadcast domain.
- To run a dynamic routing protocol between the sites.
- To support High Availability clusters that interconnect the nodes hosted in the various data centers.

RELATED DOCUMENTATION

Configuring Ethernet over MPLS (Layer 2 Circuit) | **2043**

Configuring Ethernet over MPLS (Layer 2 Circuit)

IN THIS SECTION

- Configuring the Local PE Switch for Port-Based Layer 2 Circuit (Pseudo-wire) | **2044**
- Configuring the Remote PE Switch for Port-Based Layer 2 Circuit (Pseudo-wire) | **2046**
- Configuring the Local PE Switch for VLAN-Based Layer 2 Circuit | **2046**
- Configuring the Remote PE Switch for VLAN-Based Layer 2 Circuit | **2048**

To implement Ethernet over MPLS, you must configure a Layer 2 circuit on the provider edge (PE) switches. No special configuration is required on the customer edge (CE) switches. The provider switches require MPLS and LDP to be configured on the interfaces that will be receiving and transmitting MPLS packets.

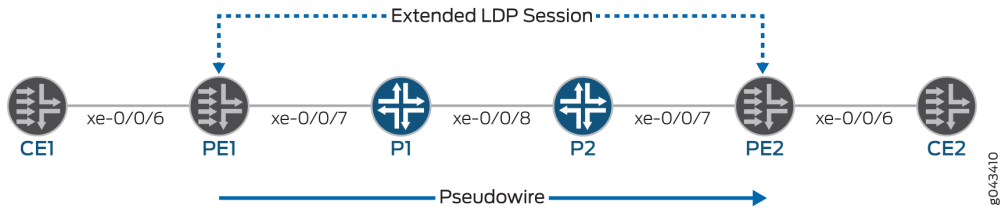
NOTE: A Layer 2 circuit is similar to a circuit cross-connect (CCC), except that multiple Layer 2 circuits can be transported over a single label-switched path (LSP) tunnel between two PE switches. In contrast, each CCC requires a dedicated LSP.

This topic describes how to configure the PE switches to support Ethernet over MPLS. You must configure interfaces and protocols on both the local PE (PE1) and the remote PE (PE2) switches. The interface configuration varies depending upon whether the Layer 2 circuit is port-based or VLAN-based.

Starting in Junos OS Release 20.3R1, support for Layer 2 circuit to provide Layer 2 VPN and VPWS with LDP signaling.

Figure 134 on page 2044 shows an example of a Layer 2 circuit configuration.

Figure 134: Ethernet over MPLS Layer 2 Circuit



NOTE: This topic refers to the local PE switch as PE1 and the remote PE switch as PE2. It also uses interface names rather than variables to help clarify the connections between the switches. The loopback addresses of the switches are configured as follows:

- PE1: 10.127.1.1
- PE2: 10.127.1.2

NOTE: On QFX Series and EX4600 switches, the Layer 2 circuit CE facing interface does not support AE interfaces.

Configuring the Local PE Switch for Port-Based Layer 2 Circuit (Pseudo-wire)

CAUTION: Configure MPLS networks with an MTU (maximum transmission unit) that is at least 12 bytes larger than the largest frame size that will be transported by the LSPs. If the size of a encapsulated packet on the ingress LSR exceeds the LSP MTU, that packet is dropped. If an egress LSR receives a packet on a VC LSP with a length (after the label stack and sequencing control word have been popped) that exceeds the MTU of the destination layer 2 interface, that packet is also dropped.

To configure the local PE switch (PE1) for a port-based layer 2 circuit (pseudo-wire):

1. Configure an access CE-facing interface for Ethernet encapsulation:

```
[edit interfaces]
user@switch# set xe-0/0/6 encapsulation ethernet-ccc
user@switch# set xe-0/0/6 unit 0
```

NOTE: Note that only unit number 0 is supported for Ethernet CCC.

2. Configure the Layer 2 circuit from PE1 to PE2:

```
[edit protocols]
user@switch#set l2circuit neighbor 10.127.1.1 interface xe-0/0/6 virtual-circuit-id 1
```

3. Configure the label switched path from PE1 to PE2:

```
[edit protocols]
user@switch#set mpls label-switched-path PE1-to-PE2 to 10.127.1.1
```

4. Configure the protocols on the core and loopback interfaces:

```
[edit protocols]
user@switch#set mpls interface xe-0/0/7
user@switch#set ldp interface xe-0/0/7
user@switch#set ldp interface lo0.0
```

The changes are shown for the local PE:

```
[edit ]
user@device# show interfaces
xe-0/0/6 {
    encapsulation ethernet-ccc;
    unit 0;
}

[edit]
user@device# show protocols
l2circuit {
    neighbor 10.127.1.1 {
        interface xe-0/0/6.0 {
            virtual-circuit-id 1;
        }
    }
}
ldp {
    interface xe-0/0/7.0;
```

```
    interface lo0.0;
}
mpls {
    label-switched-path PE1-to-PE2 {
        to 10.127.1.1;
    }
    interface xe-0/0/7.0;
}
```

Configuring the Remote PE Switch for Port-Based Layer 2 Circuit (Pseudo-wire)

To configure the remote PE switch (PE2) for a port-based layer 2 circuit:

1. Configure an access CE-facing interface for Ethernet encapsulation:

```
[edit interfaces]
user@switch# set xe-0/0/6 encapsulation ethernet-ccc
user@switch# set xe-0/0/6 unit 0
```

2. Configure the Layer 2 circuit from PE2 to PE1:

```
[edit protocols]
user@switch#set l2circuit neighbor 10.127.1.2 interface xe-0/0/6 virtual-circuit-id 1
```

3. Configure the label switched path from PE2 to PE1:

```
[edit protocols]
user@switch#set mpls label-switched-path PE2-to-PE1 to 10.127.1.2
```

4. Configure the protocols on the core and loopback interfaces:

```
[edit protocols]
user@switch#set mpls interface xe-0/0/7
user@switch#set ldp interface xe-0/0/7
user@switch#set ldp interface lo0.0
```

Configuring the Local PE Switch for VLAN-Based Layer 2 Circuit

To configure the local PE switch (PE1) for a VLAN-based layer 2 circuit:

1. Configure an access CE-facing interface for VLAN encapsulation:

```
[edit interfaces]
user@switch# set xe-0/0/6 encapsulation vlan-ccc
```

2. Configure the logical unit of the CE-facing interface for VLAN encapsulation:

```
[edit interfaces]
user@switch# set xe-0/0/6 unit 0 encapsulation vlan-ccc
```

3. Configure the logical unit of the CE-facing interface to belong to family ccc:

```
[edit interfaces]
user@switch# set xe-0/0/6 unit 0 family ccc
```

4. Configure the same interface for VLAN tagging:

```
[edit interfaces]
user@switch# set xe-0/0/6 vlan-tagging
```

5. Configure the VLAN ID of the interface:

```
[edit interfaces]
user@switch# set xe-0/0/6 unit 0 vlan-id 600
```

6. Configure the Layer 2 circuit from PE1 to PE2:

```
[edit protocols]
user@switch#set l2circuit neighbor 10.127.1.1 interface xe-0/0/6 virtual-circuit-id 1
```

7. Configure the label switched path from PE1 to PE2:

```
[edit protocols]
user@switch#set mpls label-switched-path PE1-to-PE2 to 10.127.1.1
```

8. Configure the protocols on the core and loopback interfaces:

```
[edit protocols]
user@switch#set mpls interface xe-0/0/7
user@switch#set ldp interface xe-0/0/7
user@switch#set ldp interface lo0.0
```

Configuring the Remote PE Switch for VLAN-Based Layer 2 Circuit

To configure the remote PE switch (PE2) for a VLAN-based layer 2 circuit:

1. Configure an access CE-facing interface for VLAN encapsulation:

```
[edit interfaces]
user@switch# set xe-0/0/6 encapsulation vlan-ccc
```

2. Configure the logical unit of the CE-facing interface for VLAN encapsulation:

```
[edit interfaces]
user@switch# set xe-0/0/6 unit 0 encapsulation vlan-ccc
```

3. Configure the logical unit of the CE-facing interface to belong to family ccc:

```
[edit interfaces]
user@switch# set xe-0/0/6 unit 0 family ccc
```

4. Configure the same interface for VLAN tagging:

```
[edit interfaces]
user@switch# set xe-0/0/6 vlan-tagging
```

5. Configure the VLAN ID of the interface:

```
[edit interfaces]
user@switch# set xe-0/0/6 unit 0 vlan-id 600
```

6. Configure the Layer 2 circuit from PE2 to PE1:

```
[edit protocols]
user@switch#set l2circuit neighbor 10.127.1.2 interface xe-0/0/6 virtual-circuit-id 1
```

7. Configure the label switched path from PE2 to PE1:

```
[edit protocols]
user@switch#set mpls label-switched-path PE2-to-PE1 to 10.127.1.2
```

8. Configure the protocols on the core and loopback interfaces:

```
[edit protocols]
user@switch#set mpls interface xe-0/0/7
user@switch#set ldp interface xe-0/0/7
user@switch#set ldp interface lo0.0
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
20.3R1	Starting in Junos OS Release 20.3R1, support for Layer 2 circuit to provide Layer 2 VPN and VPWS with LDP signaling.

CHAPTER 18

CCC, TCC, and Layer 2.5 Switching

IN THIS CHAPTER

- CCC, TCC, and Ethernet Over MPLS Configuration | **2050**

CCC, TCC, and Ethernet Over MPLS Configuration

IN THIS SECTION

- TCC and Layer 2.5 Switching Overview | **2051**
- Configuring VLAN TCC Encapsulation | **2051**
- Configuring TCC Interface Switching | **2053**
- CCC Overview | **2055**
- Understanding Carrier-of-Carriers VPNs | **2056**
- Understanding Interprovider and Carrier-of-Carriers VPNs | **2058**
- Configuring BGP to Gather Interprovider and Carrier-of-Carriers VPNs Statistics | **2059**
- Configuring an MPLS-Based VLAN CCC Using a Layer 2 Circuit | **2060**
- VLAN CCC Encapsulation on Transport Side of Pseudowire Client Logical Interfaces Overview | **2064**
- Transmitting Nonstandard BPDUs | **2067**
- TCC Overview | **2067**
- Configuring Layer 2 Switching Cross-Connects Using CCC | **2068**
- Configuring MPLS LSP Tunnel Cross-Connects Using CCC | **2079**
- Configuring TCC | **2084**
- CCC and TCC Graceful Restart | **2090**
- Configuring CCC and TCC Graceful Restart | **2091**
- Configuring an MPLS-Based VLAN CCC Using the Connection Method (CLI Procedure) | **2092**
- Configuring CCC Switching for Point-to-Multipoint LSPs | **2094**

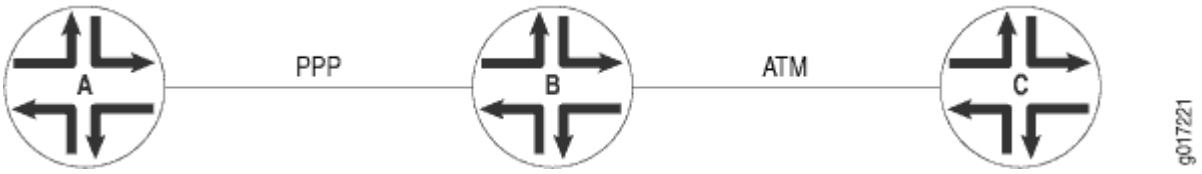
- Configuring an MPLS-Based VLAN CCC Using a Layer 2 VPN (CLI Procedure) | **2096**
- Understanding Ethernet-over-MPLS (L2 Circuit) | **2101**
- Configuring Ethernet over MPLS (Layer 2 Circuit) | **2102**

TCC and Layer 2.5 Switching Overview

Translational cross-connect (TCC) allows you to forward traffic between a variety of Layer 2 protocols or circuits. It is similar to its predecessor, CCC. However, while CCC requires the same Layer 2 encapsulations on both sides of a router (such as Point-to-Point Protocol [PPP] or Frame Relay-to-Frame Relay), TCC lets you connect different types of Layer 2 protocols interchangeably. With TCC, combinations such as PPP-to-ATM and Ethernet-to-Frame Relay cross-connections are possible. Also, TCC can be used to create Layer 2.5 VPNs and Layer 2.5 circuits.

Consider a sample topology (Figure 135 on page 2051) in which you can configure a full-duplex Layer 2.5 translational cross-connect between Router A and Router C, using a Juniper Networks router, Router B, as the TCC interface. In this topology, Router B strips all PPP encapsulation data from frames arriving from Router A and adds ATM encapsulation data before the frames are sent to Router C. All Layer 2 negotiations are terminated at the interconnecting router (Router B).

Figure 135: Sample Translation Cross-Connect Topology



TCC functionality is different from standard Layer 2 switching. TCC only swaps Layer 2 headers. No other processing, such as header checksums, time-to-live (TTL) decrementing, or protocol handling, is performed. Currently, TCC is supported in IPv4, ISO, and MPLS.

Ethernet TCC is supported on interfaces that carry IPv4 traffic only. For 8-port, 12-port, and 48-port Fast Ethernet PICs, TCC and extended VLAN CCC are not supported. For 4-port Gigabit Ethernet PICs, extended VLAN CCC and extended VLAN TCC are not supported.

Configuring VLAN TCC Encapsulation

VLAN TCC encapsulation allows circuits to have different media on either side of the forwarding path. VLAN TCC encapsulation supports TPID 0x8100 only. You must include configuration statements at the logical and physical interface hierarchy levels.

Starting in Junos OS Release 20.1R1, aggregated Ethernet interfaces support VLAN translational cross-connect (TCC) encapsulation. For configuring VLAN TCC encapsulation, you must have the member links of aggregated Ethernet with VLAN TCC encapsulation supported hardware.

NOTE: MX series routers does not perform any external commit check for member links of aggregated interfaces for the VLAN TCC encapsulation supported hardware.

To configure VLAN TCC encapsulation, include the `encapsulation` statement and specify the `vlan-tcc` option:

```
[edit interfaces interface-name unit logical-unit-number]
encapsulation vlan-tcc;
```

You can include this statement at the following hierarchy levels:

- `[edit interfaces interface-name unit logical-unit-number ]`
- `[edit logical-systems logical-system-name interfaces interface-name unit logical-unit-number]`

Additionally, configure the logical interface by including the `proxy` and `remote` statements:

```
proxy {
    inet-address;
}
remote {
    (inet-address | mac-address);
}
```

You can include these statements at the following hierarchy levels:

- `[edit interfaces interface-name unit logical-unit-number family tcc]`
- `[edit logical-systems logical-system-name interfaces interface-name unit logical-unit-number family tcc]`

The proxy address is the IP address of the non-Ethernet TCC neighbor for which the TCC router is acting as a proxy.

The remote address is the IP or MAC address of the remote router. The `remote` statement provides ARP capability from the TCC switching router to the Ethernet neighbor. The MAC address is the physical Layer 2 address of the Ethernet neighbor.

When VLAN TCC encapsulation is configured on the logical interface, you also must specify flexible Ethernet services on the physical interface. To specify flexible Ethernet services, include the `encapsulation`

statement at the [edit interfaces *interface-name*] hierarchy level and specify the flexible-ethernet-services option:

```
[edit interfaces interface-name]
encapsulation flexible-ethernet-services;
```

Extended VLAN TCC encapsulation supports TPIDs 0x8100 and 0x9901. Extended VLAN TCC is specified at the physical interface level. When configured, all units on that interface must use VLAN TCC encapsulation, and no explicit configuration is needed on logical interfaces.

One-port Gigabit Ethernet, 2-port Gigabit Ethernet, and 4-port Fast Ethernet PICs with VLAN tagging enabled can use VLAN TCC encapsulation. To configure the encapsulation on a physical interface, include the encapsulation statement at the [edit interfaces *interface-name*] hierarchy level and specify the extended-vlan-tcc option:

```
[edit interfaces interface-name]
encapsulation extended-vlan-tcc;
```

For VLAN TCC encapsulation, all VLAN IDs from 1 through 1024 are valid. VLAN ID 0 is reserved for tagging the priority of frames.

Extended VLAN TCC is not supported on 4-port Gigabit Ethernet PICs.

Configuring TCC Interface Switching

To configure a full-duplex Layer 2.5 translation cross-connect between two routers (A and C), you can configure a Juniper Networks router (Router B) as the TCC interface. Ethernet TCC encapsulation provides an Ethernet wide area circuit for interconnecting IP traffic. Consider the topology in Figure 136 on page 2053 where the Router A-to-Router B circuit is PPP, and the Router B-to-Router C circuit accepts packets carrying standard TPID values.

Figure 136: Sample Topology of Layer 2.5 Translational Cross-Connect

If traffic flows from Router A to Router C, the Junos OS strips all PPP encapsulation data from incoming packets and adds Ethernet encapsulation data before forwarding the packets. If traffic flows from Router C to Router A, the Junos OS strips all Ethernet encapsulation data from incoming packets and adds PPP encapsulation data before forwarding the packets.

To configure the router as the translational cross-connect interface:

1. In the configuration mode, at the [edit] hierarchy level, first configure the interface that is connected to Router A.

```
[edit]
user@host# edit interfaces interface-name
```

2. (Optional) Specify the description of the interface. For example, you could specify the interface name on Router A that is connected to this interface.

```
[edit interfaces interface-name]
user@host# set description description
```

3. Specify the encapsulation. If the Router A to Router B circuit is PPP, then specify ppp-tcc as the encapsulation. If the Router A to Router B circuit is frame relay, specify frame-relay-tcc.

```
[edit interfaces interface-name]
user@host# set encapsulation encapsulation-type
```

4. In the configuration mode, at the [edit] hierarchy level, first configure the interface that is connected to Router C.

```
[edit]
user@host# edit interfaces interface-name
```

5. (Optional) Specify the description of this interface. For example, you could specify the interface name on Router C that is connected to this interface.

```
[edit interfaces interface-name]
user@host# set description description
```

6. Specify the encapsulation. If the Router B to Router C circuit is Ethernet, then specify ethernet-tcc as the encapsulation. If the Router B to Router C circuit is ATM, specify atm-tcc-vc-mux.

```
[edit interfaces interface-name]
user@host# set encapsulation encapsulation-type
```

7. Specify the IP address or MAC address of the remote router to provide address resolution protocol (ARP) for the TCC router's Ethernet-based neighbor using the remote statement. You must specify the

statement at the [edit interfaces interface-name unit unit-number family tcc] hierarchy level. You can a specify the MAC address of the remote router instead of the IP address. The MAC address is the physical Layer 2 address of the Ethernet neighbor.

```
[edit interfaces interface-name]
user@host# set unit 0 family family remote inet-address ip-address
```

8. Specify the IP address of the non-Ethernet TCC neighbor for which the TCC router is acting as a proxy using the proxy statement. You must specify the statement at the [edit interfaces interface-name unit unit-number family tcc] hierarchy level.

```
[edit interfaces interface-name]
user@host# set unit 0 family family proxy inet-address ip-address
```

To verify the TCC connection, use the show connections command on TCC router.

CCC Overview

Circuit cross-connect (CCC) allows you to configure transparent connections between two circuits, where a circuit can be a Frame Relay data-link connection identifier (DLCI), an Asynchronous Transfer Mode (ATM) virtual circuit (VC), a Point-to-Point Protocol (PPP) interface, a Cisco High-Level Data Link Control (HDLC) interface, or an MPLS label-switched path (LSP). Using CCC, packets from the source circuit are delivered to the destination circuit with, at most, the Layer 2 address being changed. No other processing—such as header checksums, time-to-live (TTL) decrementing, or protocol processing—is done.

NOTE: The QFX10000 Series switches do not support ATM virtual circuits.

CCC circuits fall into two categories: logical interfaces, which include DLCIs, VCs, virtual local area network (VLAN) IDs, PPP and Cisco HDLC interfaces, and LSPs. The two circuit categories provide three types of cross-connect:

- Layer 2 switching—Cross-connects between logical interfaces provide what is essentially Layer 2 switching. The interfaces that you connect must be of the same type.
- MPLS tunneling—Cross-connects between interfaces and LSPs allow you to connect two distant interface circuits of the same type by creating MPLS tunnels that use LSPs as the conduit.
- LSP stitching—Cross-connects between LSPs provide a way to “stitch” together two label-switched paths, including paths that fall in two different traffic engineering database areas.

For Layer 2 switching and MPLS tunneling, the cross-connect is bidirectional, so packets received on the first interface are transmitted out the second interface, and those received on the second interface are transmitted out the first. For LSP stitching, the cross-connect is unidirectional.

Understanding Carrier-of-Carriers VPNs

IN THIS SECTION

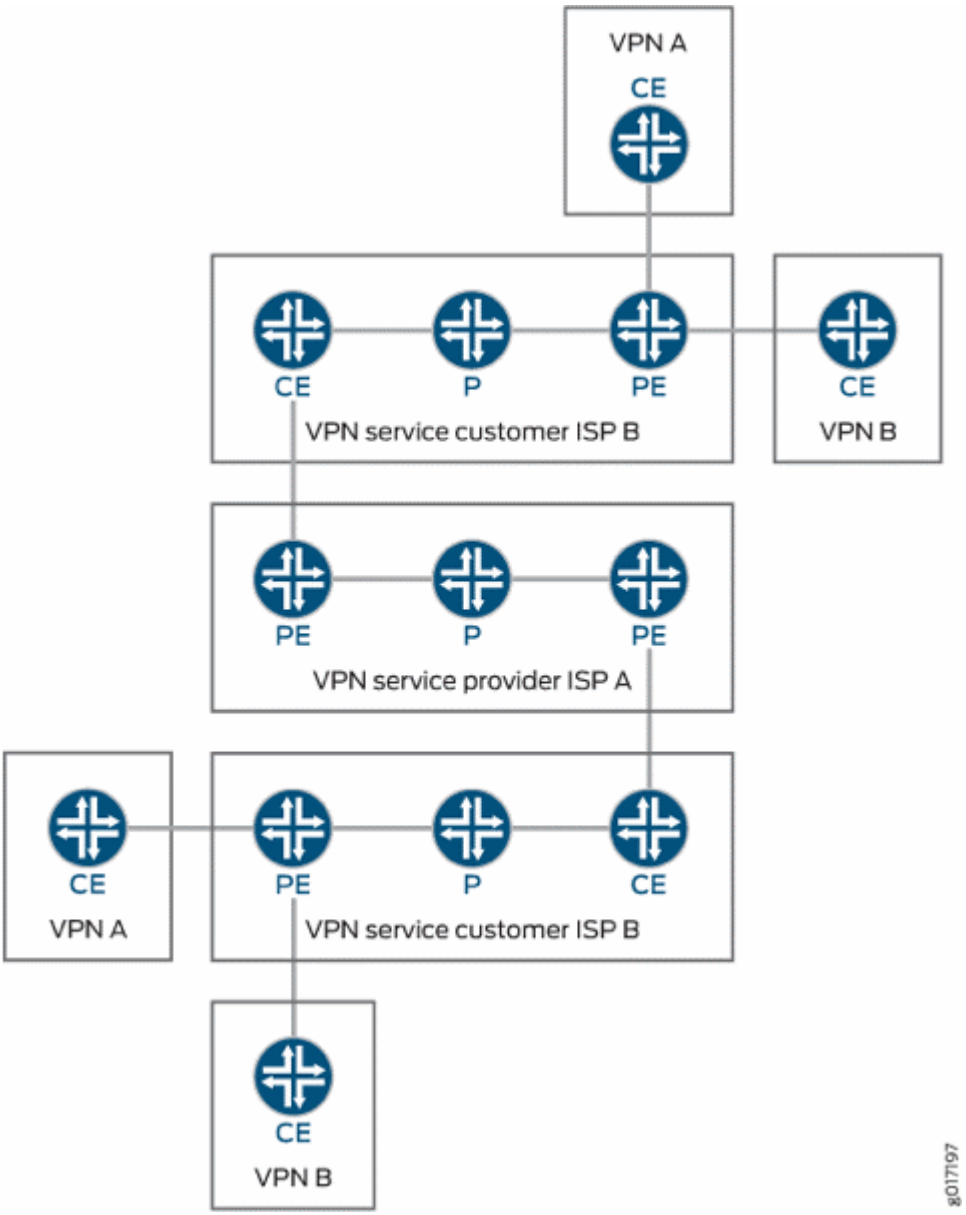
- Internet Service Provider as the Customer | **2057**
- VPN Service Provider as the Customer | **2058**

The customer of a VPN service provider might be a service provider for the end customer. The following are the two main types of carrier-of-carriers VPNs (as described in RFC 4364:

- "Internet Service Provider as the Customer" on page 2057—The VPN customer is an ISP that uses the VPN service provider's network to connect its geographically disparate regional networks. The customer does not have to configure MPLS within its regional networks.
- "VPN Service Provider as the Customer" on page 2058—The VPN customer is itself a VPN service provider offering VPN service to its customers. The carrier-of-carriers VPN service customer relies on the backbone VPN service provider for inter-site connectivity. The customer VPN service provider is required to run MPLS within its regional networks.

Figure 137 on page 2057 illustrates the network architecture used for a carrier-of-carriers VPN service.

Figure 137: Carrier-of-Carriers VPN Architecture



This topic covers the following:

Internet Service Provider as the Customer

In this type of carrier-of-carriers VPN configuration, ISP A configures its network to provide Internet service to ISP B. ISP B provides the connection to the customer wanting Internet service, but the actual Internet service is provided by ISP A.

This type of carrier-of-carriers VPN configuration has the following characteristics:

- The carrier-of-carriers VPN service customer (ISP B) does not need to configure MPLS on its network.
- The carrier-of-carriers VPN service provider (ISP A) must configure MPLS on its network.
- MPLS must also be configured on the CE routers and PE routers connected together in the carrier-of-carriers VPN service customer's and carrier-of-carriers VPN service provider's networks.

VPN Service Provider as the Customer

A VPN service provider can have customers that are themselves VPN service providers. In this type of configuration, also called a hierarchical or recursive VPN, the customer VPN service provider's VPN-IPv4 routes are considered external routes, and the backbone VPN service provider does not import them into its VRF table. The backbone VPN service provider imports only the customer VPN service provider's internal routes into its VRF table.

The similarities and differences between interprovider and carrier-of-carriers VPNs are shown in Table 45 on page 2058.

Table 45: Comparison of Interprovider and Carrier-of-Carriers VPNs

Feature	ISP Customer	VPN Service Provider Customer
Customer edge device	AS border router	PE router
IBGP sessions	Carry IPv4 routes	Carry external VPN-IPv4 routes with associated labels
Forwarding within the customer network	MPLS is optional	MPLS is required

Support for VPN service as the customer is supported on QFX10000 switches starting with Junos OS Release 17.1R1.

Understanding Interprovider and Carrier-of-Carriers VPNs

All interprovider and carrier-of-carriers VPNs share the following characteristics:

- Each interprovider or carrier-of-carriers VPN customer must distinguish between internal and external customer routes.
- Internal customer routes must be maintained by the VPN service provider in its PE routers.

- External customer routes are carried only by the customer's routing platforms, not by the VPN service provider's routing platforms.

The key difference between interprovider and carrier-of-carriers VPNs is whether the customer sites belong to the same AS or to separate ASs:

- *Interprovider VPNs*—The customer sites belong to different ASs. You need to configure EBGP to exchange the customer's external routes.
- *Understanding Carrier-of-Carriers VPNs*—The customer sites belong to the same AS. You need to configure IBGP to exchange the customer's external routes.

In general, each service provider in a VPN hierarchy is required to maintain its own internal routes in its P routers, and the internal routes of its customers in its PE routers. By recursively applying this rule, it is possible to create a hierarchy of VPNs.

The following are definitions of the types of PE routers specific to interprovider and carrier-of-carriers VPNs:

- The AS border router is located at the AS border and handles traffic leaving and entering the AS.
- The end PE router is the PE router in the customer VPN; it is connected to the CE router at the end customer's site.

Configuring BGP to Gather Interprovider and Carrier-of-Carriers VPNs Statistics

You can configure BGP to gather traffic statistics for interprovider and carrier-of-carriers VPNs.

To configure BGP to gather traffic statistics for interprovider and carrier-of-carriers VPNs, include the `traffic-statistics` statement:

```
traffic-statistics {
    file filename <world-readable | no-world-readable>;
    interval seconds;
}
```

For a list of the hierarchy levels at which you can include this statement, see the summary section for this statement.

NOTE: Traffic statistics for interprovider and carrier-of-carriers VPNs are available only for IPv4. IPv6 is not supported.

If you do not specify a filename, the statistics are not written to a file. However, if you have included the `traffic-statistics` statement in the BGP configuration, the statistics are still available and can be accessed by means of the `show bgp group traffic-statistics group-name` command.

To account for traffic from each customer separately, separate labels must be advertised for the same prefix to the peer routers in different groups. To enable separate traffic accounting, you need to include the `per-group-label` statement in the configuration for each BGP group. By including this statement, statistics are collected and displayed that account for traffic sent by the peers of the specified BGP group.

If you configure the statement at the `[edit protocols bgp family inet]` hierarchy level, rather than configuring it for a specific BGP group, then the traffic statistics are shared with all BGP groups configured with the `traffic-statistics` statement but not configured with the `per-group-label` statement.

To account for traffic from each customer separately, include the `per-group-label` statement in the configuration for each BGP group:

```
per-group-label;
```

For a list of the hierarchy levels at which you can include this statement, see the summary section for this statement.

The following shows a sample of the output to the traffic statistics file:

```
Dec 19 10:39:54 Statistics for BGP group ext2 (Index 1) NLRI inet-labeled-unicast
Dec 19 10:39:54  FEC               Packets         Bytes   EgressAS   FECLabel
Dec 19 10:39:54 10.255.245.55            0             0          I     100160
Dec 19 10:39:54 10.255.245.57            0             0          I     100112
Dec 19 10:39:54 192.0.2.1                0             0         25     100080
Dec 19 10:39:54 192.0.2.2                0             0         25     100080
Dec 19 10:39:54 192.0.2.3              109          9592         25     100048
Dec 19 10:39:54 192.0.2.4              109          9592         25     100048
Dec 19 10:39:54 192.168.25.0             0             0          I     100064
Dec 19 10:39:54  Dec 19 10:39:54, read statistics for 5 FECs in 00:00:00 seconds (10 queries)
for BGP group ext2 (Index 1) NLRI inet-labeled-unicast
```

Configuring an MPLS-Based VLAN CCC Using a Layer 2 Circuit

You can configure an 802.1Q VLAN as an MPLS-based Layer 2 circuit on the switch to interconnect multiple customer sites with Layer 2 technology.

This topic describes configuring provider edge (PE) switches in an MPLS network using a circuit cross-connect (CCC) on a tagged VLAN interface (802.1Q VLAN) rather than a simple interface.

NOTE: You do not need to make any changes to existing provider switches in your MPLS network to support this type of configuration. For information on configuring provider switches, see "Configuring MPLS on Provider Switches" on page 76.

NOTE: You can send any kind of traffic over a CCC, including nonstandard bridge protocol data units (BPDUs) generated by other vendors' equipment.

NOTE: If you configure a physical interface as VLAN-tagged and with the vlan-ccc encapsulation, you cannot configure the associated logical interfaces with the inet family. Doing so could cause the logical interfaces to drop packets.

To configure a PE switch with a VLAN CCC and an MPLS-based Layer 2 circuit:

1. Configure OSPF (or IS-IS) on the loopback (or switch address) and core interfaces:

```
[edit protocols]
user@switch# set ospf area 0.0.0.0 interface lo0.0
user@switch# set ospf area 0.0.0.0 interface interface-name
user@switch# set ospf area 0.0.0.0 interface interface-name
user@switch# set ospf area 0.0.0.0 interface interface-name
```

2. Enable traffic engineering for the routing protocol:

```
[edit protocols]
user@switch# set ospf traffic-engineering
```

3. Configure an IP address for the loopback interface and for the core interfaces:

```
[edit]
user@switch# set interfaces lo0 unit logical-unit-number family inet address address
user@switch# set interfaces interface-name unit logical-unit-number family inet address
address
user@switch# set interfaces interface-name unit logical-unit-number family inet address
address
user@switch# set interfaces interface-name unit logical-unit-number family inet address
address
```

4. Enable the MPLS protocol with CSPF disabled:

NOTE: CSPF is a shortest-path-first algorithm that has been modified to take into account specific restrictions when the shortest path across the network is calculated. You need to disable CSPF for link protection to function properly on interarea paths.

```
[edit protocols]
user@switch# set mpls no-cspf
```

5. Configure the customer edge interface as a Layer 2 circuit from the local PE switch to the other PE switch:

```
[edit protocols]
user@switch# set l2circuit neighbor address interface interface-name virtual-circuit-id
identifier
```

TIP: Use the switch address of the other switch as the neighbor address.

6. Configure MPLS on the core interfaces:

```
[edit protocols]
user@switch# set mpls interface interface-name
user@switch# set mpls interface interface-name
user@switch# set mpls interface interface-name
```

7. Configure LDP on the loopback interface and the core interfaces:

```
[edit protocols]
user@switch# set ldp interface lo0.0
user@switch# set ldp interface interface-name
user@switch# set ldp interface interface-name
user@switch# set ldp interface interface-name
```

8. Configure `family mpls` on the logical units of the core interfaces:

```
[edit]
user@switch# set interfaces interface-name unit logical-unit-number family mpls
```

```
user@switch# set interfaces interface-name unit logical-unit-number family mpls
user@switch# set interfaces interface-name unit logical-unit-number family mpls
```

NOTE: You can enable `family mpls` on either individual interfaces or aggregated Ethernet interfaces. You cannot enable it on tagged VLAN interfaces.

9. Enable VLAN tagging on the customer edge interface of the local PE switch:

```
[edit]
user@switch# set interfaces interface-name vlan-tagging
```

10. Configure the customer edge interface to use VLAN CCC encapsulation:

```
[edit]
user@switch# set interfaces interface-name encapsulation vlan-ccc
```

11. Configure the logical unit of the customer edge interface with a VLAN ID:

NOTE: The VLAN ID cannot be configured on logical interface unit `0`. The logical unit number must be `1` or higher.

The same VLAN ID must be used when configuring the customer edge interface on the other PE switch.

```
[edit ]
user@switch# set interfaces interface-name logical-unit-number vlan-idvlan-id
```

When you have completed configuring one PE switch, follow the same procedures to configure the other PE switch.

NOTE: For EX Series switches, you must use the same type of switch for the other PE switch.

VLAN CCC Encapsulation on Transport Side of Pseudowire Client Logical Interfaces Overview

IN THIS SECTION

- Pseudowire Configuration from Access Node | 2064
- Pseudowire Configuration from Aggregation Node | 2065

Currently, Junos OS does not allow the same VLAN ID to be configured on more than one logical interface under the same pseudowire client physical interface. To support `vlan-ccc` encapsulation on transport pseudowire service (PS) interface on the provider edge (PE) device, this restriction is removed and you can configure the same VLAN ID on more than one logical interface.

The primary reason to configure `vlan-ccc` on the transport PS interface is interoperability with the existing access and aggregate devices in the network. Currently, Junos OS supports `ethernet-ccc` encapsulation on the transport PS interface. Typically, while establishing a pseudowire connection, the access device initiates a VLAN-based pseudowire (also known as VLAN-tagged mode), and a PE router signals the Ethernet mode VLAN back to the access device. For this type of pseudowire connection to be established, you can use the `ignore-encapsulation-mismatch` statement. However, the Junos OS device (access device) might not support the `ignore-encapsulation-mismatch` statement and, as a result, the pseudowire connection is not formed. When the `ignore-encapsulation-mismatch` statement is not supported on the access device, you can configure `vlan-ccc` between the nodes to form a pseudowire connection.

The forwarding data path is not changed with the new `vlan-ccc` encapsulation on the transport PS interface and the behavior similar to that when the `ethernet-ccc` encapsulation is configured on the transport PS interface. The transport PS interface either encapsulates or de-encapsulate the outer Layer 2 header and MPLS headers on the transmitted or received packets on the WAN port. Inner Ethernet or VLAN headers of the packet are handled on pseudowire client service logical interfaces. You must configure pseudowire client service logical interfaces with appropriate VLAN IDs or VLAN tags.

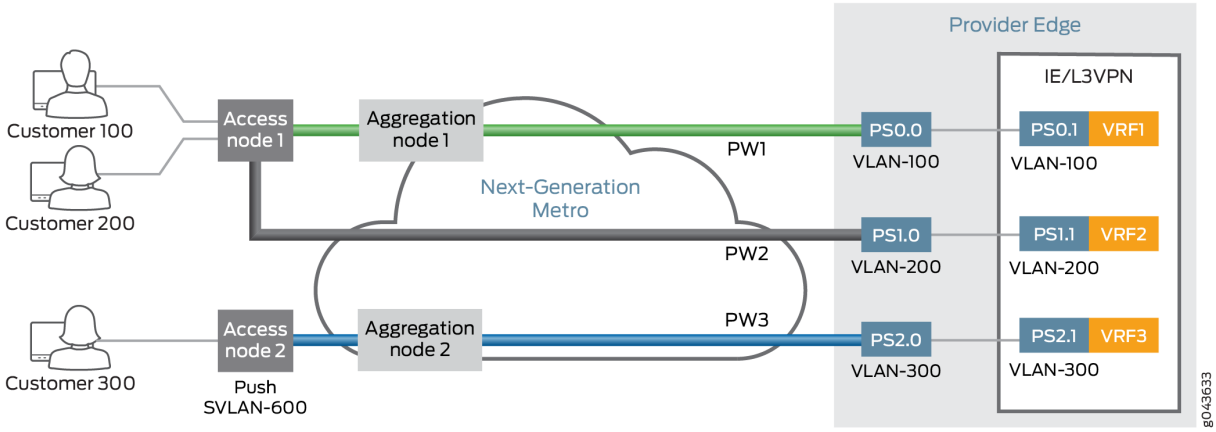
The following sections provides details, along with a sample configuration, about pseudowire configuration from both access and aggregation nodes.

Pseudowire Configuration from Access Node

These pseudowires are set up using VLANs from the access node for customer devices attached to the Layer 2 circuit configured on access and PE routers with customer VLANs (C-VLANs). The ingress traffic (from the access node side) on the PE router is single VLAN tagged (inner Ethernet header), and thus the service logical interfaces must be configured with the same VLAN IDs corresponding to the C-VLAN IDs attached to the access node.

Figure 138 on page 2065 provides the details of a transport PS interface from an access node (access node).

Figure 138: Pseudowire Client Transport Logical Interface from Access Node



The following example shows the configuration of a pseudowire client logical interface configuration on a PE router from an access node:

```
interfaces {
    ps0 {
        anchor-point lt-3;
        unit 0 {
            encapsulation VLAN-ccc;
            VLAN ID 100;
        }
        unit 1 {
            VLAN ID 100;
            family inet;
        }
    }
}
```

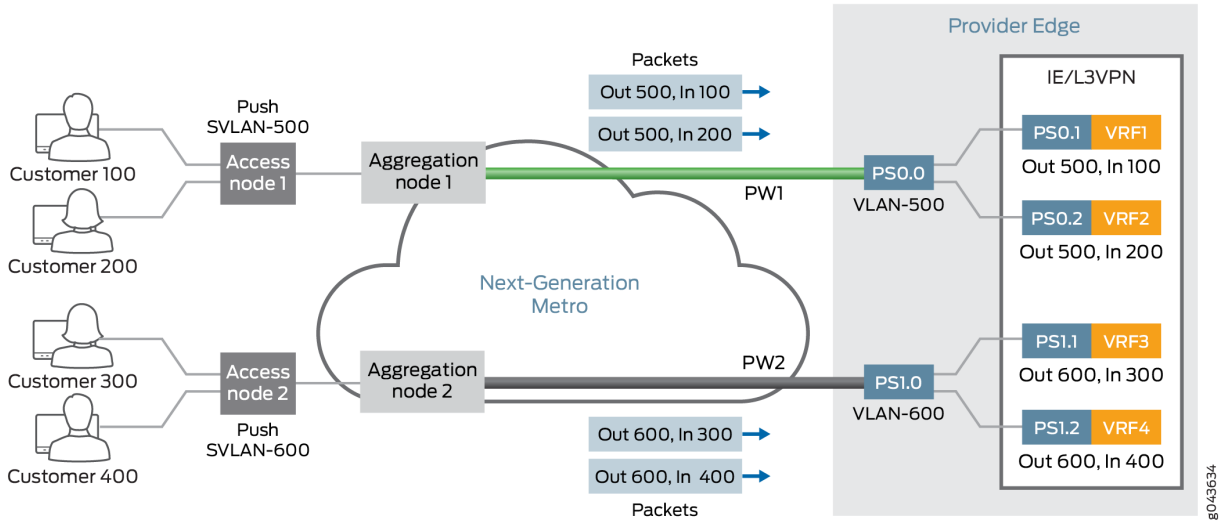
Pseudowire Configuration from Aggregation Node

In this case, the aggregation node processes a stacked VLAN (also known as Q-in-Q). The pseudowire originates from aggregation node and terminates on a PE router. The aggregation node pushes the service VLAN (S-VLAN) tag, and the PE router is expected to operate on two VLAN tags—the outer VLAN tag corresponds to an S-VLAN and the inner VLAN tag corresponds to a C-VLAN. The VLAN ID

configured on the transport PS interface at the PE router must match the VLAN tag of the S-VLAN. On the pseudowire client service logical interface, the outer VLAN tag must be configured to match the S-VLAN and the inner VLAN tag must be configured to match the C-VLAN.

Figure 139 on page 2066 provides the details of a transport PS interface from an aggregation node.

Figure 139: Pseudowire Client Transport Logical Interface from Aggregation Node



The following example shows the configuration of a pseudowire client logical interface configuration on a PE router from an aggregation node:

```
interfaces {
    ps0 {
        anchor-point lt-3;
        unit 0 {
            encapsulation VLAN-ccc;
            VLAN ID 500;
        }
        unit 1 {
            VLAN tags  {
                outer 500;
                inner 100;
            }
        }
        unit 2 {
            VLAN tags  {
                outer 500;
```

```
                inner 200;
            }
        }
    }
}
```

Transmitting Nonstandard BPDUs

CCC protocol (and Layer 2 Circuit and Layer 2 VPN) configurations can transmit nonstandard bridge protocol data units (BPDUs) generated by other vendors' equipment. This is the default behavior on all supported PICs and requires no additional configuration.

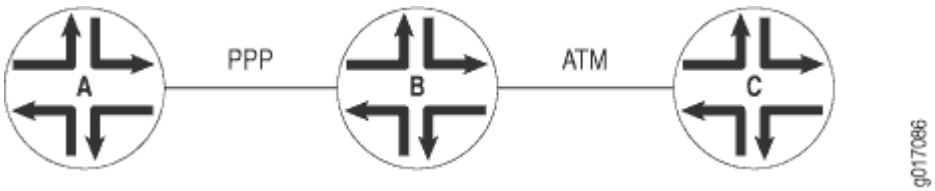
The following PICs are supported on M320 and T Series routers:

- 1-port Gigabit Ethernet PIC
- 2-port Gigabit Ethernet PIC
- 4-port Gigabit Ethernet PIC
- 10-port Gigabit Ethernet PIC

TCC Overview

Translational cross-connect (TCC) is a switching concept that enables you to establish interconnections between a variety of Layer 2 protocols or circuits. It is similar to CCC. However, whereas CCC requires the same Layer 2 encapsulations on each side of a Juniper Networks router (such as PPP-to-PPP or Frame Relay-to-Frame Relay), TCC enables you to connect different types of Layer 2 protocols interchangeably. When you use TCC, combinations such as PPP-to-ATM (see Figure 140 on page 2067) and Ethernet-to-Frame Relay connections are possible.

Figure 140: TCC Example



The Layer 2 circuits and encapsulation types that can be interconnected by TCC are:

- Ethernet
- Extended VLANs

- PPP
- HDLC
- ATM
- Frame Relay

TCC works by removing the Layer 2 header when frames enter the router and adding a different Layer 2 header on the frames before they leave the router. In Figure 140 on page 2067, the PPP encapsulation is stripped from the frames arriving at Router B, and the ATM encapsulation is added before the frames are sent to Router C.

Note that all control traffic is terminated at the interconnecting router (Router B). Examples of traffic controllers include the Link Control Protocol (LCP) and the Network Control Protocol (NCP) for PPP, keepalives for HDLC, and Local Management Interface (LMI) for Frame Relay.

TCC functionality is different from standard Layer 2 switching. TCC only swaps Layer 2 headers. No other processing, such as header checksums, TTL decrementing, or protocol handling is performed. TCC is supported for IPv4 only.

Address Resolution Protocol (APR) packet policing on TCC Ethernet interfaces is effective for releases 10.4 and onwards.

You can configure TCC for interface switching and for Layer 2 VPNs. For more information about using TCC for virtual private networks (VPNs), see the Junos OS VPNs Library for Routing Devices.

Configuring Layer 2 Switching Cross-Connects Using CCC

IN THIS SECTION

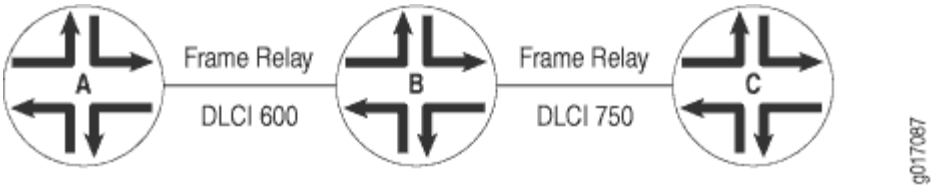
- Configuring the CCC Encapsulation for Layer 2 Switching Cross-Connects | **2069**
- Configuring the CCC Connection for Layer 2 Switching Cross-Connects | **2074**
- Configuring MPLS for Layer 2 Switching Cross-Connects | **2075**
- Example: Configuring a Layer 2 Switching Cross-Connect | **2075**
- Configuring Layer 2 Switching Cross-Connect on ACX5440 | **2078**

Layer 2 switching cross-connects join logical interfaces to form what is essentially Layer 2 switching. The interfaces that you connect must be of the same type.

Figure 141 on page 2069 illustrates a Layer 2 switching cross-connect. In this topology, Router A and Router C have Frame Relay connections to Router B, which is a Juniper Networks router. Circuit cross-connect (CCC) allows you to configure Router B to act as a Frame Relay (Layer 2) switch.

To configure Router B to act as a Frame Relay switch, you configure a circuit from Router A to Router C that passes through Router B, effectively configuring Router B as a Frame Relay switch with respect to these routers. This configuration allows Router B to transparently switch packets (frames) between Router A and Router C without regard to the packets' contents or the Layer 3 protocols. The only processing that Router B performs is to translate DLCI 600 to 750.

Figure 141: Layer 2 Switching Cross-Connect



If the Router A–to–Router B and Router B–to–Router C circuits were PPP, for example, the Link Control Protocol and Network Control Protocol exchanges occur between Router A and Router C. These messages are handled transparently by Router B, allowing Router A and Router C to use various PPP options (such as header or address compression and authentication) that Router B might not support. Similarly, Router A and Router C exchange keepalives, providing circuit-to-circuit connectivity status.

You can configure Layer 2 switching cross-connects on PPP, Cisco HDLC, Frame Relay, Ethernet, and ATM circuits. In a single cross-connect, only like interfaces can be connected.

To configure Layer 2 switching cross-connects, you must configure the following on the router that is acting as the switch (Router B in Figure 141 on page 2069):

Configuring the CCC Encapsulation for Layer 2 Switching Cross-Connects

To configure Layer 2 switching cross-connects, configure the CCC encapsulation on the router that is acting as the switch (Router B in Figure 141 on page 2069).

> **NOTE**: You cannot configure families on CCC interfaces; that is, you cannot include the `family` statement at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`*`]` hierarchy level.

For instructions for configuring the encapsulation for Layer 2 switching cross-connects, see the following sections:

Configuring ATM Encapsulation for Layer 2 Switching Cross-Connects

For ATM circuits, specify the encapsulation when configuring the virtual circuit (VC). Configure each VC as a circuit or a regular logical interface by including the following statements:

```
at-fpc/pic/port {
    atm-options {
        vpi vpi-identifier maximum-vcs maximum-vcs;
    }
    unit logical-unit-number {
        encapsulation encapsulation-type;
        point-to-point; # Default interface type
        vci vpi-identifier.vci-identifier;
    }
}
```

You can include these statements at the following hierarchy levels:

- [edit interfaces]
- [edit logical-systems logical-system-name interfaces]

Configuring Ethernet Encapsulation for Layer 2 Switching Cross-Connects

For Ethernet circuits, specify ethernet-ccc in the encapsulation statement. This statement configures the entire physical device. For these circuits to work, you must also configure a logical interface (unit 0).

Ethernet interfaces with standard Tag Protocol Identifier (TPID) tagging can use Ethernet CCC encapsulation. On M Series Multiservice Edge Routers, except the M320, one-port Gigabit Ethernet, two-port Gigabit Ethernet, four-port Gigabit Ethernet, and four-port Fast Ethernet PICs can use Ethernet CCC encapsulation. On T Series Core Routers and M320 routers, one-port Gigabit Ethernet and two-port Gigabit Ethernet PICs installed in FPC2 can use Ethernet CCC encapsulation. When you use this encapsulation type, you can configure the ccc family only.

```
fe-fpc/pic/port {
    encapsulation ethernet-ccc;
    unit 0;
}
```

You can include these statements at the following hierarchy levels:

- [edit interfaces]

- [edit logical-systems *logical-system-name* interfaces]

Configuring Ethernet VLAN Encapsulation for Layer 2 Switching Cross-Connects

An Ethernet virtual LAN (VLAN) circuit can be configured using either the vlan-ccc or extended-vlan-ccc encapsulation. If you configure the extended-vlan-ccc encapsulation on the physical interface, you cannot configure the inet family on the logical interfaces. Only the ccc family is allowed. If you configure the vlan-ccc encapsulation on the physical interface, both the inet and ccc families are supported on the logical interfaces. Ethernet interfaces in VLAN mode can have multiple logical interfaces.

For encapsulation type vlan-ccc, VLAN IDs from 512 through 4094 are reserved for CCC VLANs. For the extended-vlan-ccc encapsulation type, all VLAN IDs 1 and higher are valid. VLAN ID 0 is reserved for tagging the priority of frames.

NOTE: Some vendors use the proprietary TPIDs 0x9100 and 0x9901 to encapsulate a VLAN-tagged packet into a VLAN-CCC tunnel to interconnect a geographically separated metro Ethernet network. By configuring the extended-vlan-ccc encapsulation type, a Juniper Networks router can accept all three TPIDs (0x8100, 0x9100, and 0x9901).

Configure an Ethernet VLAN circuit with the vlan-ccc encapsulation as follows:

```
interfaces {
    type-fpc/pic/port {
        vlan-tagging;
        encapsulation vlan-ccc;
        unit logical-unit-number {
            encapsulation vlan-ccc;
            vlan-id vlan-id;
        }
    }
}
```

You can configure these statements at the following hierarchy levels:

- [edit interfaces]
- [edit logical-systems *logical-system-name* interfaces]

Configure an Ethernet VLAN circuit with the `extended-vlan-ccc` encapsulation statement as follows:

```
interfaces {
    type-fpc/pic/port {
        vlan-tagging;
        encapsulation extended-vlan-ccc;
        unit logical-unit-number {
            vlan-id vlan-id;
            family ccc;
        }
    }
}
```

You can configure these statements at the following hierarchy levels:

- `[edit interfaces]`
- `[edit logical-systems logical-system-name interfaces]`

Whether you configure the encapsulation as `vlan-ccc` or `extended-vlan-ccc`, you must enable VLAN tagging by including the `vlan-tagging` statement.

Configuring Aggregated Ethernet Encapsulation for Layer 2 Switching Cross-Connects

You can configure aggregated Ethernet interfaces for CCC connections and for Layer 2 virtual private networks (VPNs).

Aggregated Ethernet interfaces configured with VLAN tagging can be configured with multiple logical interfaces. The only encapsulation available for aggregated Ethernet logical interfaces is `vlan-ccc`. When you configure the `vlan-id` statement, you are limited to VLAN IDs 512 through 4094.

Aggregated Ethernet interfaces configured without VLAN tagging can be configured only with the `ethernet-ccc` encapsulation. All untagged Ethernet packets received are forwarded based on the CCC parameters.

To configure aggregated Ethernet interfaces for CCC connections, include the `ae0` statement at the `[edit interfaces]` hierarchy level:

```
[edit interfaces]
ae0 {
    encapsulation (ethernet-ccc | extended-vlan-ccc | vlan-ccc);
    vlan-tagging;
    aggregated-ether-options {
```

```
        minimum-links links;
        link-speed speed;
    }
    unit logical-unit-number {
        encapsulation vlan-ccc;
        vlan-id identifier;
        family ccc;
    }
}
```

Be aware of the following limitations when configuring CCC connections over aggregated Ethernet interfaces:

- If you configured load balancing between child links, be aware that a different hash key is used to distribute packets among the child links. Standard aggregated interfaces have family inet configured. An IP version 4 (IPv4) hash key (based on the Layer 3 information) is used to distribute packets among the child links. A CCC connection over an aggregated Ethernet interface has family ccc configured instead. Instead of an IPv4 hash key, an MPLS hash key (based on the destination media access control [MAC] address) is used to distributed packets among the child links.
- The extended-vlan-ccc encapsulation is not supported on the 12-port Fast Ethernet PIC and the 48-port Fast Ethernet PIC.
- The Junos OS does not support the Link Aggregation Control Protocol (LACP) when an aggregated interface is configured as a VLAN (with vlan-ccc encapsulation). LACP can be configured only when the aggregated interface is configured with the ethernet-ccc encapsulation.

For more information about how to configure aggregated Ethernet interfaces, see the Junos OS Network Interfaces Library for Routing Devices.

Configuring Frame Relay Encapsulation for Layer 2 Switching Cross-Connects

For Frame Relay circuits, specify the encapsulation when configuring the DLCI. Configure each DLCI as a circuit or a regular logical interface. The DLCI for regular interfaces must be from 1 through 511. For CCC interfaces, it must be from 512 through 4094.

```
interfaces {
    type-fpc/pic/port {
        unit logical-unit-number {
            dlci dlci-identifier;
            encapsulation encapsulation-type;
            point-to-point; # Default interface type
        }
```

```
    }
}
```

You can configure these statements at the following hierarchy levels:

- [edit interfaces]
- [edit logical-systems *logical-system-name* interfaces]

Configuring PPP and Cisco HDLC Encapsulation for Layer 2 Switching Cross-Connects

For PPP and Cisco HDLC circuits, specify the encapsulation in the encapsulation statement. This statement configures the entire physical device. For these circuits to work, you must configure a logical interface (unit 0).

```
interfaces type-fpc/pic/port {
    encapsulation encapsulation-type;
    unit 0;
}
```

You can configure these statements at the following hierarchy levels:

- [edit interfaces *type-fpc/pic/port*]
- [edit logical-systems *logical-system-name* interfaces *type-fpc/pic/port*]

Configuring the CCC Connection for Layer 2 Switching Cross-Connects

To configure Layer 2 switching cross-connects, define the connection between the two circuits by including the interface-switch statement. You configure this connection on the router that is acting as the switch (Router B in Figure 141 on page 2069). The connection joins the interface that comes from the circuit's source to the interface that leads to the circuit's destination. When you specify the interface names, include the logical portion of the name, which corresponds to the logical unit number. The cross-connect is bidirectional, so packets received on the first interface are transmitted out the second interface, and those received on the second interface are transmitted out the first.

```
interface-switch connection-name {
    interface interface-name.unit-number;
    interface interface-name.unit-number;
}
```

You can include this statement at the following hierarchy levels:

- [edit protocols connections]
- [edit logical-systems *logical-system-name* protocols connections]

Configuring MPLS for Layer 2 Switching Cross-Connects

For Layer 2 switching cross-connects to work, you must enable MPLS on the router by including at least the following statements. This minimum configuration enables MPLS on a logical interface for the switching cross-connect.

Include the `family mpls` statement:

```
family mpls;
```

You can configure this statement at the following hierarchy levels:

- [edit interfaces *interface-name* unit *logical-unit-number*]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]

You can then specify this logical interface in the MPLS protocol configuration:

```
mpls {
    interface interface-name; # Required to enable MPLS on the interface
}
```

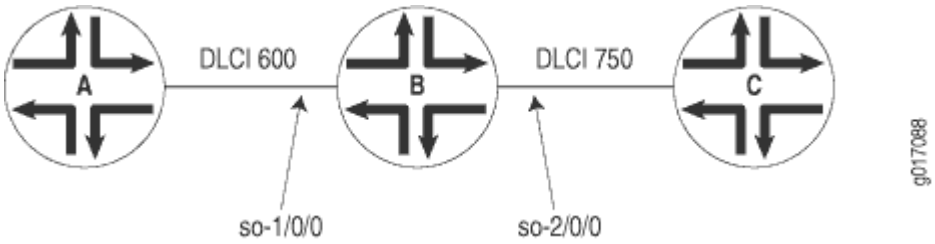
You can configure these statements at the following hierarchy levels:

- [edit protocols]
- [edit logical-systems *logical-system-name* protocols]

Example: Configuring a Layer 2 Switching Cross-Connect

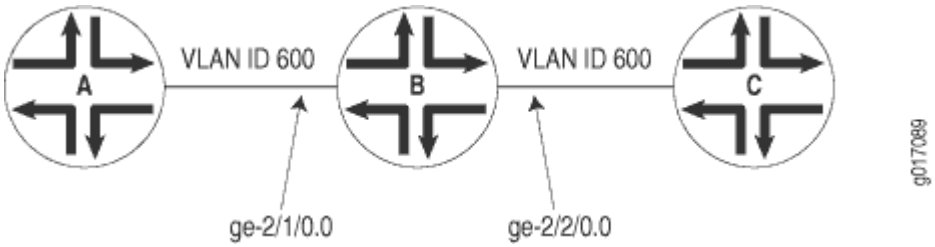
Configure a full-duplex Layer 2 switching cross-connect between Router A and Router C, using a Juniper Networks router, Router B, as the virtual switch. See the topology in Figure 142 on page 2076 and Figure 143 on page 2077.

Figure 142: Topology of a Frame Relay Layer 2 Switching Cross-Connect




```
[edit]
interfaces {
    so-1/0/0 {
        encapsulation frame-relay-ccc;
        unit 1 {
            point-to-point;
            encapsulation frame-relay-ccc;
            dlci 600;
        }
    }
    so-2/0/0 {
        encapsulation frame-relay-ccc;
        unit 2 {
            point-to-point;
            encapsulation frame-relay-ccc;
            dlci 750;
        }
    }
}
protocols {
    connections {
        interface-switch router-a-to-router-c {
            interface so-1/0/0.1;
            interface so-2/0/0.2;
        }
    }
    mpls {
        interface all;
    }
}
```

Figure 143: Sample Topology of a VLAN Layer 2 Switching Cross-Connect




```
[edit]
interfaces {
    ge-2/1/0 {
        vlan-tagging;
        encapsulation vlan-ccc;
        unit 0 {
            encapsulation vlan-ccc;
            vlan-id 600;
        }
    }
    ge-2/2/0 {
        vlan-tagging;
        encapsulation vlan-ccc;
        unit 0 {
            encapsulation vlan-ccc;
            vlan-id 600;
        }
        unit 1 {
            family inet {
                vlan-id 1;
                address 10.9.200.1/24;
            }
        }
    }
}
protocols {
    mpls {
        interface all;
    }
    connections {
        interface-switch layer2-sw {
            interface ge-2/1/0.0;
            interface ge-2/2/0.0;
```

```
        }
    }
}
```

Configuring Layer 2 Switching Cross-Connect on ACX5440

Starting in Junos OS Release 19.3R1, you can leverage the hardware support available for cross-connects on the ACX5448 device with the Layer 2 local switching functionality using certain models. With this support, you can provide the EVP and Ethernet Virtual Private Line (EVPL) services..

Local-switching with the following forwarding models are supported:

- VLAN-CCC (logical interface-level local-switching) without any map.
- VLAN-CCC (logical interface-level local-switching) with the following vlan-maps:
  - Push 0x8100.pushVLAN (QinQ type)
  - Swap 0x8100.swapVLAN
- Aggregated Ethernet (AE) static interfaces.
- AE interfaces with LACP, load-balance all active mode.
- Local-switching end-interface support for AE or LAG interface (one non-AE interface and other AE interface).
- Local-switching both interface as AE or LAG interfaces.

To enable Layer 2 local switching on the ACX5448 device, you can use the existing configuration statements for Layer 2 circuits. For example,

```
[edit protocols l2circuit]
local-switching {
    interface interface1 {
        end-interface interface3;
        ignore-encapsulation-mismatch;
        ignore-mtu-mismatch;
    }
}
```

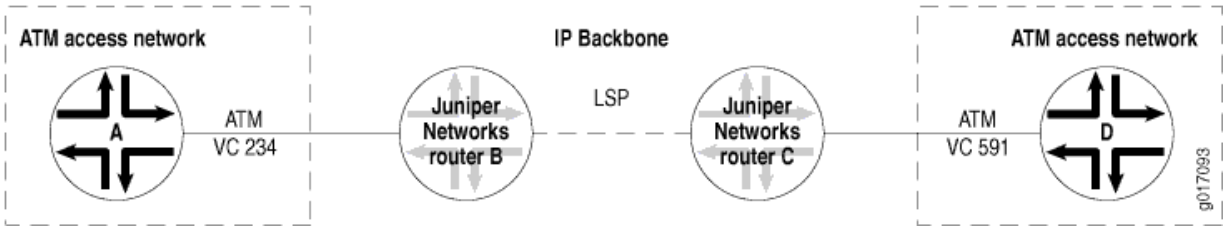
Configuring MPLS LSP Tunnel Cross-Connects Using CCC

IN THIS SECTION

- Configuring the CCC Encapsulation for LSP Tunnel Cross-Connects | **2080**
- Configuring the CCC Connection for LSP Tunnel Cross-Connects | **2082**
- Example: Configuring an LSP Tunnel Cross-Connect | **2083**

MPLS tunnel cross-connects between interfaces and LSPs allow you to connect two distant interface circuits of the same type by creating MPLS tunnels that use LSPs as the conduit. The topology in Figure 144 on page 2079 illustrates an MPLS LSP tunnel cross-connect. In this topology, two separate networks, in this case ATM access networks, are connected through an IP backbone. CCC allows you to establish an LSP tunnel between the two domains. With LSP tunneling, you tunnel the ATM traffic from one network across a SONET backbone to the second network by using an MPLS LSP.

Figure 144: MPLS Tunnel Cross-Connect



When traffic from Router A (VC 234) reaches Router B, it is encapsulated and placed into an LSP, which is sent through the backbone to Router C. At Router C, the label is removed, and the packets are placed onto the ATM permanent virtual circuit (PVC) (VC 591) and sent to Router D. Similarly, traffic from Router D (VC 591) is sent over an LSP to Router B, then placed on VC 234 to Router A.

You can configure LSP tunnel cross-connect on PPP, Cisco HDLC, Frame Relay, and ATM circuits. In a single cross-connect, only like interfaces can be connected.

When you use MPLS tunnel cross-connects to support IS-IS, you must ensure that the LSP's maximum transmission unit (MTU) can, at a minimum, accommodate a 1492-octet IS-IS protocol data unit (PDU) in addition to the link-level overhead associated with the technology being connected.

For the tunnel cross-connects to work, the IS-IS frame size on the edge routers (Routers A and D in Figure 145 on page 2083) must be smaller than the LSP's MTU.

NOTE: Frame size values do not include the frame check sequence (FCS) or delimiting flags.

To determine the LSP MTU required to support IS-IS, use the following calculation:

```
IS-IS MTU (minimum 1492, default 1497) + frame overhead + 4 (MPLS shim header) = Minimum LSP MTU
```

The framing overhead varies based on the encapsulation being used. The following lists the IS-IS encapsulation overhead values for various encapsulations:

- ATM
  - AAL5 multiplex—8 bytes (RFC 1483)
  - VC multiplex—0 bytes
- Frame Relay
  - Multiprotocol—2 bytes (RFCs 1490 and 2427)
  - VC multiplex—0 bytes
- HDLC—4 bytes
- PPP—4 bytes
- VLAN—21 bytes (802.3/LLC)

For IS-IS to work over VLAN-CCC, the LSP's MTU must be at least 1513 bytes (or 1518 for 1497-byte PDUs). If you increase the size of a Fast Ethernet MTU above the default of 1500 bytes, you might need to explicitly configure jumbo frames on intervening equipment.

To modify the MTU, include the `mtu` statement when configuring the logical interface family at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`* `encapsulation` *`family`*`]` hierarchy level. For more information about setting the MTU, see the Junos OS Network Interfaces Library for Routing Devices.

To configure an LSP tunnel cross-connect, you must configure the following on the interdomain router (Router B in Figure 145 on page 2083):

Configuring the CCC Encapsulation for LSP Tunnel Cross-Connects

To configure LSP tunnel cross-connects, you must configure the CCC encapsulation on the ingress and egress routers (Router B and Router C, respectively, in Figure 145 on page 2083).

NOTE: You cannot configure families on CCC interfaces; that is, you cannot include the `family` statement at the `[edit interfaces interface-name unit logical-unit-number]` hierarchy level.

For PPP or Cisco HDLC circuits, include the `encapsulation` statement to configure the entire physical device. For these circuits to work, you must configure logical unit 0 on the interface.

```
type-fpc/pic/port {
    encapsulation (ppp-ccc | cisco-hdlc-ccc);
    unit 0;
}
```

You can include these statements at the following hierarchy levels:

- `[edit interfaces]`
- `[edit logical-systems logical-system-name interfaces]`

For ATM circuits, specify the encapsulation when configuring the VC by including the following statements. For each VC, you configure whether it is a circuit or a regular logical interface.

```
at-fpc/pic/port {
    atm-options {
        vpi vpi-identifier maximum-vcs maximum-vcs;
    }
    unit logical-unit-number {
        point-to-point; # Default interface type
        encapsulation atm-ccc-vc-mux;
        vci vpi-identifier.vci-identifier;
    }
}
```

You can include these statements at the following hierarchy levels:

- `[edit interfaces]`
- `[edit logical-systems logical-system-name interfaces]`

For Frame Relay circuits, include the following statements to specify the encapsulation when configuring the DLCI. For each DLCI, you configure whether it is a circuit or a regular logical interface. The DLCI for

regular interfaces must be in the range 1 through 511. For CCC interfaces, it must be in the range 512 through 1022.

```
type-fpc/pic/port {
    encapsulation frame-relay-ccc;
    unit logical-unit-number {
        point-to-point; # default interface type
        encapsulation frame-relay-ccc;
        dlci dlci-identifier;
    }
}
```

You can include these statements at the following hierarchy levels:

- [edit interfaces]
- [edit logical-systems *logical-system-name* interfaces]

For more information about the encapsulation statement, see the Junos OS Network Interfaces Library for Routing Devices.

Configuring the CCC Connection for LSP Tunnel Cross-Connects

To configure LSP tunnel cross-connects, include the remote-interface-switch statement to define the connection between the two circuits on the ingress and egress routers (Router B and Router C, respectively, in Figure 145 on page 2083). The connection joins the interface or LSP that comes from the circuit's source to the interface or LSP that leads to the circuit's destination. When you specify the interface name, include the logical portion of the name, which corresponds to the logical unit number. For the cross-connect to be bidirectional, you must configure cross-connects on two routers.

```
remote-interface-switch connection-name {
    interface interface-name.unit-number;
    transmit-lsp label-switched-path;
    receive-lsp label-switched-path;
}
```

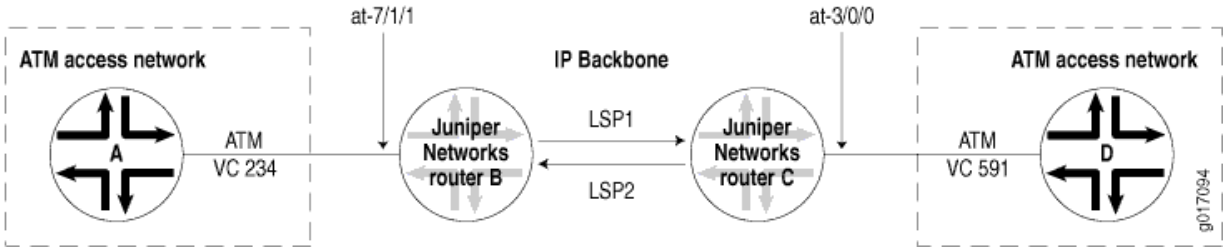
You can include these statements at the following hierarchy levels:

- [edit protocols connections]
- [edit logical-systems *logical-system-name* protocols connections]

Example: Configuring an LSP Tunnel Cross-Connect

Configure a full-duplex MPLS LSP tunnel cross-connect from Router A to Router D, passing through Router B and Router C. See the topology in Figure 145 on page 2083.

Figure 145: Example Topology of MPLS LSP Tunnel Cross-Connect



On Router B:

```
[edit]
interfaces {
    at-7/1/1 {
        atm-options {
            vpi 1 maximum-vcs 600;
        }
        unit 1 {
            point-to-point; # default interface type
            encapsulation atm-ccc-vc-mux;
            vci 1.234;
        }
    }
}
protocols {
    connections {
        remote-interface-switch router-b-to-router-c {
            interface at-7/1/1.1;
            transmit-lsp lsp1;
            receive-lsp lsp2;
        }
    }
}
```

On Router C:

```
[edit]
interfaces {
    at-3/0/0 {
        atm-options {
            vpi 2 maximum-vcs 600;
        }
        unit 2 {
            point-to-point; # default interface type
            encapsulation atm-ccc-vc-mux;
            vci 2.591;
        }
    }
}
protocols {
    connections {
        remote-interface-switch router-b-to-router-c {
            interface at-3/0/0.2;
            transmit-lsp lsp2;
            receive-lsp lsp1;
        }
    }
}
```

Configuring TCC

IN THIS SECTION

- Configuring the Encapsulation for Layer 2 Switching TCCs | **2085**
- Configuring the Connection for Layer 2 Switching TCCs | **2089**
- Configuring MPLS for Layer 2 Switching TCCs | **2090**

This section describes how to configure translational cross-connect (TCC).

To configure TCC, you must perform the following tasks on the router that is acting as the switch:

Configuring the Encapsulation for Layer 2 Switching TCCs

To configure a Layer 2 switching TCC, specify the TCC encapsulation on the desired interfaces of the router that is acting as the switch.

NOTE: You cannot configure standard protocol families on TCC or CCC interfaces. Only the CCC family is allowed on CCC interfaces, and only the TCC family is allowed on TCC interfaces.

For Ethernet circuits and Ethernet extended VLAN circuits, you must also configure the Address Resolution Protocol (ARP). See "Configuring ARP for Ethernet and Ethernet Extended VLAN Encapsulations" on page 2088.

Configuring PPP and Cisco HDLC Encapsulation for Layer 2 Switching TCCs

For PPP and Cisco HDLC circuits, configure the encapsulation type for the entire physical device by specifying the appropriate value for the `encapsulation` statement. For these circuits to work, you must also configure the logical interface `unit 0`.

```
encapsulation (ppp-tcc | cisco-hdlc-tcc);
unit 0{...}
```

You can include these statements at the following hierarchy levels:

- `[edit interfaces` *`interface-name`*`]`
- `[edit logical-systems` *`logical-system-name`* `interfaces` *`interface-name`*`]`

Configuring ATM Encapsulation for Layer 2 Switching TCCs

For ATM circuits, configure the encapsulation type by specifying the appropriate value for the `encapsulation` statement in the virtual circuit (VC) configuration. Specify whether each VC is a circuit or a regular logical interface.

```
atm-options {
    vpi vpi-identifier maximum-vcs maximum-vcs;
}
unit logical-unit-number {
    encapsulation (atm-tcc-vc-mux | atm-tcc-snap);
    point-to-point;
```

```
    vci vpi-identifier.vci-identifier;
}
```

You can include these statements at the following hierarchy levels:

- [edit interfaces at-*fpc*/*pic*/*port*]
- [edit logical-systems *logical-system-name* interfaces at-*fpc*/*pic*/*port*]

Configuring Frame Relay Encapsulation for Layer 2 Switching TCCs

For Frame Relay circuits, configure the encapsulation type by specifying the value `frame-relay-tcc` for the `encapsulation` statement when configuring the data-link connection identifier (DLCI). You configure each DLCI as a circuit or a regular logical interface. The DLCI for regular interfaces must be in the range from 1 through 511, but for TCC and CCC interfaces it must be in the range from 512 through 1022.

```
encapsulation frame-relay-tcc;
unit logical-unit-number {
    dlci dlci-identifier;
    encapsulation frame-relay-tcc;
    point-to-point;
}
```

You can include these statements at the following hierarchy levels:

- [edit interfaces *interface-name*]
- [edit logical-systems *logical-system-name* interfaces *interface-name*]

Configuring Ethernet Encapsulation for Layer 2 Switching TCCs

For Ethernet TCC circuits, configuring the encapsulation type for the entire physical device by specifying the value `ethernet-tcc` for the `encapsulation` statement.

You must also specify static values for a remote address and a proxy address at the [edit interfaces *interface-name* unit *unit-number* family tcc] or [edit logical-systems *logical-system-name* interfaces *interface-name* unit *unit-number* family tcc] hierarchy level.

The remote address is associated with the TCC switching router's Ethernet neighbor; in the `remote` statement you must specify both the IP address and the media access control (MAC) address of the Ethernet neighbor. The proxy address is associated with the TCC router's other neighbor connected by the unlike link; in the `proxy` statement you must specify the IP address of the non-Ethernet neighbor.

You can configure Ethernet TCC encapsulation for the interfaces on 1-port Gigabit Ethernet, 2-port Gigabit Ethernet, 4-port Fast Ethernet, and 4-port Gigabit Ethernet PICs.

```
encapsulation ethernet-tcc;
unit logical-unit-number {
    family tcc {
        proxy {
            inet-address ip-address;
        }
        remote {
            inet-address ip-address;
            mac-address mac-address;
        }
    }
}
```

You can include these statements at the following hierarchy levels:

- `[edit interfaces (fe | ge)-fpc/pic/port]`
- `[edit logical-systems logical-system-name interfaces (fe | ge)-fpc/pic/port]`

NOTE: For Ethernet circuits, you must also configure the Address Resolution Protocol (ARP). See "Configuring ARP for Ethernet and Ethernet Extended VLAN Encapsulations" on page 2088.

Configuring Ethernet Extended VLAN Encapsulation for Layer 2 Switching TCCs

For Ethernet extended VLAN circuits, configure the encapsulation type for the entire physical device by specifying the value `extended-vlan-tcc` for the `encapsulation` statement.

You must also enable VLAN tagging. Ethernet interfaces in VLAN mode can have multiple logical interfaces. With encapsulation type `extended-vlan-tcc`, all VLAN IDs from 0 through 4094 are valid, up to a maximum of 1024 VLANs. As with Ethernet circuits, you must also specify a proxy address and a remote address at the `[edit interfaces interface-name unit logical-unit-number family tcc]` or `[edit logical-systems logical-system-name interfaces interface-name unit unit-number family tcc]` hierarchy level (see "Configuring Ethernet Encapsulation for Layer 2 Switching TCCs" on page 2086).

```
encapsulation extended-vlan-tcc;
vlan-tagging;
unit logical-unit-number {
```

```
    vlan-id identifier;
    family tcc;
    proxy {
        inet-address ip-address;
    }
    remote {
        inet-address ip-address;
        mac-address mac-address;
    }
}
```

You can configure these statements at the following hierarchy levels:

- [edit interfaces *interface-name*]
- [edit logical-systems *logical-system-name* interfaces *interface-name*]

NOTE: For Ethernet extended VLAN circuits, you must also configure the Address Resolution Protocol (ARP). See "Configuring ARP for Ethernet and Ethernet Extended VLAN Encapsulations" on page 2088.

Configuring ARP for Ethernet and Ethernet Extended VLAN Encapsulations

For Ethernet and Ethernet extended VLAN circuits with TCC encapsulation, you must also configure ARP. Because TCC simply removes one Layer 2 header and adds another, the default form of dynamic ARP is not supported; you must configure static ARP.

Because remote and proxy addresses are specified on the router performing TCC switching, you must apply the static ARP statement to the Ethernet-type interfaces of the routers that connect to the TCC-switched router. The `arp` statement must specify the IP address and the MAC address of the remotely connected neighbor by use of the unlike Layer 2 protocol on the far side of the TCC switching router.

```
arp ip-address mac mac-address;
```

You can include this statement at the following hierarchy levels:

- [edit interfaces *interface-name* unit *logical-unit-number* family inet address *ip-address*]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *ip-address*]

Configuring the Connection for Layer 2 Switching TCCs

You must configure the connection between the two circuits of the Layer 2 switching TCC on the router acting as the switch. The connection joins the interface coming from the circuit's source to the interface leading to the circuit's destination. When you specify the interface names, include the logical portion of the name, which corresponds to the logical unit number. The cross-connect is bidirectional, so packets received on the first interface are transmitted from the second interface, and those received on the second interface are transmitted from the first.

To configure a connection for a local interface switch, include the following statements:

```
interface-switch connection-name {
    interface interface-name.unit-number;
}
lsp-switch connection-name {
    transmit-lsp lsp-number;
    receive-lsp lsp-number;
}
```

You can include these statements at the following hierarchy levels:

- `[edit protocols connections]`
- `[edit logical-systems logical-system-name protocols connections]`

To configure a connection for a remote interface switch, include the following statements:

```
remote-interface-switch connection-name {
    interface interface-name.unit-number;
    interface interface-name.unit-number;
    transmit-lsp lsp-number;
    receive-lsp lsp-number;
}
```

You can include these statements at the following hierarchy levels:

- `[edit protocols connections]`
- `[edit logical-systems logical-system-name protocols connections]`

Configuring MPLS for Layer 2 Switching TCCs

For a Layer 2 switching TCC to work, you must enable MPLS on the router by including at least the following statements. This minimum configuration enables MPLS on a logical interface for the switching cross-connect.

Include the `family mpls` statement:

```
family mpls;
```

You can configure this statement at the following hierarchy levels:

- `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`*`]`
- `[edit logical-systems` *`logical-system-name`* `interfaces` *`interface-name`* `unit` *`logical-unit-number`*`]`

You can then specify this logical interface in the MPLS protocol configuration:

```
mpls {
    interface interface-name; # Required to enable MPLS on the interface
}
```

You can configure these statements at the following hierarchy levels:

- `[edit protocols]`
- `[edit logical-systems` *`logical-system-name`* `protocols]`

NOTE: MPLS LSP link protection does not support TCC.

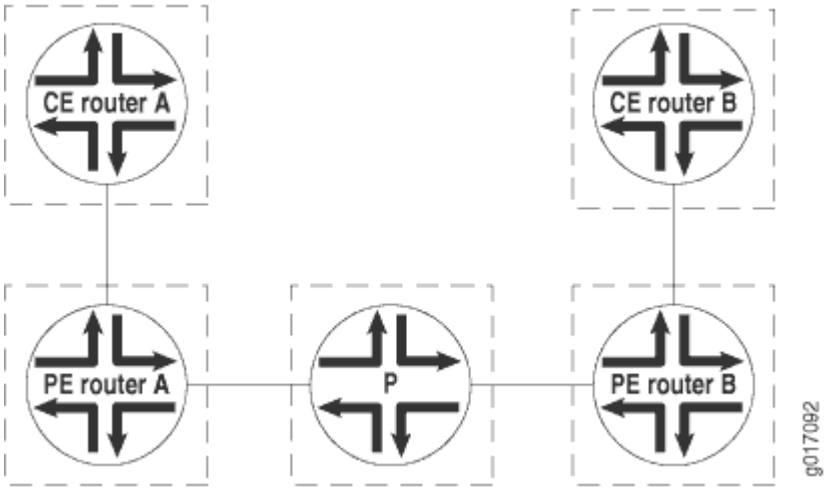
CCC and TCC Graceful Restart

CCC and TCC graceful restart allows Layer 2 connections between customer edge (CE) routers to restart gracefully. These Layer 2 connections are configured with the `remote-interface-switch` or `lsp-switch` statements. Because these CCC and TCC connections have an implicit dependency on RSVP LSPs, graceful restart for CCC and TCC uses the RSVP graceful restart capabilities.

RSVP graceful restart must be enabled on the PE routers and P routers to enable graceful restart for CCC and TCC. Also, because RSVP is used as the signaling protocol for signaling label information, the neighboring router must use helper mode to assist with the RSVP restart procedures.

Figure 146 on page 2091 illustrates how graceful restart might work on a CCC connection between two CE routers.

Figure 146: Remote Interface Switch Connecting Two CE Routers Using CCC



PE Router A is the ingress for the transmit LSP from PE Router A to PE Router B and the egress for the receive LSP from PE Router B to PE Router A. With RSVP graceful restart enabled on all the PE and P routers, the following occurs when PE router A restarts:

- PE Router A preserves the forwarding state associated with the CCC routes (those from CCC to MPLS and from MPLS to CCC).
- Traffic flows without disruption from CE router to CE router.
- After the restart, PE Router A preserves the label for the LSP for which PE Router A is the egress (the receive LSP, for example). The transmit LSP from PE Router A to PE Router B can derive new label mappings, but should not cause any traffic disruption.

Configuring CCC and TCC Graceful Restart

To enable CCC and TCC graceful restart, include the `graceful-restart` statement:

```
graceful-restart;
```

You can include this statement at the following hierarchy levels:

- `[edit routing-options]`
- `[edit logical-systems` *`logical-system-name`* `routing-options]`

Configuring an MPLS-Based VLAN CCC Using the Connection Method (CLI Procedure)

You can configure an 802.1Q VLAN as an MPLS-based connection using EX8200 and EX4500 switches to interconnect multiple customer sites with Layer 2 technology.

This topic describes configuring provider edge (PE) switches in an MPLS network using a circuit cross-connect (CCC) on a tagged VLAN interface (802.1Q VLAN) rather than a simple interface.

NOTE: You do not need to make any changes to existing provider switches in your MPLS network to support this type of configuration. For information on configuring provider switches, see "Configuring MPLS on EX8200 and EX4500 Provider Switches" on page 93.

NOTE: You can send any kind of traffic over a CCC, including nonstandard bridge protocol data units (BPDUs) generated by other vendors' equipment.

NOTE: If you configure a physical interface as VLAN-tagged and with the vlan-ccc encapsulation, you cannot configure the associated logical interfaces with the inet family. Doing so could cause the logical interfaces to drop packets.

To configure a PE switch with a VLAN CCC and an MPLS-based connections:

1. Configure OSPF (or IS-IS) on the loopback (or switch address) and core interfaces:

```
[edit protocols]
user@switch# set ospf area 0.0.0.0 interface lo0.0
user@switch# set ospf area 0.0.0.0 interface interface-name
user@switch# set ospf area 0.0.0.0 interface interface-name
user@switch# set ospf area 0.0.0.0 interface interface-name
```

2. Enable traffic engineering for the routing protocol:

```
[edit protocols]
user@switch# set ospf traffic-engineering
```

3. Configure an IP address for the loopback interface and for the core interfaces:

```
[edit]
user@switch# set interfaces lo0 unit logical-unit-number family inet address address
user@switch# set interfaces interface-name unit logical-unit-number family inet address
address
user@switch# set interfaces interface-name unit logical-unit-number family inet address
address
user@switch# set interfaces interface-name unit logical-unit-number family inet address
address
```

4. Enable the MPLS protocol with **cspf** disabled:

> **NOTE**: CSPF is a shortest-path-first algorithm that has been modified to take into account specific restrictions when the shortest path across the network is calculated. You need to disable CSPF for link protection to function properly on interarea paths.

```
[edit protocols]
user@switch# set mpls no-cspf
```

5. Enable VLAN tagging on the customer edge interface of the local PE switch:

```
[edit]
user@switch# set interfaces interface-name vlan-tagging
```

6. Configure the customer edge interface to use encapsulation **vlan-ccc**:

```
[edit]
user@switch# set interfaces interface-name encapsulation vlan-ccc
```

7. Configure the logical unit of the customer edge interface with a VLAN ID:

> **NOTE**: The VLAN ID cannot be configured on logical interface unit **0**.

The same VLAN ID must be used when configuring the customer edge interface on the other PE switch.

```
[edit ]
user@switch# set interfaces interface-name logical-unit-numberrbhadran vlan-id
```

8. Define the label switched path (LSP):

```
[edit protocols]
user@switch# set mpls label-switched-path lsp-name from address
user@switch# set mpls label-switched-path lsp-name to address
```

TIP: You will need to use the specified LSP name again when configuring the CCC.

9. Configure the connection between the two circuits in the CCC connection

```
[edit protocols]
user@switch# set connections remote-interface-switch interface-switch interface local-
interface
user@switch# set connections remote-interface-switch interface-switch transmit-lsp
destination-lsp
user@switch# set connections remote-interface-switch interface-switch receive-lsp source-lsp
```

Configuring CCC Switching for Point-to-Multipoint LSPs

IN THIS SECTION

- Configuring the Point-to-Multipoint LSP Switch on Ingress PE Routers | **2095**
- Configuring Local Receivers on a Point-to-Multipoint CCC LSP Switch on Ingress PE Routers | **2095**
- Configuring the Point-to-Multipoint LSP Switch on Egress PE Routers | **2096**

You can configure circuit cross-connect (CCC) between two circuits to switch traffic from interfaces to point-to-multipoint LSPs. This feature is useful for handling multicast or broadcast traffic (for example, a digital video stream).

To configure CCC switching for point-to-multipoint LSPs, you do the following:

- On the ingress provider edge (PE) router, you configure CCC to switch traffic from an incoming interface to a point-to-multipoint LSP.
- On the egress PE, you configure CCC to switch traffic from an incoming point-to-multipoint LSP to an outgoing interface.

The CCC connection for point-to-multipoint LSPs is unidirectional.

For more information about point-to-multipoint LSPs, see "Point-to-Multipoint LSPs Overview" on page 781.

To configure a CCC connection for a point-to-multipoint LSP, complete the steps in the following sections:

Configuring the Point-to-Multipoint LSP Switch on Ingress PE Routers

To configure the ingress PE router with a CCC switch for a point-to-multipoint LSP, include the `p2mp-transmit-switch` statement:

```
p2mp-transmit-switch switch-name {
    input-interface input-interface-name.unit-number;
    transmit-p2mp-lsp transmitting-lsp;
}
```

You can include the `p2mp-transmit-switch` statement at the following hierarchy levels:

- `[edit protocols connections]`
- `[edit logical-systems logical-system-name protocols connections]`

switch-name specifies the name of the ingress CCC switch.

`input-interface` *input-interface-name.unit-number* specifies the name of the ingress interface.

`transmit-p2mp-lsp` *transmitting-lsp* specifies the name of the transmitting point-to-multipoint LSP.

Configuring Local Receivers on a Point-to-Multipoint CCC LSP Switch on Ingress PE Routers

In addition to configuring an incoming CCC interface to a point-to-multipoint LSP on an ingress PE router, you can also configure CCC to switch traffic on an incoming CCC interface to one or more outgoing CCC interfaces by configuring output interfaces as local receivers.

To configure output interfaces, include the `output-interface` statement at the `[edit protocols connections p2mp-transmit-switch p2mp-transmit-switch-name]` hierarchy level.

```
[edit protocols connections]
p2mp-transmit-switch pc-ccc {
    input-interface fe-1/3/1.0;
    transmit-p2mp-lsp myp2mp;
    output-interface [fe-1/3/2.0 fe-1/3/3.0];
}
```

You can configure one or more output interfaces as local receivers on the ingress PE router using this statement.

Use the `show connections p2mp-transmit-switch (extensive | history | status)`, `show route ccc <interface-name> (detail | extensive)`, and `show route forwarding-table ccc <interface-name> (detail | extensive)` commands to view details of the local receiving interfaces on the ingress PE router.

Configuring the Point-to-Multipoint LSP Switch on Egress PE Routers

To configure the CCC switch for a point-to-multipoint LSP on the egress PE router, include the `p2mp-receive-switch` statement.

```
p2mp-receive-switch switch-name {
    output-interface [ output-interface-name.unit-number ];
    receive-p2mp-lsp receptive-lsp;
}
```

You can include this statement at the following hierarchy levels:

- `[edit protocols connections]`
- `[edit logical-systems logical-system-name protocols connections]`

switch-name specifies the name of the egress CCC switch.

`output-interface [ output-interface-name.unit-number ]` specifies the name of one or more egress interfaces.

`receive-p2mp-lsp receptive-lsp` specifies the name of the receptive point-to-multipoint LSP.

Configuring an MPLS-Based VLAN CCC Using a Layer 2 VPN (CLI Procedure)

You can configure an 802.1Q VLAN as an MPLS-based Layer 2 virtual private network (VPN) using EX8200 and EX4500 switches to interconnect multiple customer sites with Layer 2 technology.

This topic describes configuring provider edge (PE) switches in an MPLS network using a circuit cross-connect (CCC) on a tagged VLAN interface (802.1Q VLAN) rather than a simple interface.

NOTE: You do not need to make any changes to existing provider switches in your MPLS network to support this type of configuration. For information on configuring provider switches, see "Configuring MPLS on EX8200 and EX4500 Provider Switches" on page 93.

NOTE: You can send any kind of traffic over a CCC, including nonstandard bridge protocol data units (BPDUs) generated by other vendors' equipment.

NOTE: If you configure a physical interface as VLAN-tagged and with the vlan-ccc encapsulation, you cannot configure the associated logical interfaces with the inet family. Doing so could cause the logical interfaces to drop packets.

To configure a PE switch with a VLAN CCC and an MPLS-based Layer 2 VPN:

1. Configure OSPF (or IS-IS) on the loopback (or switch address) and core interfaces:

```
[edit protocols]
user@switch# set ospf area 0.0.0.0 interface lo0.0
user@switch# set ospf area 0.0.0.0 interface interface-name
user@switch# set ospf area 0.0.0.0 interface interface-name
user@switch# set ospf area 0.0.0.0 interface interface-name
```

2. Enable traffic engineering for the routing protocol:

```
[edit protocols]
user@switch# set ospf traffic-engineering
```

3. Configure an IP address for the loopback interface and for the core interfaces:

```
[edit]
user@switch# set interfaces lo0 unit logical-unit-number family inet address address
user@switch# set interfaces interface-name unit logical-unit-number family inet address
address
user@switch# set interfaces interface-name unit logical-unit-number family inet address
```

```
address
user@switch# set interfaces interface-name unit logical-unit-number family inet address
address
```

4. Enable the MPLS protocol with **cspf** disabled:

NOTE: CSPF is a shortest-path-first algorithm that has been modified to take into account specific restrictions when the shortest path across the network is calculated. You need to disable CSPF for link protection to function properly on interarea paths.

```
[edit protocols]
user@switch# set mpls no-cspf
```

5. Define the label switched path (LSP):

```
[edit protocols]
user@switch# set mpls label-switched-path lsp_name to address
```

TIP: You will need to use the specified LSP name again when configuring the CCC.

6. Configure MPLS on the core interfaces:

```
[edit protocols]
user@switch# set mpls interface interface-name
user@switch# set mpls interface interface-name
user@switch# set mpls interface interface-name
```

7. Configure RSVP on the loopback interface and the core interfaces:

```
[edit protocols]
user@switch# set rsvp interface lo0.0
user@switch# set rsvp interface interface-name
user@switch# set rsvp interface interface-name
```

```
user@switch# set rsvp interface interface-name
```

8. Configure **family mpls** on the logical units of the core interfaces:

```
[edit]
user@switch# set interfaces interface-name unit logical-unit-number family mpls
user@switch# set interfaces interface-name unit logical-unit-number family mpls
user@switch# set interfaces interface-name unit logical-unit-number family
mpls
```

NOTE: You can enable **family mpls** on either individual interfaces or aggregated Ethernet interfaces. You cannot enable it on tagged VLAN interfaces.

9. Enable VLAN tagging on the customer edge interface of the local PE switch:

```
[edit]
user@switch# set interfaces interface-name vlan-tagging
```

10. Configure the customer edge interface to use encapsulation **vlan-ccc**:

```
[edit]
user@switch# set interfaces interface-name encapsulation vlan-ccc
```

11. Configure the logical unit of the customer edge interface with a VLAN ID:

NOTE: The VLAN ID cannot be configured on logical interface unit **0**. The logical unit number must be **1** or higher.

The same VLAN ID must be used when configuring the customer edge interface on the other PE switch.

```
[edit ]
user@switch# set interfaces interface-name logical-unit-numbervlan-id vlan-id
```

12. Configure BGP, specifying the loopback address as the local address and enabling **family l2vpn signaling**:

```
[edit protocols bgp]
user@switchPE1# set local-address address family l2vpn signaling
```

13. Configure the BGP group, specifying the group name and type:

```
[edit protocols bgp]
user@switchPE1# set group ibgp type internal
```

14. Configure the BGP neighbor, specifying the loopback address of the remote PE switch as the neighbor's address:

```
[edit protocols bgp]
user@switchPE1# set neighbor address
```

15. Configure the routing instance, specifying the routing-instance name and using **l2vpn** as the instance type:

```
[edit routing-instances]
user@switchPE1# set routing-instance-name instance-type l2vpn
```

16. Configure the routing instance to apply to the customer edge interface:

```
[edit routing-instances]
user@switchPE1# set routing-instance-name interface interface-name
```

17. Configure the routing instance to use a route distinguisher:

```
[edit routing-instances]
user@switchPE1# set routing-instance-name route-distinguisher address
```

18. Configure the VPN routing and forwarding (VRF) target of the routing instance:

```
[edit routing-instances]
user@switchPE1# set routing-instance-name vrf-target community
```

NOTE: You can create more complex policies by explicitly configuring VRF import and export policies using the import and export options. See the *Junos OS VPNs Configuration Guide*.

19. Configure the protocols and encapsulation type used by the routing instance:

```
[edit routing-instances]
user@switchPE1# set routing-instance-name protocols l2vpn encapsulation-type ethernet-vlan
```

20. Apply the routing instance to a customer edge interface and specify a description for it:

```
[edit routing-instances]
user@switchPE1# set routing-instance-name protocols interface interface-name description
description
```

21. Configure the routing-instance protocols site:

```
[edit routing-instances]
user@switchPE1# set routing-instance-name protocols l2vpn site site-name site-identifier
identifier remote-site-id identifier
```

NOTE: The remote site ID (configured with the `remote-site-id` statement) corresponds to the site ID (configured with the `site-identifier` statement) configured on the other PE switch.

When you have completed configuring one PE switch, follow the same procedures to configure the other PE switch.

NOTE: You must use the same type of switch for the other PE switch. You cannot use an EX8200 as one PE switch and use an EX3200 or EX4200 as the other PE switch.

Understanding Ethernet-over-MPLS (L2 Circuit)

IN THIS SECTION

- Ethernet-over-MPLS in Data Centers | 2102

Ethernet-over-MPLS allows sending Layer 2 (L2) Ethernet frames transparently over MPLS. Ethernet-over-MPLS uses a tunneling mechanism for Ethernet traffic through an MPLS-enabled Layer 3 core. It encapsulates Ethernet protocol data units (PDUs) inside MPLS packets and forwards the packets, using label stacking, across the MPLS network This technology has applications in service provider, enterprise and data center environments. For disaster recovery purposes, data centers are hosted in multiple sites that are geographically distant and interconnected using a WAN network.

NOTE: A Layer 2 circuit is similar to a circuit cross-connect (CCC), except that multiple Layer 2 circuits can be transported over a single label-switched path (LSP) tunnel between two provider edge (PE) routers. In contrast, each CCC requires a dedicated LSP.

Ethernet-over-MPLS in Data Centers

For disaster recovery purposes, data centers are hosted in multiple sites that are geographically distant and interconnected using a WAN network. These data centers require L2 connectivity between them for the following reasons:

- To replicate the storage over Fiber Channel IP (FCIP). FCIP works only on the same broadcast domain.
- To run a dynamic routing protocol between the sites.
- To support High Availability clusters that interconnect the nodes hosted in the various data centers.

SEE ALSO

Configuring Ethernet over MPLS (Layer 2 Circuit) | **2043**

Configuring Ethernet over MPLS (Layer 2 Circuit)

IN THIS SECTION

- Configuring the Local PE Switch for Port-Based Layer 2 Circuit (Pseudo-wire) | **2104**
- Configuring the Remote PE Switch for Port-Based Layer 2 Circuit (Pseudo-wire) | **2105**
- Configuring the Local PE Switch for VLAN-Based Layer 2 Circuit | **2106**
- Configuring the Remote PE Switch for VLAN-Based Layer 2 Circuit | **2107**

To implement Ethernet over MPLS, you must configure a Layer 2 circuit on the provider edge (PE) switches. No special configuration is required on the customer edge (CE) switches. The provider switches require MPLS and LDP to be configured on the interfaces that will be receiving and transmitting MPLS packets.

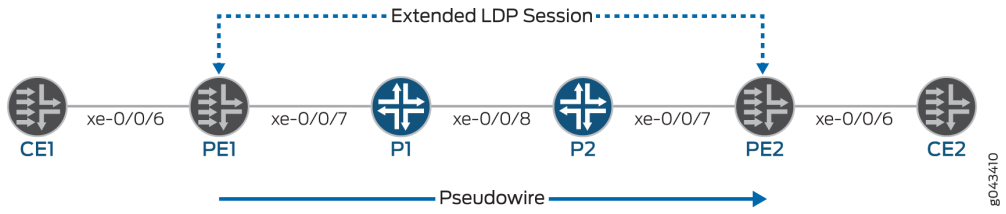
NOTE: A Layer 2 circuit is similar to a circuit cross-connect (CCC), except that multiple Layer 2 circuits can be transported over a single label-switched path (LSP) tunnel between two PE switches. In contrast, each CCC requires a dedicated LSP.

This topic describes how to configure the PE switches to support Ethernet over MPLS. You must configure interfaces and protocols on both the local PE (PE1) and the remote PE (PE2) switches. The interface configuration varies depending upon whether the Layer 2 circuit is port-based or VLAN-based.

Starting in Junos OS Release 20.3R1, support for Layer 2 circuit to provide Layer 2 VPN and VPWS with LDP signaling.

Figure 147 on page 2103 shows an example of a Layer 2 circuit configuration.

Figure 147: Ethernet over MPLS Layer 2 Circuit



NOTE: This topic refers to the local PE switch as PE1 and the remote PE switch as PE2. It also uses interface names rather than variables to help clarify the connections between the switches. The loopback addresses of the switches are configured as follows:

- PE1: 10.127.1.1
- PE2: 10.127.1.2

NOTE: On QFX Series and EX4600 switches, the Layer 2 circuit CE facing interface does not support AE interfaces.

Configuring the Local PE Switch for Port-Based Layer 2 Circuit (Pseudo-wire)

CAUTION: Configure MPLS networks with an MTU (maximum transmission unit) that is at least 12 bytes larger than the largest frame size that will be transported by the LSPs. If the size of a encapsulated packet on the ingress LSR exceeds the LSP MTU, that packet is dropped. If an egress LSR receives a packet on a VC LSP with a length (after the label stack and sequencing control word have been popped) that exceeds the MTU of the destination layer 2 interface, that packet is also dropped.

To configure the local PE switch (PE1) for a port-based layer 2 circuit (pseudo-wire):

1. Configure an access CE-facing interface for Ethernet encapsulation:

```
[edit interfaces]
user@switch# set xe-0/0/6 encapsulation ethernet-ccc
user@switch# set xe-0/0/6 unit 0
```

NOTE: Note that only unit number 0 is supported for Ethernet CCC.

2. Configure the Layer 2 circuit from PE1 to PE2:

```
[edit protocols]
user@switch#set l2circuit neighbor 10.127.1.1 interface xe-0/0/6 virtual-circuit-id 1
```

3. Configure the label switched path from PE1 to PE2:

```
[edit protocols]
user@switch#set mpls label-switched-path PE1-to-PE2 to 10.127.1.1
```

4. Configure the protocols on the core and loopback interfaces:

```
[edit protocols]
user@switch#set mpls interface xe-0/0/7
user@switch#set ldp interface xe-0/0/7
user@switch#set ldp interface lo0.0
```

The changes are shown for the local PE:

```
[edit ]
user@device# show interfaces
xe-0/0/6 {
    encapsulation ethernet-ccc;
    unit 0;
}

[edit]
user@device# show protocols
l2circuit {
    neighbor 10.127.1.1 {
        interface xe-0/0/6.0 {
            virtual-circuit-id 1;
        }
    }
}
ldp {
    interface xe-0/0/7.0;
    interface lo0.0;
}
mpls {
    label-switched-path PE1-to-PE2 {
        to 10.127.1.1;
    }
    interface xe-0/0/7.0;
}
```

Configuring the Remote PE Switch for Port-Based Layer 2 Circuit (Pseudo-wire)

To configure the remote PE switch (PE2) for a port-based layer 2 circuit:

1. Configure an access CE-facing interface for Ethernet encapsulation:

```
[edit interfaces]
user@switch# set xe-0/0/6 encapsulation ethernet-ccc
user@switch# set xe-0/0/6 unit 0
```

2. Configure the Layer 2 circuit from PE2 to PE1:

```
[edit protocols]
user@switch#set l2circuit neighbor 10.127.1.2 interface xe-0/0/6 virtual-circuit-id 1
```

3. Configure the label switched path from PE2 to PE1:

```
[edit protocols]
user@switch#set mpls label-switched-path PE2-to-PE1 to 10.127.1.2
```

4. Configure the protocols on the core and loopback interfaces:

```
[edit protocols]
user@switch#set mpls interface xe-0/0/7
user@switch#set ldp interface xe-0/0/7
user@switch#set ldp interface lo0.0
```

Configuring the Local PE Switch for VLAN-Based Layer 2 Circuit

To configure the local PE switch (PE1) for a VLAN-based layer 2 circuit:

1. Configure an access CE-facing interface for VLAN encapsulation:

```
[edit interfaces]
user@switch# set xe-0/0/6 encapsulation vlan-ccc
```

2. Configure the logical unit of the CE-facing interface for VLAN encapsulation:

```
[edit interfaces]
user@switch# set xe-0/0/6 unit 0 encapsulation vlan-ccc
```

3. Configure the logical unit of the CE-facing interface to belong to family ccc:

```
[edit interfaces]
user@switch# set xe-0/0/6 unit 0 family ccc
```

4. Configure the same interface for VLAN tagging:

```
[edit interfaces]
user@switch# set xe-0/0/6 vlan-tagging
```

5. Configure the VLAN ID of the interface:

```
[edit interfaces]
user@switch# set xe-0/0/6 unit 0 vlan-id 600
```

6. Configure the Layer 2 circuit from PE1 to PE2:

```
[edit protocols]
user@switch#set l2circuit neighbor 10.127.1.1 interface xe-0/0/6 virtual-circuit-id 1
```

7. Configure the label switched path from PE1 to PE2:

```
[edit protocols]
user@switch#set mpls label-switched-path PE1-to-PE2 to 10.127.1.1
```

8. Configure the protocols on the core and loopback interfaces:

```
[edit protocols]
user@switch#set mpls interface xe-0/0/7
user@switch#set ldp interface xe-0/0/7
user@switch#set ldp interface lo0.0
```

Configuring the Remote PE Switch for VLAN-Based Layer 2 Circuit

To configure the remote PE switch (PE2) for a VLAN-based layer 2 circuit:

1. Configure an access CE-facing interface for VLAN encapsulation:

```
[edit interfaces]
user@switch# set xe-0/0/6 encapsulation vlan-ccc
```

2. Configure the logical unit of the CE-facing interface for VLAN encapsulation:

```
[edit interfaces]
user@switch# set xe-0/0/6 unit 0 encapsulation vlan-ccc
```

3. Configure the logical unit of the CE-facing interface to belong to family ccc:

```
[edit interfaces]
user@switch# set xe-0/0/6 unit 0 family ccc
```

4. Configure the same interface for VLAN tagging:

```
[edit interfaces]
user@switch# set xe-0/0/6 vlan-tagging
```

5. Configure the VLAN ID of the interface:

```
[edit interfaces]
user@switch# set xe-0/0/6 unit 0 vlan-id 600
```

6. Configure the Layer 2 circuit from PE2 to PE1:

```
[edit protocols]
user@switch#set l2circuit neighbor 10.127.1.2 interface xe-0/0/6 virtual-circuit-id 1
```

7. Configure the label switched path from PE2 to PE1:

```
[edit protocols]
user@switch#set mpls label-switched-path PE2-to-PE1 to 10.127.1.2
```

8. Configure the protocols on the core and loopback interfaces:

```
[edit protocols]
user@switch#set mpls interface xe-0/0/7
user@switch#set ldp interface xe-0/0/7
user@switch#set ldp interface lo0.0
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
20.3R1	Starting in Junos OS Release 20.3R1, support for Layer 2 circuit to provide Layer 2 VPN and VPWS with LDP signaling.
20.1R1	Starting in Junos OS Release 20.1R1, aggregated Ethernet interfaces support VLAN translational cross-connect (TCC) encapsulation.
19.3R1	Starting in Junos OS Release 19.3R1, you can leverage the hardware support available for cross-connects on the ACX5448 device with the Layer 2 local switching functionality using certain models. With this support, you can provide the EVP and Ethernet Virtual Private Line (EVPL) services.
17.1R1	Support for VPN service as the customer is supported on QFX10000 switches starting with Junos OS Release 17.1R1.

RELATED DOCUMENTATION

Basic MPLS Configuration | 46

PART 9

MPLS for Software Defined Networking (SDN)

- Path Computation Element Protocol (PCEP) | **2111**

CHAPTER 19

Path Computation Element Protocol (PCEP)

IN THIS CHAPTER

- PCEP Configuration | **2111**

PCEP Configuration

IN THIS SECTION

- Path Computation Element Protocol (PCEP) for RSVP LSPs and Segment Routing | **2112**
- Support of the Path Computation Element Protocol for RSVP-TE Overview | **2113**
- Example: Configuring the Path Computation Element Protocol for MPLS RSVP-TE | **2132**
- Example: Configuring Path Computation Element Protocol for MPLS RSVP-TE with Support of PCE-Initiated Point-to-Point LSPs | **2150**
- Configuring Path Computation Element Protocol for MPLS RSVP-TE with Support of PCE-Initiated Point-to-Point LSPs | **2162**
- Example: Configuring Path Computation Element Protocol for MPLS RSVP-TE with Support for PCE-Controlled Point-to-Multipoint LSPs | **2166**
- Understanding Path Computation Element Protocol for MPLS RSVP-TE with Support for PCE-Initiated Point-to-Multipoint LSPs | **2187**
- Enable Segment Routing for the Path Computation Element Protocol | **2192**
- Static Segment Routing Label Switched Path | **2234**
- Enabling Distributed CSPF for Segment Routing LSPs | **2273**
- Example: Configuring CoS-Based Forwarding and Policy-Based Routing For SR-TE LSPs | **2281**
- Enable Multiple Paths for SR-TE LSPs in PCEP | **2291**
- Enable Transport Layer Security for PCEP Sessions | **2295**
- Reporting Path Optimization and Computed Metrics in PCEP | **2301**
- SRv6-TE Tunnels with micro-SIDs in PCEP | **2308**

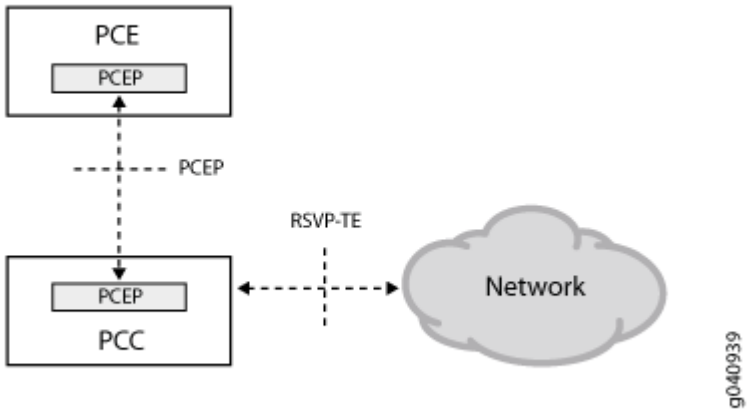
Path Computation Element Protocol (PCEP) for RSVP LSPs and Segment Routing

A Path Computation Element (PCE) is an entity (component, application, or network node) that is capable of computing a network path or route based on a network graph and applying computational constraints. A Path Computation Client (PCC) is any client application requesting a path computation to be performed by a PCE. The Path Computation Element Protocol (PCEP) enables communications between a PCC and a PCE, or between two PCEs (defined in RFC 5440).

PCEP is a TCP-based protocol defined by the IETF PCE Working Group, and defines a set of messages and objects used to manage PCEP sessions and to request and send paths for multidomain traffic engineered LSPs (TE LSPs). It provides a mechanism for a PCE to perform path computation for a PCC's external LSPs. The PCEP interactions include LSP status reports sent by the PCC to the PCE, and PCE updates for the external LSPs.

Figure 148 on page 2112 illustrates the role of PCEP in the client-side implementation of a stateful PCE architecture in an MPLS RSVP-TE enabled network.

Figure 148: PCEP Session



A TCP-based PCEP session connects a PCC to an external PCE. The PCC initiates the PCEP session and stays connected to the PCE for the duration of the PCEP session. During the PCEP session, the PCC requests LSP parameters from the stateful PCE. On receiving one or more LSP parameters from the PCE, the PCC re-signals the TE LSP. When the PCEP session is terminated, the underlying TCP connection is closed immediately, and the PCC attempts to re-establish the PCEP session.

Thus, the PCEP functions include:

- LSP tunnel state synchronization between a PCC and a stateful PCE—When an active stateful PCE connection is detected, a PCC tries to delegate all LSPs to this PCE in a procedure called LSP state synchronization. PCEP enables synchronization of the PCC LSP state to the PCE.
- Delegation of control over LSP tunnels to a stateful PCE—An active stateful PCE controls one or more LSP attributes for computing paths, such as bandwidth, path (ERO), and priority (setup and hold). PCEP enables such delegation of LSPs for path computation.
- Stateful PCE control of timing and sequence of path computations within and across PCEP sessions– An active stateful PCE modifies one or more LSP attributes, such as bandwidth, path (ERO), and priority (setup and hold). PCEP communicates these new LSP attributes from the PCE to the PCC, after which the PCC re-signals the LSP in the specified path.

Support of the Path Computation Element Protocol for RSVP-TE Overview

IN THIS SECTION

- Understanding MPLS RSVP-TE | **2114**
- Current MPLS RSVP-TE Limitations | **2115**
- Use of an External Path Computing Entity | **2116**
- Components of External Path Computing | **2117**
- Interaction Between a PCE and a PCC Using PCEP | **2119**
- LSP Behavior with External Computing | **2122**
- Configuration Statements Supported for External Computing | **2124**
- PCE-Controlled LSP Protection | **2125**
- PCE-Controlled LSP ERO | **2125**
- PCE-Controlled Point-to-Multipoint RSVP-TE LSPs | **2125**
- PCE-Initiated Point-to-Point LSPs | **2126**
- PCE-Initiated Bypass LSP | **2127**
- PCE-Initiated Point-to-Multipoint LSPs | **2129**
- SRv6 LSP in PCEP | **2129**
- Benefits of SRv6 LSPs in PCEP | **2129**
- Auto-Bandwidth and PCE-Controlled LSP | **2130**
- TCP-MD5 Authentication for PCEP Sessions | **2130**
- Impact of Client-Side PCE Implementation on Network Performance | **2132**

Understanding MPLS RSVP-TE

Traffic engineering (TE) deals with performance optimization of operational networks, mainly mapping traffic flows onto an existing physical topology. Traffic engineering provides the ability to move traffic flow away from the shortest path selected by the interior gateway protocol (IGP) and onto a potentially less congested physical path across a network.

For traffic engineering in large, dense networks, MPLS capabilities can be implemented because they potentially provide most of the functionality available from an overlay model, in an integrated manner, and at a lower cost than the currently competing alternatives. The primary reason for implementing MPLS traffic engineering is to control paths along which traffic flows through a network. The main advantage of implementing MPLS traffic engineering is that it provides a combination of the traffic engineering capabilities of ATM, along with the class-of-service (CoS) differentiation of IP.

In an MPLS network, data plane information is forwarded using label switching. A packet arriving on a provider edge (PE) router from the customer edge (CE) router has labels applied to it, and it is then forwarded to the egress PE router. The labels are removed at the egress router and it is then forwarded out to the appropriate destination as an IP packet. The label-switching routers (LSRs) in the MPLS domain use label distribution protocols to communicate the meaning of labels used to forward traffic between and through the LSRs. RSVP-TE is one such label distribution protocol that enables an LSR peer to learn about the label mappings of other peers.

When both MPLS and RSVP are enabled on a router, MPLS becomes a client of RSVP. The primary purpose of the Junos OS RSVP software is to support dynamic signaling within label-switched paths (LSPs). RSVP reserves resources, such as for IP unicast and multicast flows, and requests quality-of-service (QoS) parameters for applications. The protocol is extended in MPLS traffic engineering to enable RSVP to set up LSPs that can be used for traffic engineering in MPLS networks.

When MPLS and RSVP are combined, labels are associated with RSVP flows. Once an LSP is established, the traffic through the path is defined by the label applied at the ingress node of the LSP. The mapping of label to traffic is accomplished using different criteria. The set of packets that are assigned the same label value by a specific node belong to the same forwarding equivalence class (FEC), and effectively define the RSVP flow. When traffic is mapped onto an LSP in this way, the LSP is called an LSP tunnel.

LSP tunnels are a way to establish unidirectional label-switched paths. RSVP-TE builds on the RSVP core protocol by defining new objects and modifying existing objects used in the PATH and RESV objects for LSP establishment. The new objects—LABEL-REQUEST object (LRO), RECORD-ROUTE object (RRO), LABEL object, and EXPLICIT-ROUTE object (ERO)—are optional with respect to the RSVP protocol, except for the LRO and LABEL objects, which are both mandatory for establishing LSP tunnels.

In general, RSVP-TE establishes a label-switched path that ensures frame delivery from ingress to egress router. However, with the new traffic engineering capabilities, the following functions are supported in an MPLS domain:

- Possibility to establish a label-switched path using either a full or partial explicit route (RFC 3209).

- Constraint-based LSP establishment over links that fulfill requirements, such as bandwidth and link properties.
- Endpoint control, which is associated with establishing and managing LSP tunnels at the ingress and egress routers.
- Link management, which manages link resources to do resource-aware routing of traffic engineering LSPs and to program MPLS labels.
- MPLS fast reroute (FRR), which manages the LSPs that need protection and assigns backup tunnel information to these LSPs.

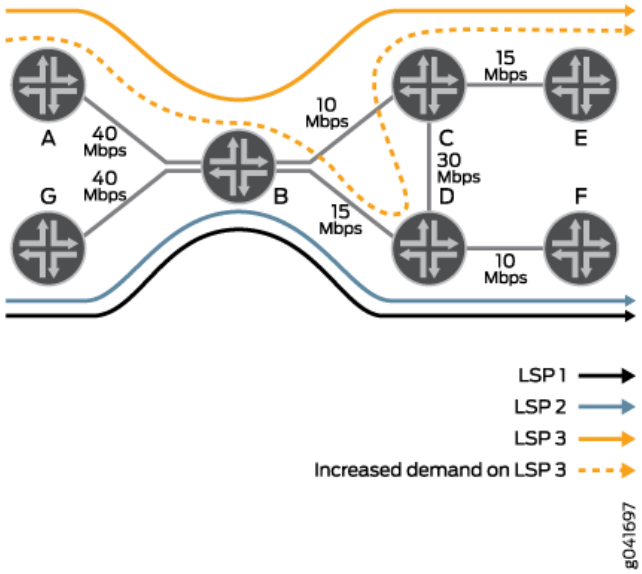
Current MPLS RSVP-TE Limitations

Although the RSVP extensions for traffic engineering enable better network utilization and meet requirements of classes of traffic, today's MPLS RSVP-TE protocol suite has several issues inherent to its distributed nature. This causes a number of issues during contention for bisection capacity, especially within an LSP priority class where a subset of LSPs share common setup and hold priority values. The limitations of RSVP-TE include:

- Lack of visibility of individual per-LSP, per-device bandwidth demands—The ingress routers in an MPLS RSVP-TE network establish LSPs without having a global view of the bandwidth demand on the network. Information about network resource utilization is only available as total reserved capacity by traffic class on a per interface basis. Individual LSP state is available locally on each label edge router (LER) for its own LSPs only. As a result, a number of issues related to demand pattern arise, particularly within a common setup and hold priority.
- Asynchronous and independent nature of RSVP signaling—In RSVP-TE, the constraints for path establishment are controlled by an administrator. As such, bandwidth reserved for an LSP tunnel is set by the administrator and does not automatically imply any limit on the traffic sent over the tunnel. Therefore, bandwidth available on a traffic engineering link is the bandwidth configured for the link, excluding the sum of all reservations made on the link. Thus, the unsignaled demands on an LSP tunnel lead to service degradation of the LSP requiring excess bandwidth, as well as the other LSPs that comply with the bandwidth requirements of the traffic engineering link.
- LSPs established based on dynamic or explicit path options in the order of preference—The ingress routers in an MPLS RSVP-TE network establish LSPs for demands based on the order of arrival. Because the ingress routers do not have a global view of the bandwidth demand on the network, using the order of preference to establish LSPs can cause traffic to be dropped or LSPs not being established at all when there is an excess of bandwidth demand.

As an example, Figure 149 on page 2116 is configured with MPLS RSVP-TE, in which A and G are the label edge routers (LERs). These ingress routers establish LSPs independently based on the order of demands and have no knowledge or control over each other's LSPs. Routers B, C, and D are intermediate or transit routers that connect to the egress routers E and F.

Figure 149: Example MPLS Traffic Engineering



The ingress routers establish LSPs based on the order in which the demands arrive. If Router G receives two demands of capacity 5 each for G-F, then G signals two LSPs – LSP1 and LSP2 – through G-B-D-F. In the same way, when Router A receives the third demand of capacity 10 for A-E, then it signals an LSP, LSP3, through A-B-C-E. However, if the demand on the A-E LSP increases from 10 to 15, Router A cannot signal LSP3 using the same (A-B-C-E) path, because the B-C link has a lower capacity.

Router A should have signaled the increased demand on LSP3 using the A-B-D-C-E path. Since LSP1 and LSP2 have utilized the B-D link based on the order of demands received, LSP3 is not signaled.

Thus, although adequate max-flow bandwidth is available for all the LSPs, LSP3 is subject to potentially prolonged service degradation. This is due to Router A's lack of global demand visibility and the lack of systemic coordination in demand placement by the ingress routers A and G.

Use of an External Path Computing Entity

As a solution to the current limitations found in the MPLS RSVP-TE path computation, an external path computing entity with a global view of per-LSP, per-device demand in the network independent of available capacity is required.

Currently, only online and real-time constraint-based routing path computation is provided in an MPLS RSVP-TE network. Each router performs constraint-based routing calculations independent of the other routers in the network. These calculations are based on currently available topology information—information that is usually recent, but not completely accurate. LSP placements are locally optimized, based on current network status. The MPLS RSVP-TE tunnels are set up using the CLI. An operator configures the TE LSP, which is then signaled by the ingress router.

In addition to the existing traffic engineering capabilities, the MPLS RSVP-TE functionality is extended to include an external path computing entity, called the Path Computation Element (PCE). The PCE computes the path for the TE LSPs of ingress routers that have been configured for external control. The ingress router that connects to a PCE is called a Path Computation Client (PCC). The PCC is configured with the Path Computation Client Protocol (PCEP) to facilitate external path computing by a PCE.

For more information, see "Components of External Path Computing" on page 2117.

To enable external path computing for a PCC's TE LSPs, include the `lsp-external-controller` *pccd* statement at the `[edit mpls]` and `[edit mpls lsp` *lsp-name*`]` hierarchy levels.

Components of External Path Computing

The components that make up an external path computing system are:

Path Computation Element

A Path Computation Element (PCE) can be any entity (component, application, or network node) that is capable of computing a network path or route based on a network graph and applying computational constraints. However, a PCE can compute the path for only those TE LSPs of a PCC that have been configured for external control.

A PCE can either be stateful or stateless.

- Stateful PCE—A stateful PCE maintains strict synchronization between the PCE and network states (in terms of topology and resource information), along with the set of computed paths and reserved resources in use in the network. In other words, a stateful PCE utilizes information from the traffic engineering database as well as information about existing paths (for example, TE LSPs) in the network when processing new requests from the PCC.

  A stateful PCE is of two types:

  - Passive stateful PCE—Maintains synchronization with the PCC and learns the PCC LSP states to better optimize path calculations, but does not have control over them.
  - Active stateful PCE—Actively modifies the PCC LSPs, in addition to learning about the PCC LSP states.

    NOTE: In a redundant configuration with main and backup active stateful PCEs, the backup active stateful PCE cannot modify the attributes of delegated LSPs until it becomes the main PCE at the time of a failover. There is no preempting of PCEs in the case of a switchover. The main PCE is backed by a backup PCE, and when the main PCE goes down, the backup PCE assumes the role of the main PCE and

remains the main PCE even after the PCE that was previously the main PCE is operational again.

A stateful PCE provides the following functions:

- Offers offline LSP path computation.
- Triggers LSP re-route when there is a need to re-optimize the network.
- Changes LSP bandwidth when there is an increase in bandwidth demand from an application.
- Modifies other LSP attributes on the router, such as ERO, setup priority, and hold priority.

A PCE has a global view of the bandwidth demand in the network and maintains a traffic-engineered database to perform path computations. It performs statistics collection from all the routers in the MPLS domain using SNMP and NETCONF. This provides a mechanism for offline control of the PCC's TE LSPs. Although an offline LSP path computation system can be embedded in a network controller, the PCE acts like a full-fledged network controller that provides control over the PCC's TE LSPs, in addition to computing paths.

Although a stateful PCE allows for optimal path computation and increased path computation success, it requires reliable state synchronization mechanisms, with potentially significant control plane overhead and the maintenance of a large amount of data in terms of states, as in the case of a full mesh of TE LSPs.

- Stateless PCE—A stateless PCE does not remember any computed path, and each set of requests is processed independently of each other (RFC 5440).

Path Computation Client

A Path Computation Client (PCC) is any client application requesting a path computation to be performed by a PCE.

A PCC can connect to a maximum of 10 PCEs at one time. The PCC to PCE connection can be a configured static route or a TCP connection that establishes reachability. The PCC assigns each connected PCE a priority number. It sends a message to all the connected PCEs with information about its current LSPs, in a process called LSP state synchronization. For the TE LSPs that have external control enabled, the PCC delegates those LSPs to the main PCE. The PCC elects, as the main PCE, a PCE with the lowest priority number, or the PCE that it connects to first in the absence of a priority number.

The PCC re-signals an LSP based on the computed path it receives from a PCE. When the PCEP session with the main PCE is terminated, the PCC elects a new main PCE, and all delegated LSPs to the previously main PCE are delegated to the newly available main PCE.

Path Computation Element Protocol

The Path Computation Element Protocol (PCEP) is used for communication between PCC and PCE (as well as between two PCEs) (RFC 5440). PCEP is a TCP-based protocol defined by the IETF PCE Working Group, and defines a set of messages and objects used to manage PCEP sessions and to request and send paths for multidomain TE LSPs. The PCEP interactions include PCC messages, as well as notifications of specific states related to the use of a PCE in the context of MPLS RSVP-TE. When PCEP is used for PCE-to-PCE communication, the requesting PCE assumes the role of a PCC.

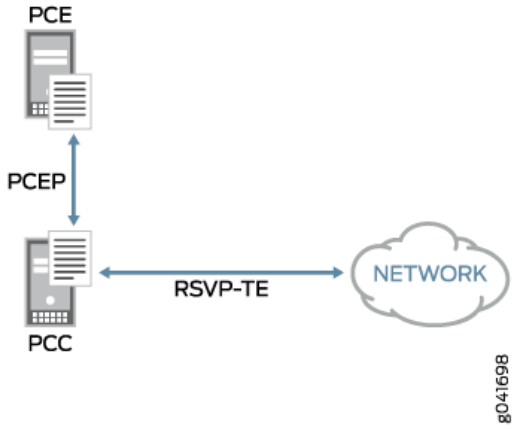
Thus, the PCEP functions include:

- LSP tunnel state synchronization between PCC and a stateful PCE.
- Delegation of control over LSP tunnels to a stateful PCE.

Interaction Between a PCE and a PCC Using PCEP

Figure 150 on page 2119 illustrates the relationship between a PCE, PCC, and the role of PCEP in the context of MPLS RSVP-TE.

Figure 150: PCC and RSVP-TE



The PCE to PCC communication is enabled by the TCP-based PCEP. The PCC initiates the PCEP session and stays connected to a PCE for the duration of the PCEP session.

NOTE: Starting with Junos OS Release 16.1, you can secure a PCEP session using TCP-MD5 authentication as per RFC 5440. To enable the MD5 security mechanism for a PCEP session, it is recommended that you define and bind the MD5 authentication key at the `[edit protocols pcep pce pce-id]` hierarchy level for a PCEP session. You can,

however, also use a predefined keychain from the `[edit security authentication-key-chains key-chain]` hierarchy level to secure a PCEP session. In this case, you should bind the predefined keychain into the PCEP session at the `[edit protocols pcep pce pce-id]` hierarchy level.

The PCE and PCC use the same key to verify the authenticity of each segment sent on the TCP connection of the PCEP session, thereby securing the PCEP communication between the devices, which might be subject to attacks and can disrupt services on the network.

For more information on securing PCEP sessions using MD5 authentication, see "TCP-MD5 Authentication for PCEP Sessions" on page 2130.

Once the PCEP session is established, the PCC performs the following tasks:

1. LSP state synchronization—The PCC sends information about all the LSPs (local and external) to all connected PCEs. For external LSPs, the PCC sends information about any configuration change, RRO change, state change, and so on, to the PCE.

   For PCE-initiated LSPs, there is no LSP configuration present on the PCC. The PCE initiating the LSP sends the LSP parameters to the PCC that has indicated its capability of supporting PCE-initiated LSPs.

   NOTE: Support for PCE-initiated LSPs is provided in Junos OS Release 13.3 and later releases.

2. LSP delegation—After the LSP state information is synchronized, the PCC then delegates the external LSPs to one PCE, which is the main active stateful PCE. Only the main PCE can set parameters for the external LSP. The parameters that the main PCE modifies include bandwidth, path (ERO), and priority (setup and hold). The parameters specified in the local configuration are overridden by the parameters that are set by the main PCE.

   NOTE: When the PCEP session with the main PCE is terminated, the PCC elects a new main PCE, and all delegated LSPs to the previously main PCE are delegated to the newly available main PCE.

   In the case of PCE-initiated LSPs, the PCC creates the LSP using the parameters received from the PCE. The PCC assigns the PCE-initiated LSP a unique LSP-ID, and automatically delegates the LSP to the PCE. A PCC cannot revoke the delegation for the PCE-initiated LSPs for an active PCEP session.

   When a PCEP session terminates, the PCC starts two timers without immediately deleting the PCE-initiated LSPs – **`delegation cleanup timeout`** and **`lsp cleanup timer`** – to avoid disruption of services. During

this time, an active stateful PCE can acquire control of the LSPs provisioned by the failed PCE, by sending a create request for the LSP.

Control over PCE-initiated LSPs reverts to the PCC at the expiration of the **delegation cleanup timeout**. When the **delegation cleanup timeout** expires, and no other PCE has acquired control over the LSP from the failed PCE, the PCC takes local control of the non-delegated PCE-initiated LSP. Later, when the original or a new active stateful PCE wishes to acquire control of the locally controlled PCE-initiated LSPs, the PCC delegates these LSPs to the PCE and the **lsp cleanup timer** timer is stopped.

A PCE may return the delegation of the PCE-initiated LSP to the PCC to allow LSP transfer between PCEs. This triggers the **lsp cleanup timer** for the PCE-initiated LSP. The PCC waits for the LSP cleanup timer to expire before removing the non-delegated PCE-initiated LSPs from the failed PCE.

When the **lsp cleanup timer** expires, and no other PCE has acquired control over the LSPs from the failed PCE, the PCC deletes all the LSPs provisioned by the failed PCE.

NOTE: In compliance with *draft-ietf-pce-stateful-pce-09*, revoking of PCE-initiated LSP delegations by a PCC happens in a make-before-break fashion before the LSPs are redelegated to an alternate PCE. Starting in Junos OS Release 18.1R1, the `lsp-cleanup-timer` must be greater than or equal to the `delegation-cleanup-timeout` for the PCC to revoke the LSP delegations. If not, the redelegation timeout interval for the PCC can be set to infinity, where the LSP delegations to that PCE remain intact until specific action is taken by the PCC to change the parameters set by the PCE.

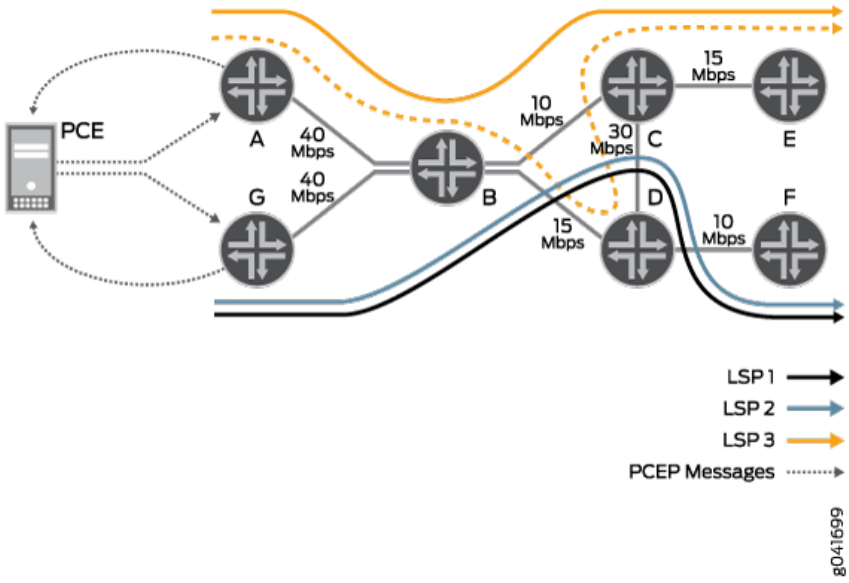
3. LSP signaling—On receiving one or more LSP parameters from the main active stateful PCE, the PCC re-signals the TE LSP based on the PCE provided path. If the PCC fails to set up the LSP, it notifies the PCE of the setup failure and waits for the main PCE to provide new parameters for that LSP, and then re-signals it.

   When the PCE specifies a path that is incomplete or has loose hops where only the path endpoints are specified, the PCC does not perform local constraint-based routing to find out the complete set of hops. Instead, the PCC provides RSVP with the PCE provided path, as it is, for signaling, and the path gets set up using IGP hop-by-hop routing.

Considering the topology used in Figure 149 on page 2116, Figure 151 on page 2122 illustrates the partial client-side PCE implementation in the MPLS RSVP-TE enabled network. The ingress routers A and G are the PCCs that are configured to connect to the external stateful PCE through a TCP connection.

The PCE has a global view of the bandwidth demand in the network and performs external path computations after looking up the traffic engineering database. The active stateful PCE then modifies one or more LSP attributes and sends an update to the PCC. The PCC uses the parameters it receives from the PCE to re-signal the LSP.

Figure 151: Example PCE for MPLS RSVP-TE



This way, the stateful PCE provides a cooperative operation of distributed functionality used to address specific challenges of a shortest interdomain constrained path computation. It eliminates congestion scenarios in which traffic streams are inefficiently mapped onto available resources, causing overutilization of some subsets of network resources, while other resources remain underutilized.

LSP Behavior with External Computing

LSP Types

In a client-side PCE implementation, there are three types of TE LSPs:

- CLI-controlled LSPs—The LSPs that do not have the `lsp-external-controller` *pccd* statement configured are called CLI-controlled LSPs. Although these LSPs are under local control, the PCC updates the connected PCEs with information about the CLI-controlled LSPs during the initial LSP synchronization process. After the initial LSP synchronization, the PCC informs the PCE of any new and deleted LSPs as well.

- PCE-controlled LSPs—The LSPs that have the `lsp-external-controller` *pccd* statement configured are called PCE-controlled LSPs. The PCC delegates the PCC-initiated LSPs to the main PCE for external path computation.

  The PCC informs the PCE about the configured parameters of a PCE-controlled LSP, such as bandwidth, ERO, and priorities. It also informs the PCE about the actual values used for these parameters to set up the LSP including the RRO, when available.

The PCC sends such LSP status reports to the PCE only when a reconfiguration has occurred or when there is a change in the ERO, RRO, or status of the PCE-controlled LSPs under external control.

There are two types of parameters that come from the CLI configuration of an LSP for a PCE:

- Parameters that are not overridden by a PCE, and that are applied immediately.
- Parameters that are overridden by a PCE. These parameters include bandwidth, path, and priority (setup and hold values). When the control mode switches from external to local, the CLI-configured values for these parameters are applied at the next opportunity to re-signal the LSP. The values are not applied immediately.

- Externally-provisioned LSPs (or PCE-initiated LSPs)—The LSPs that have the `lsp-provisioning` statement configured are called PCE-initiated LSPs. A PCE-initiated LSP is dynamically created by an external PCE; as a result, there is no LSP configuration present on the PCC. The PCC creates the PCE-initiated LSP using the parameters provided by the PCE, and automatically delegates the LSP to the PCE.

  NOTE: Support for PCE-initiated LSPs is provided in Junos OS Release 13.3 and later releases.

The CLI-controlled LSPs, PCE-controlled LSPs, and PCE-initiated LSPs can coexist on a PCC.

The CLI-controlled LSPs and PCE-controlled LSPs can coexist on a PCC.

LSP Control Mode

In a client-side PCE implementation, there are two types of control modes for a PCC-controlled LSP:

- External—By default, all PCE-controlled LSPs are under external control. When an LSP is under external control, the PCC uses the PCE-provided parameters to set up the LSP.
- Local—A PCE-controlled LSP can come under local control. When the LSP switches from external control to local control, path computation is done using the CLI-configured parameters and constraint-based routing. Such a switchover happens only when there is a trigger to re-signal the LSP. Until then, the PCC uses the PCE-provided parameters to signal the PCE-controlled LSP, although the LSP remains under local control.

A PCE-controlled LSP switches to local control from its default external control mode in cases such as no connectivity to a PCE or when a PCE returns delegation of LSPs back to the PCC.

For more information about CLI-controlled LSPs and PCE-controlled LSPs, see "LSP Types" on page 2122.

Configuration Statements Supported for External Computing

Table 46 on page 2124 lists the MPLS and existing LSP configuration statements that apply to a PCE-controlled LSP.

Table 46: Applicability of MPLS and Existing LSP Configurations to a PCE-Controlled LSP

Support for PCE-Controlled LSP	Applicable LSP Configuration Statements	Applicable MPLS Configuration Statements
These configuration statements can be configured along with the PCE configuration. However, they take effect only when the local configuration is in use. During PCE control, these configuration statements remain inactive.	• admin-group • auto-bandwidth • hop-limit • least-fill • most-fill • random	• admin-group • admin-groups • admin-group-extended • hop-limit • no-cspf • smart-optimize-timer
These configuration statements can be configured along with the PCE configuration, but are overridden by the PCE-controlled LSP attributes. However, when the local configuration is in use, the configured values for these configuration statements are applied. **NOTE**: Changes to the local configuration using the CLI while the LSP is under the control of a stateful PCE do not have any effect on the LSP. These changes come into effect only when the local configuration is applied.	• bandwidth • primary • priority	• priority
These configuration statements cannot be configured along with the PCE configuration.	• p2mp • template	• p2mp-lsp-next-hop

The rest of the LSP configuration statements are applicable in the same way as for existing LSPs. On configuring any of the above configuration statements for a PCE-controlled LSP, an MPLS log message is generated to indicate when the configured parameters take effect.

PCE-Controlled LSP Protection

The protection paths, including fast reroute and bypass LSPs, are locally computed by the PCC using constraint-based routing. A stateful PCE specifies the primary path (ERO) only. A PCE can also trigger a non-standby secondary path, even if the local configuration does not have a non-standby secondary path for LSP protection.

PCE-Controlled LSP ERO

For PCE-controlled LSPs (PCC-delegated LSPs and PCE-initated LSPs), only a full-blown Explicit Route Object (ERO) object has to be sent from the PCE to the PCC; otherwise the PCC rejects the PCUpdate or PCCreate message for that PCEP session.

Starting in Junos OS Release 17.2, in addition to `external cspf`, two new path computation types are introduced for the PCE-controlled LSPs: `local cspf` and `no cspf`.

- `local cspf`—A PCC uses the `local cspf` computation type only when the PCE sends in a Juniper Vendor TLV (enterprise number: 0x0a4c) of type 5.
- `no cspf`—Neither the PCE nor the PCC performs a constrained path calculation. The endpoints and constraints are given to the RSVP module for setting up the LSP with the IGP path.

  A PCC uses `no cspf` computation type in the following cases:

  - When the PCE sends `local cspf` TLV, and when the Junos OS configuration or matching template for this LSP included `no-cspf` in the PCC-delegated LSP.
  - When the PCE sends `local cspf` TLV, and when the Junos OS configuration template for this LSP included `no-cspf` in the PCE-initiated LSP.
  - When the PCE does not send `local cspf` TLV with an empty ERO or loose ERO (with loose bit set in the ERO object).

With these new computation types, a PCC can accept an ERO object either as a loose ERO, or as an empty ERO. An external path computing entity that is not capable of computing a path can modify parameters such as bandwidth and color, based on the analytics. In such cases, an empty ERO object or loose ERO is used and the path to be taken is decided by the PCC.

PCE-Controlled Point-to-Multipoint RSVP-TE LSPs

After a PCEP session is established between a PCE and a PCC, the PCC reports all the LSPs in the system to the PCE for LSP state synchronization. This includes PCC-controlled, PCE-delegated, and PCE-initiated point-to-point LSPs. Starting with Junos OS Release 15.1F6 and 16.1R1, this capability is extended to report point-to-multipoint LSPs as well. For a PCE, the point-to-multipoint LSP is similar to

that of RSVP point-to-multipoint LSP, where the point-to-multipoint LSP is treated as collection of point-to-point LSPs grouped under a point-to-multipoint identifier.

By default, PCE control of point-to-multipoint LSPs is not supported on a PCC. To add this capability, include the `p2mp-lsp-report-capability` statement at the `[edit protocols pcep pce` *`pce-name`*`]` or `[edit protocols pcep pce-group` *`group-id`*`]` hierarchy levels. After the point-to-multipoint report capability is configured on a PCC, the PCC advertises this capability to the PCE. If the PCE advertises the same point-to-multipoint report capability in return, then the PCC reports the complete point-to-multipoint LSP tree to the PCE for LSP state synchronization.

A PCC with the point-to-multipoint TE LSP capability supports reporting of point-to-multipoint TE LSPs for stateful PCEs, point-to-multipoint update, and LSP database supporting point-to-multipoint LSP name as key. However, the following features and functions are not supported for Junos OS Release 15.1F6 and 16.1:

- Static point-to-multipoint LSPs
- PCE-delegated and PCE-initiated point-to-multipoint LSPs
- Auto-bandwidth
- TE++
- PCE request and reply message
- Creation of point-to-multipoint LSPs using templates
- Configuring forward entry on the PCE-initiated point-to-multipoint LSPs
- Configuring forward entry on the router pointing to a provisioned LSP.

PCE-Initiated Point-to-Point LSPs

Starting with Junos OS Release 16.1, the PCEP functionality is extended to allow a stateful PCE to initiate and provision traffic engineering LSPs through a PCC. Earlier, the LSPs were configured on the PCC and the PCC delegated control over the external LSPs to a PCE. The ownership of the LSP state was maintained by the PCC. With the introduction of the PCE-initiated LSPs, a PCE can initiate and provision a traffic engineering point-to-point LSP dynamically without the need for a locally configured LSP on the PCC. On receiving a PCCreate message from a PCE, the PCC creates the PCE-initiated LSP and automatically delegates the LSP to the PCE.

By default, a PCC rejects the request for provisioning PCE-initiated point-to-point LSPs from a PCE. To enable support of PCE-initated LSPs on the PCC, include the lsp-provisioning statement at the `[edit protocols pcep pce` *`pce-id`*`]` or `[edit protocols pcep pce-group` *`group-id`*`]` hierarchy levels.

A PCC indicates its capability of supporting PCE-initiated point-to-point LSPs while establishing the Path Computation Element Protocol (PCEP) session with the PCE. A PCE selects a PCC with this

capability to initiate an LSP. The PCE provides the PCC with the PCE-initiated LSP parameters. On receiving the PCE-initiated point-to-point LSP parameters, the PCC sets up the LSP, assigns an LSP ID, and automatically delegates the LSP to the PCE.

When the PCE initiating the LSP does not provide the PCE-initiated point-to-point LSP parameters, the PCC uses the default parameters. An optional LSP template may also be configured to specify values for the PCE-initiated point-to-point LSP when the LSP parameters are not provided by the PCE. To configure an LSP template for PCE-initiated point-to-point LSPs on the PCC, include the *label-switched-path-template* statement at the `[edit protocols mpls lsp-external-controller` *`lsp-external-controller`*`]` hierarchy level.

When a PCEP session terminates, the PCC starts two timers without immediately deleting the PCE-initiatedLSPs—`delegation cleanup timeout` and `lsp cleanup timer`—to avoid disruption of services. During this time, an active stateful PCE can acquire control of the LSPs provisioned by the failed PCE.

A PCE may return the delegation of the PCE-initiated point-to-point LSP to the PCC to allow LSP transfer between PCEs. Control over PCE-initiated LSPs reverts to the PCC at the expiration of the delegation cleanup timeout. When the delegation cleanup timeout expires, and no other PCE has acquired control over the LSP from the failed PCE, the PCC takes local control of the non-delegated PCE-initiated LSP. Later, when the original or a new active stateful PCE wishes to acquire control of the locally controlled PCE-initiated point-to-point LSPs, the PCC delegates these LSPs to the PCE and the LSP cleanup timer is stopped.

The PCC waits for the LSP cleanup timer to expire before deleting the non-delegated PCE-initiated point-to-point LSPs from the failed PCE. When the LSP cleanup timer expires, and no other PCE has acquired control over the LSPs from the failed PCE, the PCC deletes all the LSPs provisioned by the failed PCE.

Starting in Junos OS Release 21.1R1, we support nonstop active routing (NSR) for PCE-initiated RSVP-based point-to-point and point-to-multipoint LSPs. Only the primary Routing Engine maintains the PCEP session with the controller. It synchronizes all RSVP LSPs initiated by PCEs, including multicast flow specifications for any PCE-initiated P2MP LSPs, with the backup Routing Engine. During a switchover, the PCEP session goes down and re-establishes when the backup Routing Engine becomes the primary Routing Engine. This reduces traffic loss for the traffic carried over PCE-initiated RSVP LSPs during Routing Engine switchovers. This feature is enabled when NSR is configured.

PCE-Initiated Bypass LSP

Understanding PCE-Initiated Bypass LSPs

There can be traffic outages at the time of a link or node failure because the backup protection paths in the network do not have sufficient bandwidth to handle traffic. In such networks, although a PCE may be used to compute all the paths, to optimize network performance, the local protection paths also need to be controlled through the PCE.

Junos OS Release 19.2R1 and later releases provide partial support for the Internet draft draft-cbrt-pce-stateful-local-protection-01 (expires December 2018), *PCEP Extensions for RSVP-TE Local-Protection with PCE-Stateful*, where the PCEP functionality is extended to allow a stateful PCE to initiate, provision, and manage bypass LSPs for a protected interface. Multiple bypass LSPs with bandwidth reservation can be initiated by the PCE to protect a link or node. The bandwidth on the bypass LSP is expected to be smaller than the total bandwidth of the primary LSPs that it might protect.

The existing bypass selection mechanism, that prefers manual bypass LSPs (if available) over dynamic bypass LSPs, is extended to prefer PCE-provisioned bypass LSPs (if available) over dynamic bypass LSPs. The PCE-provisioned bypass LSPs have a higher preference over dynamic bypass LSPs, but are less preferred over manual bypass LSPs.

The set of operations that are used to perform on any operational bypass LSPs, such as `clear rsvp session`, can also be performed on the PCE-initiated bypass LSPs. You can use commands, such as `show path-computation-client status extensive` and `show path-computation-client lsp` to view PCE-initiated bypass LSP statistics.

With the support of PCE-initiated bypass LSP, you can:

- Create a RSVP bypass LSP through PCEP from an external controller, where the bypass LSP:
  - can be for link or node protection.
  - must have a non-zero bandwidth.
  - must have a specified strict ERO.
- Update the bandwidth and ERO for an existing PCE-created bypass LSP.
- Oversubscribe the bypass LSP bandwidth for admission control of primary LSPs. This must be a per-bypass parameter, and should allow updating the subscription per bypass LSP.

Benefits of PCE-Initiated Bypass LSP

The PCE-initiated bypass LSPs provide the following benefits:

- Better control over traffic after a failure and more deterministic path computation of protection paths.
- Meet complex constraints and diversity requirements, such as maintaining diverse paths for LSPs, as well as their local protection paths.
- Ensure links are not overloaded during failure events.

Behavior of PCE-Initiated Bypass LSPs During PCEP Session Failure

At the time of a PCEP session failure, the PCE-initiated bypass LSPs become orphan until the expiration of the state timeout timer. The PCE-initiated bypass LSPs get cleaned up on the expiration of the state timeout timer. To obtain control of a PCE-initiated bypass LSP (after PCEP session fails), a PCE (either the primary PCE, or any secondary PCE) sends a PCInitiate message before the expiration of the state timeout timer.

PCE-Initiated Point-to-Multipoint LSPs

With the introduction of point-to-multipoint PCE-initiated LSPs, a PCE can initiate and provision a point-to-multipoint LSP dynamically without the need for local LSP configuration on the PCC. This enables the PCE to control the timing and sequence of the point-to-multipoint path computations within and across Path Computation Element Protocol (PCEP) sessions, thereby creating a dynamic network that is centrally controlled and deployed.

For more information, see "Understanding Path Computation Element Protocol for MPLS RSVP-TE with Support for PCE-Initiated Point-to-Multipoint LSPs" on page 2187.

SRv6 LSP in PCEP

Segment routing can be applied to both MPLS and IPv6 forwarding plane. The Path Computation Element (PCE) computes SR paths for both MPLS and IPv6 forwarding plane. Segment routing for PCEP supports SR LSPs such as PCE-initiated, locally created, and delegated SR LSPs in IPv6 forwarding plane.

Benefits of SRv6 LSPs in PCEP

- Allows you to create PCE initiated SRv6 LSPs.
- Delegate the SRv6 LSPs created on the router to the controller.
- Report the LSPs which are locally created on the router to the controller.
- SRv6 network programming provides the flexibility to leverage segment routing without deploying MPLS.

PCEP supports creation, updation, and deletion of PCE initiated colored and uncolored SRv6 LSPs. When PCE initiated SRv6 LSP co-exists along with a static SRv6 LSP for the same IP or color-based IP, then the static SRv6 TE LSP contributing route is preferred over PCE initiated SRv6 TE LSP contributing route.

To configure a PCEP session to be SRv6 capable, you need to enable the `srv6-capability` configuration statement at the [edit protocols pcep pce pce-id] or at the [`edit protocols pcep pce-group pce-id`] hierarchy levels. If the `srv6-capability` configuration statement is enabled, then you must also enable srv6

configuration statement at the [edit protocols source-packet-routing] hierarchy level otherwise during commit and error will be displayed.

To configure SRv6 for SR-TE, you need to add the srv6 configuration statement at the [edit protocols source-packet-routing] hierarchy level.

[See Understanding SR-TE Policy for SRv6 Tunnel for more information.

To configure the maximum segment list depth for SRv6 LSP, you need to enable the maximum-srv6-segment-list-depth configuration statement at the [edit protocols pcep] hierarchy level.

Auto-Bandwidth and PCE-Controlled LSP

Starting in Junos OS Release 14.2R4, support of auto-bandwidth is provided for PCE-controlled LSPs. In earlier releases, the auto-bandwidth option did not apply to PCE-controlled LSPs, although LSPs under the control of auto-bandwdith and constraint-based routing could coexist with PCE-controlled LSPs. The statistics collection for auto-bandwidth was taking effect only when the control mode of a PCE-controlled LSP changes from external to local. This was happening in cases such as no connectivity to a PCE or when a PCE returns delegation of LSPs back to the PCC.

TCP-MD5 Authentication for PCEP Sessions

A stateful PCE server automates the creation of traffic engineering paths across the network, increasing network utilization and enabling a customized programmable networking experience with the use of PCEP communication with a PCC. A PCC sends LSP reports to a PCE server, and the PCE updates or provisions LSPs back to the PCC. The data sent over a PCEP session is crucial for a PCE server to perform external path computing. As a result, an attack on the PCEP communication can disrupt network services. If altered PCEP messages are sent to a PCC, inappropriate LSPs can be set up. Similarly, if altered PCEP messages are sent to a PCE, an incorrect view of the network is learned by the PCE.

Considering the significance of the PCEP communication between a PCE and PCC in executing the PCE functionalities effectively, Junos OS Release 16.1 introduces the feature of securing a PCEP session using TCP-MD5 authentication as per RFC 5440. This feature protects the communication between a PCE and PCC over a PCEP session, which might be subject to an attack, and can disrupt network services.

To enable the MD5 security mechanism for a PCEP session, it is recommended that you define and bind the MD5 authentication key at the [edit protocols pcep pce *pce-id*] hierarchy level for a PCEP session. You can, however, also use a predefined keychain from the [edit security authentication-key-chains *key-chain*] hierarchy level to secure a PCEP session. In this case, you should bind the predefined keychain into the PCEP session at the [edit protocols pcep pce *pce-id*] hierarchy level.

The following configuration is executed on the PCC to establish a secure PCEP session with a PCE:

- Using MD5 authentication key:

```
[edit protocols pcep pce pce-id]
user@PCC# set authentication-key key
```

- Using predefined authentication keychain:

```
[edit protocols pcep pce pce-id]
user@PCC# set authentication-key-chain key-chain
user@PCC# set authentication-algorithm md5
```

For secure PCEP sessions to be established successfully, the MD5 authentication should be configured with the pre-shared authentication key on both the PCE server and the PCC. The PCE and PCC use the same key to verify the authenticity of each segment sent on the TCP connection of the PCEP session.

NOTE:

- Junos OS Release 16.1 supports only TCP-MD5 authentication for PCEP sessions, without extending support for TLS and TCP-AO, such as protection against eavesdropping, tampering, and message forgery.
- Initial application of security mechanism to a PCEP session causes the session to reset.
- If MD5 is misconfigured or not configured on one side of the PCEP session, the session does not get established. Verify that the configurations on the PCC and PCE are matching.
- This feature does not provide support for any session authentication mechanism.
- To view the authentication keychain used by the PCEP session, use the `show path-computation-client status` and `show protocols pcep` command outputs.
- Use the `show system statistics tcp | match auth` command to view the number of packets that get dropped by TCP because of authentication errors.
- Operation of the keychain can be verified by using the `show security keychain detail` command output.

Impact of Client-Side PCE Implementation on Network Performance

The maintenance of a stateful database can be non-trivial. In a single centralized PCE environment, a stateful PCE simply needs to remember all the TE LSPs that the PCE has computed, the TE LSPs that were actually set up (if this can be known), and when the TE LSPs were torn down. However, these requirements cause substantial control protocol overhead in terms of state, network usage and processing, and optimizing links globally across the network. Thus, the concerns of a stateful PCE implementation include:

- Any reliable synchronization mechanism results in significant control plane overhead. PCEs might synchronize state by communicating with each other, but when TE LSPs are set up using distributed computation performed among several PCEs, the problems of synchronization and race condition avoidance become larger and more complex.
- Out-of-band traffic engineering database synchronization can be complex with multiple PCEs set up in a distributed PCE computation model, and can be prone to race conditions, scalability concerns, and so on.
- Path calculations incorporating total network state is highly complex, even if the PCE has detailed information on all paths, priorities, and layers.

In spite of the above concerns, the partial client-side implementation of the stateful PCE is extremely effective in large traffic engineering systems. It provides rapid convergence and significant benefits in terms of optimal resource usage, by providing the requirement for global visibility of a TE LSP state and for ordered control of path reservations across devices within the system being controlled.

Example: Configuring the Path Computation Element Protocol for MPLS RSVP-TE

IN THIS SECTION

- Requirements | **2133**
- Overview | **2133**
- Configuration | **2136**
- Verification | **2143**

This example shows how to enable external path computing by a Path Computation Element (PCE) for traffic engineered label-switched paths (TE LSPs) on a Path Computation Client (PCC). It also shows how to configure the Path Computation Element Protocol (PCEP) on the PCC to enable PCE to PCC communication.

Requirements

This example uses the following hardware and software components:

- Three routers that can be a combination of ACX Series routers, M Series Multiservice Edge Routers, MX Series 5G Universal Routing Platforms, T Series Core Routers, or PTX Series Transport Router, one of which is configured as a PCC.
- A TCP connection to an external stateful PCE from the PCC.
- Junos OS Release 12.3 or later running on the PCC along with the JSDN add-on package.

NOTE: The JSDN add-on package is required to be installed along with the core Junos OS installation package.

Before you begin:

1. Configure the device interfaces.
2. Configure MPLS and RSVP-TE.
3. Configure IS-IS or any other IGP protocol.

Overview

IN THIS SECTION

- Topology | 2135

Starting with Junos OS Release 12.3, the MPLS RSVP-TE functionality is extended to provide a partial client-side implementation of the stateful PCE architecture (draft-ietf-pce-stateful-pce) on a PCC.

NOTE: The partial client-side implementation of the stateful PCE architecture is based on version 2 of Internet draft draft-ietf-pce-stateful-pce. Starting with Junos OS Release 16.1, this implementation is upgraded to support version 7, as defined in Internet draft draft-ietf-pce-stateful-pce-07. Releases prior to 16.1 support the older version of the PCE draft, causing interoperability issues between a PCC running a previous release and a stateful PCE server that adheres to Internet draft draft-ietf-pce-stateful-pce-07.

To enable external path computing by a PCE, include the `lsp-external-controller` statement on the PCC at the `[edit mpls]` and `[edit mpls lsp` *`lsp-name`*`]` hierarchy levels.

```
lsp-external-controller pccd;
```

An LSP configured with the `lsp-external-controller` statement is referred to as a PCE-controlled LSP and is under the external control of a PCE by default. An active stateful PCE can override the parameters set from the CLI, such as bandwidth, path (ERO), and priority, for such PCE-controlled LSPs of the PCC.

To enable PCE to PCC communication, configure PCEP on the PCC at the `[edit protocols]` hierarchy level.

```
pcep { ... }
```

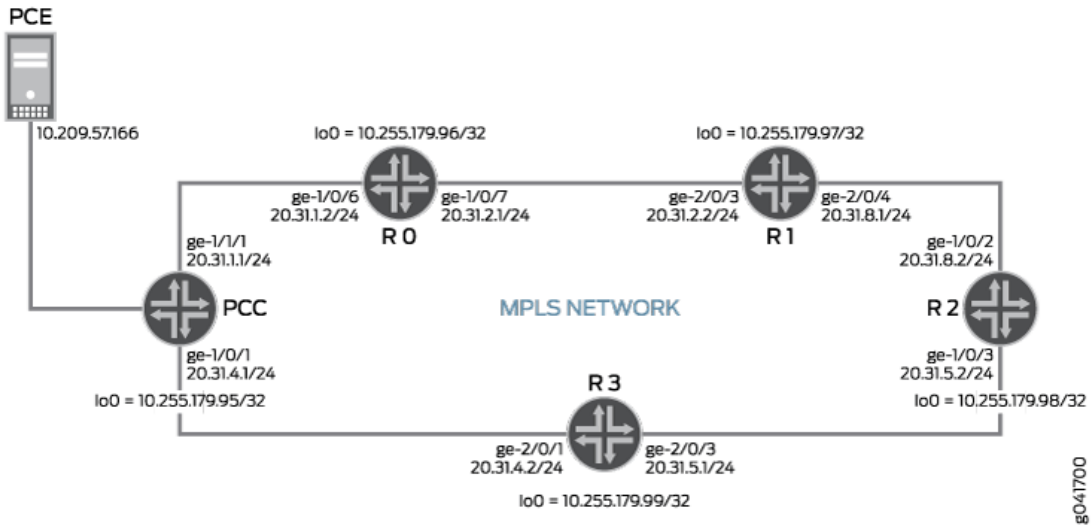
When configuring PCEP on a PCC, be aware of the following considerations:

- The JSDN add-on package is required to be installed along with the core Junos OS installation package.
- Junos OS Release 12.3 supports only stateful PCEs.
- A PCC can connect to a maximum of 10 stateful PCEs. At any given point in time, there can be only one main PCE (the PCE with the lowest priority value, or the PCE that connects to the PCC first in the absence of a PCE priority) to which the PCC delegates LSPs for path computation.
- For Junos OS Release 12.3, the PCC always initiates the PCEP sessions. PCEP sessions initiated by remote PCEs are not accepted by the PCC.
- Existing LSP features, such as LSP protection and make-before-break, work for PCE-controlled LSPs.
- The auto-bandwidth option is turned off for PCE-controlled LSPs, although LSPs under the control of auto-bandwdith and constraint-based routing can coexist with PCE-controlled LSPs.
- PCE-controlled LSPs can be referred to by other CLI configurations, such as lsp-nexthop to routes, forwarding adjacencies, CCC connections, and logical tunnels.
- PCE-controlled LSPs do not support GRES.
- PCE-controlled LSPs under logical-systems are not supported.
- PCE-controlled LSPs cannot be point-to-multipoint LSPs.
- Bidirectional LSPs are not supported.
- PCE-controlled LSPs cannot have secondary paths without a primary path.

- PCE-controlled LSPs depend on external path computation, which impacts overall setup time, reroutes, and make-before-break features.
- Setup time and convergence time (reroute, MBB) for exisiting LSPs is the same as in previous releases, in the absence of PCE-controlled LSPs. However, a small impact is seen in the presence of PCE-controlled LSPs.
- ERO computation time is expected to be significantly higher than local-CSPF.

Topology

Figure 152: Configuring PCEP for MPLS RSVP-TE



In this example, PCC is the ingress router that connects to the external active stateful PCE.

The external LSPs of Router PCC are computed as follows:

1. Router PCC receives the LSP tunnel configuration that was set up using the CLI. Assuming that the received configuration is enabled with external path computing, Router PCC becomes aware that some of the LSP attributes – bandwidth, path, and priority – are under the control of the stateful PCE and delegates the LSP to the PCE.

   In this example, the external LSP is called `PCC-to-R2` and it is being set up from Router PCC to Router R2 . The CLI-configured ERO for `PCC-to-R2` is PCC-R0-R1-R2. The bandwidth for `PCC-to-R2` is 10m, and both the setup and hold priority values are 4.

2. Router PCC tries to retrieve the PCE-controlled LSP attributes. To do this, Router PCC sends out a PCRpt message to the stateful PCE stating that the LSP has been configured. The PCRpt message communicates the status of the LSP and contains the local configuration parameters of the LSP.

3. The stateful PCE modifies one or more of the delegated LSP attributes and sends the new LSP parameters to Router PCC through the PCUpd message.

4. On receiving the new LSP parameters, Router PCC sets up a new LSP and re-signals it using the PCE-provided path.

   In this example, the PCE-provided ERO for `PCC-to-R2` is PCC-R3-R2. The bandwidth for `PCC-to-R2` is 8m, and both the setup and hold priority values are 3.

5. Router PCC sends a PCRpt with the new RRO to the stateful PCE.

Configuration

IN THIS SECTION

- CLI Quick Configuration | **2136**
- Procedure | **2139**
- Results | **2141**

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the `[edit]` hierarchy level.

PCC

```
set interfaces ge-1/0/1 unit 0 family inet address 20.31.4.1/24
set interfaces ge-1/0/1 unit 0 family iso
set interfaces ge-1/0/1 unit 0 family mpls
set interfaces ge-1/1/1 unit 0 family inet address 20.31.1.1/24
set interfaces ge-1/1/1 unit 0 family iso
set interfaces ge-1/1/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.179.95/32
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols mpls lsp-external-controller pccd
set protocols mpls label-switched-path PCC-to-R2 to 10.255.179.98
set protocols mpls label-switched-path PCC-to-R2 bandwidth 10m
set protocols mpls label-switched-path PCC-to-R2 priority 4 4
```

```
set protocols mpls label-switched-path PCC-to-R2 primary to-R2-path
set protocols mpls label-switched-path PCC-to-R2 lsp-external-controller pccd
set protocols mpls path to-R2-path 20.31.1.2 strict
set protocols mpls path to-R2-path 20.31.2.2 strict
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols isis level 1 disable
set protocols isis interface all
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0
set protocols pcep pce pce1 destination-ipv4-address 10.209.57.166
set protocols pcep pce pce1 destination-port 4189
set protocols pcep pce pce1 pce-type active
set protocols pcep pce pce1 pce-type stateful
```

R0

```
set interfaces ge-1/0/6 unit 0 family inet address 20.31.1.2/24
set interfaces ge-1/0/6 unit 0 family iso
set interfaces ge-1/0/6 unit 0 family mpls
set interfaces ge-1/0/7 unit 0 family inet address 20.31.2.1/24
set interfaces ge-1/0/7 unit 0 family iso
set interfaces ge-1/0/7 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.179.96/32
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols isis level 1 disable
set protocols isis interface all
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0
```

R1

```
set system ports console log-out-on-disconnect
set interfaces ge-2/0/3 unit 0 family inet address 20.31.2.2/24
set interfaces ge-2/0/3 unit 0 family iso
set interfaces ge-2/0/3 unit 0 family mpls
set interfaces ge-2/0/4 unit 0 family inet address 20.31.8.1/24
set interfaces ge-2/0/4 unit 0 family iso
```

```
set interfaces ge-2/0/4 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.179.97/32
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols isis level 1 disable
set protocols isis interface all
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0
```

R2

```
set interfaces ge-1/0/2 unit 0 family inet address 20.31.8.2/24
set interfaces ge-1/0/2 unit 0 family iso
set interfaces ge-1/0/2 unit 0 family mpls
set interfaces ge-1/0/3 unit 0 family inet address 20.31.5.2/24
set interfaces ge-1/0/3 unit 0 family iso
set interfaces ge-1/0/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.179.98/32
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols isis level 1 disable
set protocols isis interface all
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0
```

R3

```
set interfaces ge-2/0/1 unit 0 family inet address 20.31.4.2/24
set interfaces ge-2/0/1 unit 0 family iso
set interfaces ge-2/0/1 unit 0 family mpls
set interfaces ge-2/0/3 unit 0 family inet address 20.31.5.1/24
set interfaces ge-2/0/3 unit 0 family iso
set interfaces ge-2/0/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.179.99/32
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols mpls interface all
```

```
set protocols mpls interface fxp0.0 disable
set protocols isis level 1 disable
set protocols isis interface all
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode*.

To configure Router PCC:

NOTE: Repeat this procedure for every Juniper Networks ingress router in the MPLS domain, after modifying the appropriate interface names, addresses, and any other parameters for each router.

1. Configure the interfaces.

   To enable MPLS, include the protocol family on the interface so that the interface does not discard incoming MPLS traffic.

```
[edit interfaces]
user@PCC# set ge-1/0/1 unit 0 family inet address 20.31.4.1/24
user@PCC# set ge-1/0/1 unit 0 family iso
user@PCC# set ge-1/0/1 unit 0 family mpls
user@PCC# set ge-1/1/1 unit 0 family inet address 20.31.1.1/24
user@PCC# set ge-1/1/1 unit 0 family iso
user@PCC# set ge-1/1/1 unit 0 family mpls
user@PCC# set lo0 unit 0 family inet address 10.255.179.95/32
```

2. Enable RSVP on all interfaces of Router PCC, excluding the management interface.

```
[edit protocols]
user@PCC# set rsvp interface all
user@PCC# set rsvp interface fxp0.0 disable
```

3. Configure the label-switched path (LSP) from Router PCC to Router R2 and enable external control of LSPs by the PCE.

```
[edit protocols]
user@PCC# set mpls lsp-external-controller pccd
user@PCC# set mpls label-switched-path PCC-to-R2 to 10.255.179.98/32
user@PCC# set mpls label-switched-path PCC-to-R2 bandwidth 10m
user@PCC# set protocols mpls label-switched-path PCC-to-R2 priority 4 4
user@PCC# set protocols mpls label-switched-path PCC-to-R2 primary to-R2-path
user@PCC# set protocols mpls label-switched-path PCC-to-R2 lsp-external-controller pccd
```

4. Configure the LSP from Router PCC to Router R2, which has local control and is overridden by the PCE-provided LSP parameters.

```
[edit protocols]
user@PCC# set mpls path to-R2-path 20.31.1.2/30 strict
user@PCC# set mpls path to-R2-path 20.31.2.2/30 strict
```

5. Enable MPLS on all interfaces of Router PCC, excluding the management interface.

```
[edit protocols]
user@PCC# set mpls interface all
user@PCC# set mpls interface fxp0.0 disable
```

6. Configure IS-IS on all interfaces of Router PCC, excluding the management interface.

```
[edit protocols]
user@PCC# set isis level 1 disable
user@PCC# set isis interface all
user@PCC# set isis interface fxp0.0 disable
user@PCC# set isis interface lo0.0
```

7. Define the PCE that Router PCC connects to, and configure the IP address of the PCE.

```
[edit protocols]
user@PCC# set pcep pce pce1 destination-ipv4-address 10.209.57.166
```

8. Configure the destination port for Router PCC that connects to a PCE using the TCP-based PCEP.

```
[edit protocols]
user@PCC# set pcep pce pce1 destination-port 4189
```

9. Configure the PCE type.

```
[edit protocols]
user@PCC# set pcep pce pce1 pce-type active
user@PCC# set pcep pce pce1 pce-type stateful
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@PCC# show interfaces
ge-1/0/1 {
    unit 0 {
        family inet {
            address 20.31.4.1/24;
        }
        family iso;
        family mpls;
    }
}
ge-1/1/1 {
    unit 0 {
        family inet {
            address 20.31.1.1/24;
        }
        family iso;
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.255.179.95/32;
```

```
        }
    }
}
```

```
user@PCC# show protocols
rsvp {
    interface all;
    interface fxp0.0 {
        disable;
    }
}
mpls {
    lsp-external-controller pccd;
    label-switched-path PCC-to-R2 {
        to 10.255.179.98;
        bandwidth 10m;
        priority 4 4;
        primary to-R2-path;
        lsp-external-controller pccd;
    }
    path to-R2-path {
        20.31.1.2 strict;
        20.31.2.2 strict;
    }
    interface all;
    interface fxp0.0 {
        disable;
    }
}
isis {
    level 1 disable;
    interface all;
    interface fxp0.0 {
        disable;
    }
    interface lo0.0;
}
pcep {
    pce pce1 {
        destination-ipv4-address 10.209.57.166;
        destination-port 4189;
```

```
        pce-type active stateful;
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

IN THIS SECTION

- Verifying the PCEP Session Status | **2143**
- Verifying the PCE-Controlled LSP Status When LSP Control Is External | **2144**
- Verifying the PCE-Controlled LSP Status When LSP Control Is Local | **2146**

Confirm that the configuration is working properly.

Verifying the PCEP Session Status

Purpose

Verify the PCEP session status between the PCE and Router PCC when the PCE status is up.

Action

From operational mode, run the **show path-computation-client active-pce** command.

```
user@PCC> show path-computation-client active-pce
PCE pce1
General
    IP address              : 10.209.57.166
    Priority                : 0
    PCE status              : PCE_STATE_UP
    Session type            : PCE_TYPE_STATEFULACTIVE
    PCE-mastership          : main

Counters
    PCReqs              Total: 0           last 5min: 0           last hour: 0
```

```
    PCReps              Total: 0           last 5min: 0           last hour: 0
    PCRpts              Total: 5           last 5min: 5           last hour: 5
    PCUpdates           Total: 1           last 5min: 1           last hour: 1

Timers
    Local            Keepalive timer:       30 [s]   Dead timer:       120 [s]
    Remote           Keepalive timer:       30 [s]   Dead timer:       120 [s]

Errors
    PCErr-recv
    PCErr-sent
    PCE-PCC-NTFS
    PCC-PCE-NTFS
```

Meaning

The output displays information about the current active stateful PCE that Router PCC is connected to. The **PCE status** output field indicates the current status of the PCEP session between the PCE and Router PCC.

For **pce1**, the status of the PCEP session is **PCE_STATE_UP**, which indicates that the PCEP session has been established between the PCEP peers.

The statistics of **PCRpts** indicate the number of messages sent by Router PCC to the PCE to report the current status of LSPs. The **PCUpdates** statistics indicate the number of messages received by Router PCC from the PCE. The **PCUpdates** messages include the PCE modified parameters for the PCE-controlled LSPs.

Verifying the PCE-Controlled LSP Status When LSP Control Is External

Purpose

Verify the status of the PCE-controlled LSP from Router PCC to Router R2 when the LSP is under external control.

Action

From operational mode, run the **show mpls lsp name PCC-to-R2 extensive** command.

```
user@PCC> show mpls lsp name PCC-to-R2 extensive
Ingress LSP: 1 sessions
```

```
10.255.179.98
  From: 10.255.183.59, State: Up, ActiveRoute: 0, LSPname: PCC-to-R2
  ActivePath: to-R2-path (primary)
  LSPtype: Externally controlled, Penultimate hop popping
  LSP Control Status: Externally controlled
    LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary   to-R2-path     State: Up
    Priorities: 3 3
    Bandwidth: 8Mbps
    SmartOptimizeTimer: 180
        No computed ERO.
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
          20.31.4.2 20.31.5.2
   21 Mar 11 05:00:56.736 EXTCTRL LSP: Sent Path computation request and LSP status
   20 Mar 11 05:00:56.736 EXTCTRL_LSP: Computation request/lsp status contains:  bandwidth
10000000 priority - setup 4 hold 4 hops:  20.31.1.2 20.31.2.2
   19 Mar 11 05:00:56.735 Selected as active path
   18 Mar 11 05:00:56.734 EXTCTRL LSP: Sent Path computation request and LSP status
   17 Mar 11 05:00:56.734 EXTCTRL_LSP: Computation request/lsp status contains:  bandwidth
10000000 priority - setup 4 hold 4 hops:  20.31.1.2 20.31.2.2
   16 Mar 11 05:00:56.734 Record Route:  20.31.4.2 20.31.5.2
   15 Mar 11 05:00:56.734 Up
   14 Mar 11 05:00:56.713 EXTCTRL LSP: Sent Path computation request and LSP status
   13 Mar 11 05:00:56.713 EXTCTRL_LSP: Computation request/lsp status contains:  bandwidth
10000000 priority - setup 4 hold 4 hops:  20.31.1.2 20.31.2.2
   12 Mar 11 05:00:56.712 Originate Call
   11 Mar 11 05:00:56.712 EXTCTRL_LSP: Received setup parameters :  20.31.4.2 20.31.5.2
   10 Mar 11 05:00:49.283 EXTCTRL LSP: Sent Path computation request and LSP status
    9 Mar 11 05:00:49.283 EXTCTRL_LSP: Computation request/lsp status contains:  bandwidth
10000000 priority - setup 4 hold 4 hops:  20.31.1.2 20.31.2.2
    8 Mar 11 05:00:20.581 EXTCTRL_LSP: Computation request/lsp status contains:  bandwidth
10000000 priority - setup 4 hold 4 hops:  20.31.1.2 20.31.2.2
    7 Mar 11 05:00:20.581 EXTCTRL LSP: Sent Path computation request and LSP status
    6 Mar 11 05:00:20.581 EXTCTRL_LSP: Computation request/lsp status contains:  bandwidth
10000000 priority - setup 4 hold 4 hops:  20.31.1.2 20.31.2.2
    5 Mar 11 05:00:20.580 EXTCTRL_LSP: Control status became external
    4 Mar 11 05:00:03.716 EXTCTRL_LSP: Control status became local
    3 Mar 11 05:00:03.714 EXTCTRL LSP: Sent Path computation request and LSP status
    2 Mar 11 05:00:03.714 EXTCTRL_LSP: Computation request/lsp status contains:  bandwidth
10000000 priority - setup 4 hold 4 hops:  20.31.1.2 20.31.2.2
    1 Mar 11 05:00:00.279 EXTCTRL LSP: Awaiting external controller connection
  Created: Mon Mar 11 05:00:00 2013
```

```
Total 1 displayed, Up 1, Down 0

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Meaning

In the output, the **LSPtype** and **LSP Control Status** output fields show that the LSP is externally controlled. The output also shows a log of the PCEP messages sent between Router PCC and the PCE.

The PCEP session between the PCE and Router PCC is up, and Router PCC receives the following PCE-controlled LSP parameters:

- ERO (path)—20.31.4.2 and 20.31.5.2
- Bandwidth—8Mbps
- Priorities—3 3 (setup and hold values)

Verifying the PCE-Controlled LSP Status When LSP Control Is Local

Purpose

Verify the status of the PCE-controlled LSP from Router PCC to Router R2 when the LSP control becomes local.

Action

From operational mode, run the **show mpls lsp name PCC-to-R2 extensive** command.

```
user@PCC> show mpls lsp name PCC-to-R2 extensive
Ingress LSP: 1 sessions

10.255.179.98
  From: 10.255.183.59, State: Up, ActiveRoute: 0, LSPname: PCC-to-R2
  ActivePath: to-R2-path (primary)
  LSPtype: Externally controlled, Penultimate hop popping
  LSP Control Status: Locally controlled
  LoadBalance: Random
```

```
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary   to-R2-path     State: Up
    Priorities: 4 4 (ActualPriorities 3 3)
    Bandwidth: 10Mbps (ActualBandwidth: 8Mbps)
        SmartOptimizeTimer: 180
        No computed ERO.
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
          20.31.4.2 20.31.5.2
   22 Mar 11 05:02:09.618 EXTCTRL_LSP: Control status became local
   21 Mar 11 05:00:56.736 EXTCTRL LSP: Sent Path computation request and LSP status
   20 Mar 11 05:00:56.736 EXTCTRL_LSP: Computation request/lsp status contains:  bandwidth
10000000 priority - setup 4 hold 4 hops:  20.31.1.2 20.31.2.2
   19 Mar 11 05:00:56.735 Selected as active path
   18 Mar 11 05:00:56.734 EXTCTRL LSP: Sent Path computation request and LSP status
   17 Mar 11 05:00:56.734 EXTCTRL_LSP: Computation request/lsp status contains:  bandwidth
10000000 priority - setup 4 hold 4 hops:  20.31.1.2 20.31.2.2
   16 Mar 11 05:00:56.734 Record Route:  20.31.4.2 20.31.5.2
   15 Mar 11 05:00:56.734 Up
   14 Mar 11 05:00:56.713 EXTCTRL LSP: Sent Path computation request and LSP status
   13 Mar 11 05:00:56.713 EXTCTRL_LSP: Computation request/lsp status contains:  bandwidth
10000000 priority - setup 4 hold 4 hops:  20.31.1.2 20.31.2.2
   12 Mar 11 05:00:56.712 Originate Call
   11 Mar 11 05:00:56.712 EXTCTRL_LSP: Received setup parameters :  20.31.4.2 20.31.5.2
   10 Mar 11 05:00:49.283 EXTCTRL LSP: Sent Path computation request and LSP status
    9 Mar 11 05:00:49.283 EXTCTRL_LSP: Computation request/lsp status contains:  bandwidth
10000000 priority - setup 4 hold 4 hops:  20.31.1.2 20.31.2.2
    8 Mar 11 05:00:20.581 EXTCTRL_LSP: Computation request/lsp status contains:  bandwidth
10000000 priority - setup 4 hold 4 hops:  20.31.1.2 20.31.2.2
    7 Mar 11 05:00:20.581 EXTCTRL LSP: Sent Path computation request and LSP status
    6 Mar 11 05:00:20.581 EXTCTRL_LSP: Computation request/lsp status contains:  bandwidth
10000000 priority - setup 4 hold 4 hops:  20.31.1.2 20.31.2.2
    5 Mar 11 05:00:20.580 EXTCTRL_LSP: Control status became external
    4 Mar 11 05:00:03.716 EXTCTRL_LSP: Control status became local
    3 Mar 11 05:00:03.714 EXTCTRL LSP: Sent Path computation request and LSP status
    2 Mar 11 05:00:03.714 EXTCTRL_LSP: Computation request/lsp status contains:  bandwidth
10000000 priority - setup 4 hold 4 hops:  20.31.1.2 20.31.2.2
    1 Mar 11 05:00:00.279 EXTCTRL LSP: Awaiting external controller connection
  Created: Mon Mar 11 05:00:00 2013
Total 1 displayed, Up 1, Down 0

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

```
Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Meaning

In the output, the **LSP Control Status** output field shows that the LSP is under local control. Although the PCE-controlled LSP is under local control, Router PCC continues to use the PCE-provided parameters, until the next opportunity to re-signal the LSP.

The output now displays the LSP parameters that were configured using the CLI along with the PCE-provided parameters used to establish the LSP as the actual values in use.

- Bandwidth—10Mbps (ActualBandwidth: 8Mbps)
- Priorities—4 4 (ActualPriorities 3 3)

On the trigger to re-signal the LSP, Router PCC uses the local configuration parameters to establish the PCE-controlled LSP.

```
user@PCC> show mpls lsp name PCC-to-R2 extensive externally-controlled
Ingress LSP: 1 sessions

10.255.179.98
  From: 10.255.183.59, State: Up, ActiveRoute: 0, LSPname: PCC-to-R2
  ActivePath: to-R2-path (primary)
  LSPtype: Externally controlled, Penultimate hop popping
  LSP Control Status: Locally controlled
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary   to-R2-path     State: Up
  Priorities: 4 4
  Bandwidth: 10Mbps
    SmartOptimizeTimer: 180
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 30)
 20.31.1.2 S 20.31.2.2 S 20.31.8.2 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
          20.31.1.2 20.31.2.2 20.31.8.2
   28 Mar 11 05:02:51.787 Record Route:  20.31.1.2 20.31.2.2 20.31.8.2
   27 Mar 11 05:02:51.787 Up
   26 Mar 11 05:02:51.697 EXTCTRL_LSP: Applying local parameters with this signalling attempt
   25 Mar 11 05:02:51.697 Originate Call
   24 Mar 11 05:02:51.696 Clear Call
   23 Mar 11 05:02:51.696 CSPF: computation result accepted  20.31.1.2 20.31.2.2 20.31.8.2
```

```
   22 Mar 11 05:02:09.618 EXTCTRL_LSP: Control status became local
   21 Mar 11 05:00:56.736 EXTCTRL LSP: Sent Path computation request and LSP status
   20 Mar 11 05:00:56.736 EXTCTRL_LSP: Computation request/lsp status contains:  bandwidth
10000000 priority - setup 4 hold 4 hops:  20.31.1.2 20.31.2.2
   19 Mar 11 05:00:56.735 Selected as active path
   18 Mar 11 05:00:56.734 EXTCTRL LSP: Sent Path computation request and LSP status
   17 Mar 11 05:00:56.734 EXTCTRL_LSP: Computation request/lsp status contains:  bandwidth
10000000 priority - setup 4 hold 4 hops:  20.31.1.2 20.31.2.2
   16 Mar 11 05:00:56.734 Record Route:  20.31.4.2 20.31.5.2
   15 Mar 11 05:00:56.734 Up
   14 Mar 11 05:00:56.713 EXTCTRL LSP: Sent Path computation request and LSP status
   13 Mar 11 05:00:56.713 EXTCTRL_LSP: Computation request/lsp status contains:  bandwidth
10000000 priority - setup 4 hold 4 hops:  20.31.1.2 20.31.2.2
   12 Mar 11 05:00:56.712 Originate Call
   11 Mar 11 05:00:56.712 EXTCTRL_LSP: Received setup parameters :  20.31.4.2 20.31.5.2
   10 Mar 11 05:00:49.283 EXTCTRL LSP: Sent Path computation request and LSP status
    9 Mar 11 05:00:49.283 EXTCTRL_LSP: Computation request/lsp status contains:  bandwidth
10000000 priority - setup 4 hold 4 hops:  20.31.1.2 20.31.2.2
    8 Mar 11 05:00:20.581 EXTCTRL_LSP: Computation request/lsp status contains:  bandwidth
10000000 priority - setup 4 hold 4 hops:  20.31.1.2 20.31.2.2
    7 Mar 11 05:00:20.581 EXTCTRL LSP: Sent Path computation request and LSP status
    6 Mar 11 05:00:20.581 EXTCTRL_LSP: Computation request/lsp status contains:  bandwidth
10000000 priority - setup 4 hold 4 hops:  20.31.1.2 20.31.2.2
    5 Mar 11 05:00:20.580 EXTCTRL_LSP: Control status became external
    4 Mar 11 05:00:03.716 EXTCTRL_LSP: Control status became local
    3 Mar 11 05:00:03.714 EXTCTRL LSP: Sent Path computation request and LSP status
    2 Mar 11 05:00:03.714 EXTCTRL_LSP: Computation request/lsp status contains:  bandwidth
10000000 priority - setup 4 hold 4 hops:  20.31.1.2 20.31.2.2
    1 Mar 11 05:00:00.279 EXTCTRL LSP: Awaiting external controller connection
  Created: Mon Mar 11 05:00:00 2013
Total 1 displayed, Up 1, Down 0

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

The **Computed ERO** is 20.31.1.2, 20.31.2.2, and 20.31.8.2. The PCE-controlled LSP is established using the local configuration parameters.

Example: Configuring Path Computation Element Protocol for MPLS RSVP-TE with Support of PCE-Initiated Point-to-Point LSPs

IN THIS SECTION

- Requirements | 2150
- Overview | 2150
- Configuration | 2153
- Verification | 2158

This example shows how to configure the Path Computation Client (PCC) with the capability of supporting Path Computation Element (PCE)-initiated traffic-engineered point-to-point label-switched paths (LSPs).

Requirements

This example uses the following hardware and software components:

- Three routers that can be a combination of ACX Series, M Series, MX Series, or T Series routers.
- A TCP connection to two external stateful PCEs from the ingress router (PCC).
- Junos OS Release 16.1 or later running on the PCC.

Before you begin:

- Configure the device interfaces.
- Configure MPLS and RSVP-TE (RSVP-Traffic Engineering).
- Configure OSPF or any other IGP protocol.

Overview

IN THIS SECTION

- Topology | 2152

Starting with Junos OS Release 16.1, the PCEP functionality is extended to allow a stateful PCE to initiate and provision traffic engineering LSPs through a PCC. Earlier, the LSPs were configured on the PCC and the PCC delegated control over the external LSPs to a PCE. The ownership of the LSP state was maintained by the PCC. With the introduction of the PCE-initiated LSPs, a PCE can initiate and provision a traffic engineering point-to-point LSP dynamically without the need for a locally configured LSP on the PCC. On receiving a PCCreate message from a PCE, the PCC creates the PCE-initiated LSP and automatically delegates the LSP to the PCE.

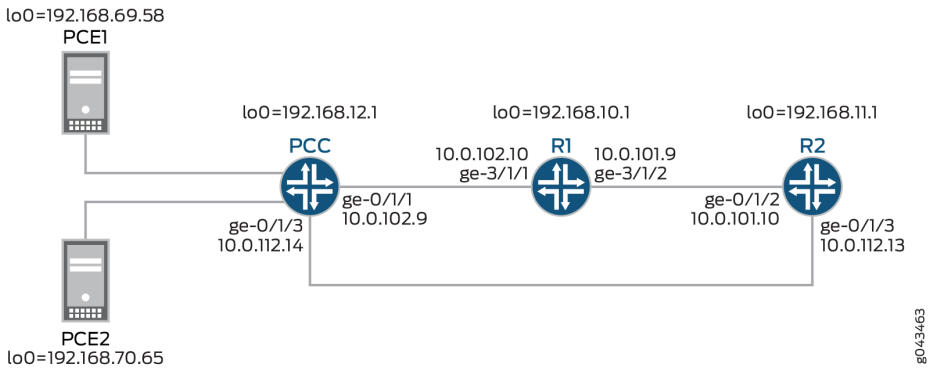
When configuring the support of PCE-initiated point-to-point LSPs for a PCC, be aware of the following considerations:

- Junos OS Release 13.3 supports only stateful PCEs.
- For Junos OS Release 13.3, the PCC always initiates the PCEP sessions. PCEP sessions initiated by remote PCEs are not accepted by the PCC.
- Existing LSP features, such as LSP protection and make-before-break, work for PCE-initiated LSPs.
- PCE-initiated LSPs do not support graceful Routing Engine switchover (GRES).
- PCE-initiated LSPs under logical systems are not supported.
- PCE-initiated LSPs cannot be point-to-multipoint LSPs.
- Bidirectional LSPs are not supported.
- RSVP-TE for unnumbered links is not supported. PCE-initiated LSPs support only numbered links.
- The PCE initiating a segment routing LSP can use the binding segment ID (SID) labels associated with non-colored segment routing LSPs to provision the PCE-initiated segment routing LSP paths.

  Starting in Junos OS Release 18.2R1, statically configured non-colored segment routing LSPs on the ingress device are reported to a PCE through a PCEP session. These non-colored segment routing LSPs may have binding SID labels associated with them. With this feature, the PCE can use this binding SID label in the label stack to provision PCE-initiated segment routing LSP paths.

Topology

Figure 153: Example PCE-Initated Point-to-Point LSP for MPLS RSVP-TE



In this example, PCC is the ingress router that connects to two external stateful PCEs: PCE1 and PCE2.

When there is a new demand, the active stateful PCE dynamically initiates an LSP to meet the requirement. Since PCC is configured with the capability of supporting the PCE-initiated LSP, path computation on PCC is performed as follows:

1. A PCE sends a PCCreate message to the PCC to initiate and provision an LSP. The PCC sets up the PCE-initiated LSP using the parameters received from the PCE, and automatically delegates the PCE-initiated LSP to the PCE that initiated it.

   In this example, PCE1 is the active stateful PCE that initiates and provisions the PCE-initiated LSP on PCC. On receiving the PCE-initiated LSP parameters, PCC sets up the LSP and automatically delegates the PCE-initiated LSP to PCE1.

2. When the PCEP session between PCC and PCE1 is terminated, PCC starts two timers for the PCE1-initiated LSP: delgation cleanup timeout and the LSP cleanup timer. During this time, PCE1 or PCE2 can acquire control of the PCE-initiated LSP.

3. If PCE2 acquires control over the PCE-initiated LSP before the expiration of the LSP cleanup timer, PCC delegates the PCE-initiated LSP to PCE2, and the LSP cleanup timer and the delegation cleanup timeout are stopped.

4. If the delegation cleanup timeout expired, and neither PCE1 nor PCE2 acquired control over the PCE-initiated LSP, PCC takes local control of the non-delegated PCE-initiated LSP until the expiration of the LSP cleanup timer.

5. After the expiration of the LSP cleanup timer, PCC deletes the PCE-initiated LSP provisioned by PCE1.

Configuration

IN THIS SECTION

- CLI Quick Configuration | 2153
- Procedure | 2155
- Results | 2157

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the `[edit]` hierarchy level.

PCC

```
set interfaces ge-0/1/1 unit 0 family inet address 10.0.102.9/24
set interfaces ge-0/1/1 unit 0 family iso
set interfaces ge-0/1/1 unit 0 family mpls
set interfaces ge-0/1/3 unit 0 family inet address 10.0.112.14/24
set interfaces ge-0/1/3 unit 0 family iso
set interfaces ge-0/1/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.12.1/32
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols mpls lsp-external-controller ppcd
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols pcep pce-group PCEGROUP pce-type active
set protocols pcep pce-group PCEGROUP pce-type stateful
set protocols pcep pce-group PCEGROUP lsp-provisioning
set protocols pcep pce-group PCEGROUP lsp-cleanup-timer 30
set protocols pcep pce PCE1 destination-ipv4-address 192.168.69.58
set protocols pcep pce PCE1 destination-port 4189
set protocols pcep pce PCE1 pce-group PCEGROUP
set protocols pcep pce PCE2 destination-ipv4-address 192.168.70.65
```

```
set protocols pcep pce PCE2 destination-port 4189
set protocols pcep pce PCE2 pce-group PCEGROUP
```

R1

```
set interfaces ge-3/1/1 unit 0 family inet address 10.0.102.10/24
set interfaces ge-3/1/1 unit 0 family iso
set interfaces ge-3/1/1 unit 0 family mpls
set interfaces ge-3/1/2 unit 0 family inet address 10.0.101.9/24
set interfaces ge-3/1/2 unit 0 family iso
set interfaces ge-3/1/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.10.1/32
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
```

R2

```
set interfaces ge-0/1/1 unit 0 family inet address 10.0.101.10/24
set interfaces ge-0/1/1 unit 0 family iso
set interfaces ge-0/1/1 unit 0 family mpls
set interfaces ge-0/1/3 unit 0 family inet address 10.0.112.13/24
set interfaces ge-0/1/3 unit 0 family iso
set interfaces ge-0/1/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.11.1/32
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode*.

To configure the PCC router:

NOTE: Repeat this procedure for every Juniper Networks ingress router in the MPLS domain, after modifying the appropriate interface names, addresses, and any other parameters for each router.

1. Configure the interfaces.

   To enable MPLS, include the protocol family on the interface so that the interface does not discard incoming MPLS traffic.

```
[edit interfaces]
user@PCC# set ge-0/1/1 unit 0 family inet address 10.0.102.9/24
user@PCC# set ge-0/1/1 unit 0 family iso
user@PCC# set ge-0/1/1 unit 0 family mpls
user@PCC# set ge-0/1/3 unit 0 family inet address 10.0.112.14/24
user@PCC# set ge-0/1/3 unit 0 family iso
user@PCC# set ge-0/1/3 unit 0 family mpls
user@PCC# set lo0 unit 0 family inet address 192.168.12.1/32
```

2. Enable RSVP on all interfaces of the PCC, excluding the management interface.

```
[edit protocols]
user@PCC# set rsvp interface all
user@PCC# set rsvp interface fxp0.0 disable
```

3. Enable external control of LSPs by the PCEs.

```
[edit protocols]
user@PCC# set mpls lsp-external-controller pccd
```

4. Enable MPLS on all interfaces of the PCC, excluding the management interface.

```
[edit protocols]
user@PCC# set mpls interface all
user@PCC# set mpls interface fxp0.0 disable
```

5. Configure OSPF on all interfaces of the PCC, excluding the management interface.

```
[edit protocols]
user@PCC# set ospf traffic-engineering
user@PCC# set ospf area 0.0.0.0 interface all
user@PCC# set ospf area 0.0.0.0 interface fxp0.0 disable
user@PCC# set ospf interface lo0.0
```

6. Define the PCE group and enable support of PCE-initiated LSPs for the PCE group.

```
[edit protocols]
user@PCC# set protocols pcep pce-group PCEGROUP pce-type active
user@PCC# set protocols pcep pce-group PCEGROUP pce-type stateful
user@PCC# set protocols pcep pce-group PCEGROUP lsp-provisioning
user@PCC# set protocols pcep pce-group PCEGROUP lsp-cleanup-timer 30
```

7. Define the PCEs that connect to the PCC.

```
[edit protocols]
user@PCC# set pcep pce PCE1 destination-ipv4-address 192.168.69.58
user@PCC# set pcep pce PCE1 destination-port 4189
user@PCC# set pcep pce PCE1 pce-group PCEGROUP
user@PCC# set pcep pce PCE2 destination-ipv4-address 192.168.70.65
user@PCC# set pcep pce PCE2 destination-port 4189
user@PCC# set pcep pce PCE2 pce-group PCEGROUP
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@PCC# show interfaces
ge-0/1/1 {
    unit 0 {
        family inet {
            address 10.0.102.9/24;
        }
        family iso;
        family mpls;
    }
}
ge-0/1/3 {
    unit 0 {
        family inet {
            address 10.0.112.14/24;
        }
        family iso;
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 192.168.12.1/32;
        }
    }
}
```

```
user@PCC# show protocols
rsvp {
    interface all;
    }
    interface fxp0.0 {
        disable;
    }
}
```

```
mpls {
    lsp-external-controller pccd;
    interface all;
    interface fxp0.0 {
        disable;
    }
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
}
pce-group PCEGROUP {
    pce-type active stateful;
    lsp-provisioning;
    lsp-cleanup-timer 30;
}
pce PCE1 {
    destination-ipv4-address 192.168.69.58;
    destination-port 4189;
    pce-group PCEGROUP;
}
pce PCE2 {
    destination-ipv4-address 192.168.70.65;
    destination-port 4189;
    pce-group PCEGROUP;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying PCC Status | **2159**
- Verifying PCE1 Status | **2160**

- Verifying the PCE-Initiated LSP Status When the LSP Is Externally Provisioned | **2161**

Confirm that the configuration is working properly.

Verifying PCC Status

Purpose

Verify the PCEP session status and LSP summary between the PCC and the connected PCEs.

Action

From operational mode, run the `show path-computation-client status` command.

```
user@PCC# show path-computation-client status
Session              Type                       Provisioning   Status
  PCE1               Stateful Active            On             Up
  PCE2               Stateful Active            On             Up

LSP Summary
  Total number of LSPs        : 1
  Static LSPs                 : 0
  Externally controlled LSPs  : 0
  Externally provisioned LSPs : 1/16000 (current/limit)
  Orphaned LSPs               : 0

PCE1 (main)
  Delegated              : 1
  Externally provisioned : 1
PCE2
  Delegated              : 0
  Externally provisioned : 0
```

Meaning

The output displays the status of the PCEP session between the active stateful PCEs and the PCC. It also displays information about the different types of LSPs on the PCC, and the number of LSPs provisioned by the connected PCEs and delegated to them.

PCE1 is the main active PCE and has one PCE-initiated LSP that has been automatically delegated to it by the PCC.

Verifying PCE1 Status

Purpose

Verify the status of the main active stateful PCE.

Action

From operational mode, run the `show path-computation-client active-pce detail` command.

```
user@PCC# show path-computation-client active-pce
PCE PCE1
-----------------------------------------
General
    IP address               : 192.168.69.58
    Priority                 : 0
    PCE status               : PCE_STATE_UP
    Session type             : PCE_TYPE_STATEFULACTIVE
    LSP provisioning allowed : On
    LSP cleanup timer        : 30 [s]
    PCE-mastership           : main
    Max unknown messages     : 5
    Keepalives received      : 0
    Keepalives sent          : 0
    Dead timer               : 0 [s]
    Elapsed as main current  : 1 [s]
    Elapsed as main total    : 446380 [s]
    Unknown msgs/min rate    : 0
    Session failures         : 2198
    Corrupted messages       : 0
    Delegation timeout set   : 30
    Delegation timeout in    : 0 [s]
    Delegation failures      : 0
```

```
    Connection down          : 167092 [s]

Counters
    PCReqs              Total: 0           last 5min: 0           last hour: 0
    PCReps              Total: 0           last 5min: 0           last hour: 0
    PCRpts              Total: 5           last 5min: 5           last hour: 5
    PCUpdates           Total: 0           last 5min: 0           last hour: 0
    PCCreates           Total: 1           last 5min: 1           last hour: 1

Timers
    Local  Keepalive timer:   30 [s]  Dead timer:  120 [s]  LSP cleanup timer:   30 [s]
    Remote Keepalive timer:    0 [s]  Dead timer:    0 [s]  LSP cleanup timer:    - [s]

Errors
    PCErr-recv
    PCErr-sent
    PCE-PCC-NTFS
    PCC-PCE-NTFS
```

Meaning

The output displays information about the current active stateful PCE to which the PCC is connected. The `PCE status` output field indicates the current status of the PCEP session between a PCE and PCC.

For PCE1, the status of the PCEP session is `PCE_STATE_UP`, which indicates that the PCEP session has been established with the PCC.

Verifying the PCE-Initiated LSP Status When the LSP Is Externally Provisioned

Purpose

Verify the status of the PCE-initiated LSP.

Action

From operational mode, run the `show mpls lsp externally-provisioned detail` command.

```
user@PCC# show mpls lsp externally-provisioned detail
Ingress LSP: 1 sessions

10.0.101.10
```

```
  From: 10.0.102.9, State: Up, ActiveRoute: 0, LSPname: lsp15
  ActivePath: path1 (primary)
  Link protection desired
  LSPtype: Externally Provisioned, Penultimate hop popping
  LSP Control Status: Externally controlled
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary   path1            State: Up
    Priorities: 7 0
    Bandwidth: 8Mbps
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 3)
 10.0.102.10 S 10.0.101.9 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node
10=SoftPreempt 20=Node-ID):
 10.0.102.10 S 10.0.101.9 S
```

Meaning

In the output, the `LSPtype` output field shows that the LSP is externally provisioned.

The PCEP session between PCC and PCE1 is up, and the PCC receives the following PCE-initiated LSP parameters:

- ERO (path)—10.0.102.10 and 10.0.101.9
- Bandwidth—8 Mbps
- Priority—7 0 (setup and hold values)

Configuring Path Computation Element Protocol for MPLS RSVP-TE with Support of PCE-Initiated Point-to-Point LSPs

You can configure a Path Computation Client (PCC) with the capability of supporting dynamically created label switched paths (LSPs) from a centralized external path computing entity. A stateful Path Computaiton Element (PCE) can be used to perform external path computation and generate dynamic LSPs when there is an increase in demand.

A PCC creates the PCE-initiated point-to-point LSP using the PCE-provided LSP parameters, or parameters from a pre-configured LSP template when the PCE does not provision the LSP, and automatically delegates the PCE-initiated point-to-point LSP to the respective PCE. As a result, for PCE-initiated LSPs, there is no need for a locally configured LSP on the PCC.

A CLI-controlled LSP, PCE-controlled LSP, and PCE-initiated LSP can coexist with each other on a PCC.

Before you begin:

- Configure the device interfaces.
- Configure MPLS and RSVP-TE.
- Configure OSPF or any other IGP protocol.

To configure PCC to support PCE-initiated point-to-point LSPs, complete the following tasks:

1. In configuration mode, go to the following hierarchy level:

```
[edit]
user@PCC# edit protocols pcep
```

2. Specify the number of messages per minute that the PCC can receive at maximum.

```
[edit protocols pcep]
user@PCC# set message-rate-limit messages-per-minute
```

3. Specify the number of externally provisioned label switched paths (LSPs) over all connected PCEs that the PCC can accept at maximum.

```
[edit protocols pcep]
user@PCC# set max-provisioned-lsps max-count
```

4. Specify the unique user defined ID for the connected PCE to configure the PCE parameters.

```
[edit protocols pcep]
user@PCC# edit pce pce-id
```

5. Specify the amount of time (in seconds) that the PCC must wait before returning control of LSPs to the routing protocol process after a PCEP session is disconnected.

```
[edit protocols pcep pce pce-id]
user@PCC# set delegation-cleanup-timeout seconds
```

6. Specify the IPv4 address of the PCE to connect with.

```
[edit protocols pcep pce pce-id]
user@PCC# set destination-ipv4-address ipv4-address
```

7. Specify the TCP port number that the PCE is using

```
[edit protocols pcep pce pce-id]
user@PCC# set destination-port port-number
```

The value can range from 1 through 65535 and the default value is 4189.

8. Specify the amount of time (in seconds) that the PCC must wait before deleting any non-delegated PCE-initiated LSPs from the failed PCE after a PCEP session terminates.

```
[edit protocols pcep pce pce-id]
user@PCC# set lsp-cleanup-timer seconds
```

9. Configure the PCC to accept SPs that are externally provisioned by connected PCEs. By default, the PCC rejects PCE-initiated LSPs.

```
[edit protocols pcep pce pce-id]
user@PCC# set lsp-provisioning
```

10. Specify the number of unknown messages per minute that the PCC can receive at maximum after which the PCEP session is closed.

```
[edit protocols pcep pce pce-id]
user@PCC# set max-unknown-messages messages-per-minute
```

The value can range from 1 through 16384, and the default value is 0 (disabled or no limit).

11. Specify the number of unknown requests per minute that the PCC can receive at maximum after which the PCEP session is terminated.

```
[edit protocols pcep pce pce-id]
user@PCC# set max-unknown-requests requests-per-minute
```

The value can range from 0 through 16384, and the default value is 5. A value of 0 disables this statement.

12. Configure the PCE type.

```
[edit protocols pcep pce pce-id]
user@PCC# set pce-type active stateful
```

13. Specify the amount of time (in seconds) that the PCC must wait for a reply before resending a request.

```
[edit protocols pcep pce pce-id]
user@PCC# set request-timer seconds
```

The value can range from 0 through 65535 seconds.

14. Verify and commit the configuration.

```
user@PCC# show
user@PCC# commit
```

Sample Output

```
[edit]
user@PCC# edit protocols pcep

[edit protocols pcep]
user@PCC# set message-rate-limit 50

[edit protocols pcep]
user@PCC# set max-provisioned-lsps 16000

[edit protocols pcep]
user@PCC# edit pce PCE

[edit protocols pcep pce PCE]
user@PCC# set delegation-cleanup-timeout 20

[edit protocols pcep pce PCE]
user@PCC# set destination-ipv4-address 192.168.69.58

[edit protocols pcep pce PCE]
user@PCC# set destination-port 4189

[edit protocols pcep pce PCE]
user@PCC# set lsp-cleanup-timer 50

[edit protocols pcep pce PCE]
user@PCC# set lsp-provisioning
```

```
[edit protocols pcep pce PCE]
user@PCC# set max-unknown-messages 5

[edit protocols pcep pce PCE]
user@PCC# set max-unknown-requests 5

[edit protocols pcep pce PCE]
user@PCC# set request-timer 50

[edit protocols pcep pce PCE]
user@PCC# up

[edit protocols pcep]
user@PCC# show
message-rate-limit 50;
max-provisioned-lsps 16000;
pce PCE {
    destination-ipv4-address 192.168.69.58;
    destination-port 4189;
    lsp-provisioning;
    lsp-cleanup-timer 50;
    request-timer 50;
    max-unknown-requests 5;
    max-unknown-messages 5;
    delegation-cleanup-timeout 20;
}

[edit protocols pcep]
user@PCC# commit
commit complete
```

Example: Configuring Path Computation Element Protocol for MPLS RSVP-TE with Support for PCE-Controlled Point-to-Multipoint LSPs

IN THIS SECTION

- Requirements | **2167**
- Overview | **2167**
- Configuration | **2169**

- Verification | 2183

This example shows how to configure the Path Computation Client (PCC) with the capability of reporting point-to-multipoint traffic engineered label-switched paths (TE LSPs) to a Path Computation Element (PCE).

Requirements

This example uses the following hardware and software components:

- Three routers that can be a combination of ACX Series, M Series, MX Series, or T Series routers.
- One virtual machine configured with Virtual Route Reflector (VRR) feature.
- A TCP connection to an external stateful PCE from the VRR.
- Junos OS Release 16.1 or later running on the PCC.

Before you begin:

- Configure the device interfaces.
- Configure MPLS and RSVP-TE.
- Configure OSPF or any other IGP protocol.

Overview

IN THIS SECTION

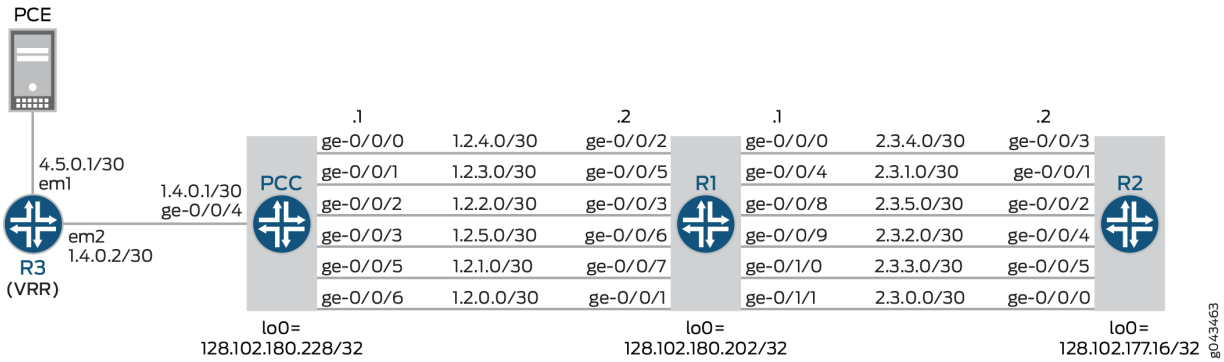
- Topology | 2168

After a PCEP session is established between a PCE and a PCC, the PCC reports all the LSPs in the system to the PCE for LSP state synchronization. This includes PCC-controlled, PCE-delegated, and PCE-initiated point-to-point LSPs. Starting with Junos OS Release 15.1F6 and 16.1R1, this capability is extended to report point-to-multipoint LSPs as well.

By default, PCE control of point-to-multipoint LSPs is not supported on a PCC. To add this capability, include the `p2mp-lsp-report-capability` statement at the `[edit protocols pcep pce pce-name]` or `[edit protocols pcep pce-group group-id]` hierarchy levels.

Topology

Figure 154: Example PCE-Controlled Point-to-Multipoint LSPs



In this example, PCC is the ingress router, Router R1 is the transit router, and Router R2 is the egress router. PCC is connected to a Virtual Route Reflector (VRR) that is connected to a PCE. There are many point-to-multipoint interfaces between PCC, Router R1, and Router R2.

The reporting of point-to-multipoint LSPs is executed as follows:

1. If Router PCC is configured with point-to-point and point-to-multipoint LSPs without the support for point-to-multipoint reporting capability, only the point-to-point LSPs are reported to the connected PCE. By default, a PCC does not support point-to-multipoint LSP reporting capability.
2. When Router PCC is configured with point-to-multipoint LSP reporting capability, PCC first advertises this capability to PCE through a report message.
3. By default, a PCE provides support for point-to-multipoint LSP capability. On receiving the PCC's advertisement for point-to-multipoint LSP capability, the PCE in return advertises its capability to the PCC.
4. On receiving the PCE's advertisement of the point-to-multipoint capability, PCC reports all branches of point-to-multipoint LSPs to the PCE using the update message.
5. Once all the LSPs are reported to the PCE, LSP state is synchronized between the PCE and PCC.

Configuration

IN THIS SECTION

- CLI Quick Configuration | 2169
- Procedure | 2174
- Results | 2178

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the `[edit]` hierarchy level.

PCC

```
set interfaces ge-0/0/0 unit 0 family inet address 1.2.4.1/30
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 1.2.3.1/30
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 1.2.2.1/30
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 family inet address 1.2.5.1/30
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces ge-0/0/4 unit 0 family inet address 1.4.0.1/30
set interfaces ge-0/0/4 unit 0 family mpls
set interfaces ge-0/0/5 unit 0 family inet address 1.2.1.1/30
set interfaces ge-0/0/5 unit 0 family mpls
set interfaces ge-0/0/6 unit 0 family inet address 1.2.0.1/30
set interfaces ge-0/0/6 unit 0 family mpls
set routing-options autonomous-system 100
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols mpls lsp-external-controller pccd pce-controlled-lsp pcc_delegated_no_cspf_* label-
switched-path-template lsp_template_no_cspf
set protocols mpls lsp-external-controller pccd pce-controlled-lsp
pce_initiated_no_ero_no_cspf_* label-switched-path-template lsp_template_no_cspf
set protocols mpls lsp-external-controller pccd pce-controlled-lsp
pce_initiated_loose_ero_no_cspf_* label-switched-path-template lsp_template_no_cspf
```

```
set protocols mpls traffic-engineering database import policy TE
set protocols mpls admin-groups violet 1
set protocols mpls admin-groups indigo 2
set protocols mpls admin-groups blue 3
set protocols mpls admin-groups green 4
set protocols mpls admin-groups yellow 5
set protocols mpls admin-groups orange 6
set protocols mpls label-switched-path lsp_template_no_cspf template
set protocols mpls label-switched-path lsp_template_no_cspf no-cspf
set protocols mpls label-switched-path lsp1-pcc to 128.102.177.16
set protocols mpls label-switched-path lsp2-pcc to 128.102.177.16
set protocols mpls label-switched-path lsp2-pcc lsp-external-controller pccd
set protocols mpls path loose-path 1.2.3.2 loose
set protocols mpls path strict-path 1.2.3.2 strict
set protocols mpls path strict-path 2.3.3.2 strict
set protocols mpls path path-B
set protocols mpls path path-C
set protocols mpls interface all
set protocols mpls interface ge-0/0/6.0 admin-group violet
set protocols mpls interface ge-0/0/5.0 admin-group indigo
set protocols mpls interface ge-0/0/2.0 admin-group blue
set protocols mpls interface ge-0/0/1.0 admin-group green
set protocols mpls interface ge-0/0/0.0 admin-group yellow
set protocols mpls interface ge-0/0/3.0 admin-group orange
set protocols mpls interface fxp0.0 disable
set protocols bgp group northstar type internal
set protocols bgp group northstar local-address 128.102.180.228
set protocols bgp group northstar family traffic-engineering unicast
set protocols bgp group northstar export TE
set protocols bgp group northstar neighbor 128.102.180.215
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/6.0
set protocols ospf area 0.0.0.0 interface ge-0/0/5.0
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0
set protocols ospf area 0.0.0.0 interface ge-0/0/4.0 interface-type p2p
set protocols pcep pce pce1 local-address 10.102.180.228
set protocols pcep pce pce1 destination-ipv4-address 10.102.180.246
set protocols pcep pce pce1 destination-port 4189
set protocols pcep pce pce1 pce-type active
```

```
set protocols pcep pce pce1 pce-type stateful
set protocols pcep pce pce1 lsp-provisioning
set protocols pcep pce pce1 lsp-cleanup-timer 0
set protocols pcep pce pce1 delegation-cleanup-timeout 60
set protocols pcep pce pce1 p2mp-lsp-report-capability
set policy-options policy-statement TE term 1 from family traffic-engineering
set policy-options policy-statement TE term 1 then accept
```

R1

```
set interfaces ge-0/0/0 unit 0 family inet address 2.3.4.1/30
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 1.2.0.2/30
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 1.2.4.2/30
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 family inet address 1.2.2.2/30
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces ge-0/0/4 unit 0 family inet address 2.3.1.1/30
set interfaces ge-0/0/4 unit 0 family mpls
set interfaces ge-0/0/5 unit 0 family inet address 1.2.3.2/30
set interfaces ge-0/0/5 unit 0 family mpls
set interfaces ge-0/0/6 unit 0 family inet address 1.2.5.2/30
set interfaces ge-0/0/6 unit 0 family mpls
set interfaces ge-0/0/7 unit 0 family inet address 1.2.1.2/30
set interfaces ge-0/0/7 unit 0 family mpls
set interfaces ge-0/0/8 unit 0 family inet address 2.3.5.1/30
set interfaces ge-0/0/8 unit 0 family mpls
set interfaces ge-0/0/9 unit 0 family inet address 2.3.2.1/30
set interfaces ge-0/0/9 unit 0 family mpls
set interfaces ge-0/1/0 unit 0 family inet address 2.3.3.1/30
set interfaces ge-0/1/0 unit 0 family mpls
set interfaces ge-0/1/1 unit 0 family inet address 2.3.0.1/30
set interfaces ge-0/1/1 unit 0 family mpls
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols mpls admin-groups violet 1
set protocols mpls admin-groups indigo 2
set protocols mpls admin-groups blue 3
set protocols mpls admin-groups green 4
set protocols mpls admin-groups yellow 5
set protocols mpls admin-groups orange 6
```

```
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols mpls interface ge-0/0/1.0 admin-group violet
set protocols mpls interface ge-0/0/7.0 admin-group indigo
set protocols mpls interface ge-0/0/3.0 admin-group blue
set protocols mpls interface ge-0/0/5.0 admin-group green
set protocols mpls interface ge-0/0/2.0 admin-group yellow
set protocols mpls interface ge-0/0/6.0 admin-group orange
set protocols mpls interface ge-0/1/1.0 admin-group violet
set protocols mpls interface ge-0/0/4.0 admin-group indigo
set protocols mpls interface ge-0/0/9.0 admin-group blue
set protocols mpls interface ge-0/1/0.0 admin-group green
set protocols mpls interface ge-0/0/0.0 admin-group yellow
set protocols mpls interface ge-0/0/8.0 admin-group orange
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/7.0
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0
set protocols ospf area 0.0.0.0 interface ge-0/0/5.0
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface ge-0/0/6.0
set protocols ospf area 0.0.0.0 interface ge-0/1/1.0
```

R2

```
set interfaces ge-0/0/0 unit 0 family inet address 2.3.0.2/30
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 2.3.1.2/30
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 2.3.5.2/30
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 family inet address 2.3.4.2/30
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces ge-0/0/4 unit 0 family inet address 2.3.2.2/30
set interfaces ge-0/0/4 unit 0 family mpls
set interfaces ge-0/0/5 unit 0 family inet address 2.3.3.2/30
set interfaces ge-0/0/5 unit 0 family mpls
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols mpls admin-groups violet 1
set protocols mpls admin-groups indigo 2
```

```
set protocols mpls admin-groups blue 3
set protocols mpls admin-groups green 4
set protocols mpls admin-groups yellow 5
set protocols mpls admin-groups orange 6
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols mpls interface ge-0/0/0.0 admin-group violet
set protocols mpls interface ge-0/0/1.0 admin-group indigo
set protocols mpls interface ge-0/0/4.0 admin-group blue
set protocols mpls interface ge-0/0/5.0 admin-group green
set protocols mpls interface ge-0/0/3.0 admin-group yellow
set protocols mpls interface ge-0/0/2.0 admin-group orange
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/4.0
set protocols ospf area 0.0.0.0 interface ge-0/0/5.0
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

R3

```
set interfaces em0 unit 0 family inet address 10.102.180.215/19
set interfaces em1 unit 0 family inet address 4.5.0.1/30
set interfaces em2 unit 0 family inet address 1.4.0.2/30
set interfaces em2 unit 0 family mpls
set routing-options router-id 128.102.180.215
set routing-options autonomous-system 100
set protocols topology-export
set protocols rsvp interface all
set protocols mpls lsp-external-controller pccd
set protocols mpls traffic-engineering database import igp-topology
set protocols mpls traffic-engineering database import policy TE
set protocols mpls interface all
set protocols bgp group northstar type internal
set protocols bgp group northstar local-address 128.102.180.215
set protocols bgp group northstar family traffic-engineering unicast
set protocols bgp group northstar neighbor 128.102.180.228
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface em2.0 interface-type p2p
```

```
set policy-options policy-statement TE from family traffic-engineering
set policy-options policy-statement TE then accept
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode*.

To configure the PCC router:

1. Configure the interfaces of Router PCC. To enable MPLS, include the protocol family on the interface so that the interface does not discard incoming MPLS traffic.

```
[edit interfaces]
user@PCC# set ge-0/0/0 unit 0 family inet address 1.2.4.1/30
user@PCC# set ge-0/0/0 unit 0 family mpls
user@PCC# set ge-0/0/1 unit 0 family inet address 1.2.3.1/30
user@PCC# set ge-0/0/1 unit 0 family mpls
user@PCC# set ge-0/0/2 unit 0 family inet address 1.2.2.1/30
user@PCC# set ge-0/0/2 unit 0 family mpls
user@PCC# set ge-0/0/3 unit 0 family inet address 1.2.5.1/30
user@PCC# set ge-0/0/3 unit 0 family mpls
user@PCC# set ge-0/0/4 unit 0 family inet address 1.4.0.1/30
user@PCC# set ge-0/0/4 unit 0 family mpls
user@PCC# set ge-0/0/5 unit 0 family inet address 1.2.1.1/30
user@PCC# set ge-0/0/5 unit 0 family mpls
user@PCC# set ge-0/0/6 unit 0 family inet address 1.2.0.1/30
user@PCC# set ge-0/0/6 unit 0 family mpls
```

2. Configure the autonomous system number for Router PCC.

```
[edit routing-options]
user@PCC# set autonomous-system 100
```

3. Enable RSVP on all interfaces of Router PCC, excluding the management interface.

```
[edit protocols]
user@PCC# set rsvp interface all
user@PCC# set rsvp interface fxp0.0 disable
```

4. Enable MPLS on all the interfaces of Router PCC, excluding the management interface.

```
[edit protocols]
user@PCC# set mpls interface all
user@PCC# set mpls interface fxp0.0 disable
```

5. Configure a dynamic LSP and disable automatic path computation for the LSP.

```
[edit protocols]
user@PCC# set mpls label-switched-path lsp_template_no_cspf template
user@PCC# set mpls label-switched-path lsp_template_no_cspf no-cspf
```

6. Configure point-to-multipoint LSPs and define external path computing entity for the LSP.

```
[edit protocols]
user@PCC# set mpls label-switched-path lsp1-pcc to 128.102.177.16
user@PCC# set mpls label-switched-path lsp2-pcc to 128.102.177.16
user@PCC# set mpls label-switched-path lsp2-pcc lsp-external-controller pccd
```

7. Enable external path computing for the MPLS LSPs and assign a template for externally provisioned LSPs.

```
[edit protocols]
user@PCC# set mpls lsp-external-controller pccd pce-controlled-lsp pcc_delegated_no_cspf_*
label-switched-path-template lsp_template_no_cspf
user@PCC# set mpls lsp-external-controller pccd pce-controlled-lsp
pce_initiated_no_ero_no_cspf_* label-switched-path-template lsp_template_no_cspf
user@PCC# set mpls lsp-external-controller pccd pce-controlled-lsp
pce_initiated_loose_ero_no_cspf_* label-switched-path-template lsp_template_no_cspf
```

8. Configure the LSPs that have local control and are overridden by the PCE-provided LSP parameters.

```
[edit protocols]
user@PCC# set mpls path loose-path 1.2.3.2 loose
user@PCC# set mpls path strict-path 1.2.3.2 strict
user@PCC# set mpls path strict-path 2.3.3.2 strict
user@PCC# set mpls path path-B
user@PCC# set mpls path path-C
```

9. Configure MPLS administrative group policies for constrained-path LSP computation.

```
[edit protocols]
user@PCC# set mpls admin-groups violet 1
user@PCC# set mpls admin-groups indigo 2
user@PCC# set mpls admin-groups blue 3
user@PCC# set mpls admin-groups green 4
user@PCC# set mpls admin-groups yellow 5
user@PCC# set mpls admin-groups orange 6
```

10. Assign the configured administrative group policies to Router PCC interfaces.

```
[edit protocols]
user@PCC# set mpls interface ge-0/0/6.0 admin-group violet
user@PCC# set mpls interface ge-0/0/5.0 admin-group indigo
user@PCC# set mpls interface ge-0/0/2.0 admin-group blue
user@PCC# set mpls interface ge-0/0/1.0 admin-group green
user@PCC# set mpls interface ge-0/0/0.0 admin-group yellow
user@PCC# set mpls interface ge-0/0/3.0 admin-group orange
```

11. Configure a traffic engineering database (TED) import policy.

```
[edit protocols]
user@PCC# set mpls traffic-engineering database import policy TE
```

12. Configure a BGP internal group.

```
[edit protocols]
user@PCC# set bgp group northstar type internal
```

```
user@PCC# set bgp group northstar local-address 128.102.180.228
user@PCC# set bgp group northstar neighbor 128.102.180.215
```

13. Configure traffic engineering for BGP and assign the export policy.

```
[edit protocols]
user@PCC# set bgp group northstar family traffic-engineering unicast
user@PCC# set bgp group northstar export TE
```

14. Configure OSPF area 0 on all the point-to-multipoint interfaces of Router PCC.

```
[edit protocols]
user@PCC# set ospf area 0.0.0.0 interface lo0.0
user@PCC# set ospf area 0.0.0.0 interface ge-0/0/6.0
user@PCC# set ospf area 0.0.0.0 interface ge-0/0/5.0
user@PCC# set ospf area 0.0.0.0 interface ge-0/0/2.0
user@PCC# set ospf area 0.0.0.0 interface ge-0/0/1.0
user@PCC# set ospf area 0.0.0.0 interface ge-0/0/0.0
user@PCC# set ospf area 0.0.0.0 interface ge-0/0/3.0
```

15. Configure OSPF area 0 on the point-to-point interface of Router PCC.

```
[edit protocols]
user@PCC# set ospf area 0.0.0.0 interface ge-0/0/4.0 interface-type p2p
```

16. Enable traffic engineering for OSPF.

```
[edit protocols]
user@PCC# set ospf traffic-engineering
```

17. Define the PCE that Router PCC connects to, and configure the the PCE parameters.

```
[edit protocols]
user@PCC# set pcep pce pce1 local-address 10.102.180.228
user@PCC# set pcep pce pce1 destination-ipv4-address 10.102.180.246
user@PCC# set pcep pce pce1 destination-port 4189
user@PCC# set pcep pce pce1 pce-type active
user@PCC# set pcep pce pce1 pce-type stateful
```

```
user@PCC# set pcep pce pce1 lsp-provisioning
user@PCC# set pcep pce pce1 lsp-cleanup-timer 0
user@PCC# set pcep pce pce1 delegation-cleanup-timeout 60
```

18. Configure Router PCC to enable point-to-multipoint LSP capability for external path computing.

```
[edit protocols]
set pcep pce pce1 p2mp-lsp-report-capability
```

19. Configure the traffic engineering policy.

```
[edit policy-options]
user@PCC# set policy-statement TE term 1 from family traffic-engineering
user@PCC# set policy-statement TE term 1 then accept
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@PCC# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 1.2.4.1/30;
        }
        family mpls;
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 1.2.3.1/30;
        }
        family mpls;
    }
}
ge-0/0/2 {
```

```
    unit 0 {
        family inet {
            address 1.2.2.1/30;
        }
        family mpls;
    }
}
ge-0/0/3 {
    unit 0 {
        family inet {
            address 1.2.5.1/30;
        }
        family mpls;
    }
}
ge-0/0/4 {
    unit 0 {
        family inet {
            address 1.4.0.1/30;
        }
        family mpls;
    }
}
ge-0/0/5 {
    unit 0 {
        family inet {
            address 1.2.1.1/30;
        }
        family mpls;
    }
}
ge-0/0/6 {
    unit 0 {
        family inet {
            address 1.2.0.1/30;
        }
        family mpls;
```

```
    }
}
```

```
user@PCC# show protocols
rsvp {
    interface all;
    interface fxp0.0 {
        disable;
    }
}
mpls {
    lsp-external-controller pccd {
        pce-controlled-lsp pcc_delegated_no_cspf_* {
            label-switched-path-template {
                lsp_template_no_cspf;
            }
        }
        pce-controlled-lsp pce_initiated_no_ero_no_cspf_* {
            label-switched-path-template {
                lsp_template_no_cspf;
            }
        }
        pce-controlled-lsp pce_initiated_loose_ero_no_cspf_* {
            label-switched-path-template {
                lsp_template_no_cspf;
            }
        }
    }
    traffic-engineering {
        database {
            import {
                policy TE;
            }
        }
    }
    admin-groups {
        violet 1;
        indigo 2;
        blue 3;
        green 4;
        yellow 5;
```

```
        orange 6;
    }
    label-switched-path lsp_template_no_cspf {
        template;
        no-cspf;
    }
    label-switched-path lsp1-pcc {
        to 128.102.177.16;
    }
    label-switched-path lsp2-pcc {
        to 128.102.177.16;
        lsp-external-controller pccd;
    }
    path loose-path {
        1.2.3.2 loose;
    }
    path strict-path {
        1.2.3.2 strict;
        2.3.3.2 strict;
    }
    path path-B;
    path path-C;
    interface all;
    interface ge-0/0/6.0 {
        admin-group violet;
    }
    interface ge-0/0/5.0 {
        admin-group indigo;
    }
    interface ge-0/0/2.0 {
        admin-group blue;
    }
    interface ge-0/0/1.0 {
        admin-group green;
    }
    interface ge-0/0/0.0 {
        admin-group yellow;
    }
    interface ge-0/0/3.0 {
        admin-group orange;
    }
    interface fxp0.0 {
        disable;
```

```
    }
}
bgp {
    group northstar {
        type internal;
        local-address 128.102.180.228;
        family traffic-engineering {
            unicast;
        }
        export TE;
        neighbor 128.102.180.215;
    }
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface lo0.0;
        interface ge-0/0/6.0;
        interface ge-0/0/5.0;
        interface ge-0/0/2.0;
        interface ge-0/0/1.0;
        interface ge-0/0/0.0;
        interface ge-0/0/3.0;
        interface ge-0/0/4.0 {
            interface-type p2p;
        }
    }
}
pcep {
    pce pce1 {
        local-address 10.102.180.228;
        destination-ipv4-address 10.102.180.246;
        destination-port 4189;
        pce-type active stateful;
        lsp-provisioning;
        lsp-cleanup-timer 0;
        delegation-cleanup-timeout 60;
        p2mp-lsp-report-capability;
    }
}
```

Verification

IN THIS SECTION

- Verifying LSP Configuration on the PCC | **2183**
- Verifying PCE Configuration on the PCC | **2186**

Confirm that the configuration is working properly.

Verifying LSP Configuration on the PCC

Purpose

Verify the LSP type and running state of the point-to-multipoint LSP.

Action

From operational mode, run the `show mpls lsp extensive` command.

```
user@PCC> show mpls lsp extensive
Ingress LSP: 2 sessions

128.102.177.16
  From: 128.102.180.228, State: Up, ActiveRoute: 0, LSPname: lsp1-pcc
  ActivePath:  (primary)
  LSPtype: Static Configured, Penultimate hop popping
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                    State: Up
    Priorities: 7 0
    SmartOptimizeTimer: 180
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 2)
 1.2.1.2 S 2.3.0.2 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
          1.2.1.2 2.3.0.2
    6 Jul 12 14:44:10.620 Selected as active path
    5 Jul 12 14:44:10.617 Record Route:  1.2.1.2 2.3.0.2
    4 Jul 12 14:44:10.615 Up
```

```
    3 Jul 12 14:44:10.175 Originate Call
    2 Jul 12 14:44:10.174 CSPF: computation result accepted  1.2.1.2 2.3.0.2
    1 Jul 12 14:43:41.442 CSPF failed: no route toward 128.102.177.16[2 times]
  Created: Tue Jul 12 14:42:43 2016

128.102.177.16
  From: 128.102.180.228, State: Up, ActiveRoute: 0, LSPname: lsp2-pcc
  ActivePath:  (primary)
  LSPtype: Externally controlled - static configured, Penultimate hop popping
  LSP Control Status: Externally controlled
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                    State: Up
    Priorities: 7 0
    External Path CSPF Status: external
    SmartOptimizeTimer: 180
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
          1.2.4.2 2.3.0.2
   50 Jul 12 14:50:14.699 EXTCTRL LSP: Sent Path computation request and LSP status
   49 Jul 12 14:50:14.698 EXTCTRL_LSP: Computation request/lsp status contains:  signalled bw 0
req BW 0 admin group(exclude 0 include any 0 include all 0) priority setup 7 hold 0
   48 Jul 12 14:49:27.859 EXTCTRL LSP: Sent Path computation request and LSP status
   47 Jul 12 14:49:27.859 EXTCTRL_LSP: Computation request/lsp status contains:  signalled bw 0
req BW 0 admin group(exclude 0 include any 0 include all 0) priority setup 7 hold 0
   46 Jul 12 14:49:27.858 EXTCTRL LSP: Sent Path computation request and LSP status
   45 Jul 12 14:49:27.858 EXTCTRL_LSP: Computation request/lsp status contains:  signalled bw 0
req BW 0 admin group(exclude 0 include any 0 include all 0) priority setup 7 hold 0
   44 Jul 12 14:49:27.858 EXTCTRL_LSP: Control status became external
   43 Jul 12 14:49:03.746 EXTCTRL_LSP: Control status became local
   42 Jul 12 14:46:52.367 EXTCTRL LSP: Sent Path computation request and LSP status
   41 Jul 12 14:46:52.367 EXTCTRL_LSP: Computation request/lsp status contains:  signalled bw 0
req BW 0 admin group(exclude 0 include any 0 include all 0) priority setup 7 hold 0
   40 Jul 12 14:46:52.367 EXTCTRL LSP: Sent Path computation request and LSP status
   39 Jul 12 14:46:52.366 EXTCTRL_LSP: Computation request/lsp status contains:  signalled bw 0
req BW 0 admin group(exclude 0 include any 0 include all 0) priority setup 7 hold 0
   38 Jul 12 14:46:52.366 EXTCTRL_LSP: Control status became external
   37 Jul 12 14:46:41.584 Selected as active path
   36 Jul 12 14:46:41.565 Record Route:  1.2.4.2 2.3.0.2
   35 Jul 12 14:46:41.565 Up
   34 Jul 12 14:46:41.374 EXTCTRL_LSP: Applying local parameters with this signalling attempt
   33 Jul 12 14:46:41.374 Originate Call
   32 Jul 12 14:46:41.374 CSPF: computation result accepted  1.2.4.2 2.3.0.2
   31 Jul 12 14:46:28.254 EXTCTRL_LSP: Control status became local
```

```
   30 Jul 12 14:46:12.494 EXTCTRL LSP: Sent Path computation request and LSP status
   29 Jul 12 14:46:12.494 EXTCTRL_LSP: Computation request/lsp status contains:  signalled bw 0
req BW 0 admin group(exclude 0 include any 0 include all 0) priority setup 7 hold 0
   28 Jul 12 14:45:43.164 EXTCTRL LSP: Sent Path computation request and LSP status
   27 Jul 12 14:45:43.164 EXTCTRL_LSP: Computation request/lsp status contains:  signalled bw 0
req BW 0 admin group(exclude 0 include any 0 include all 0) priority setup 7 hold 0
   26 Jul 12 14:45:13.424 EXTCTRL LSP: Sent Path computation request and LSP status
   25 Jul 12 14:45:13.423 EXTCTRL_LSP: Computation request/lsp status contains:  signalled bw 0
req BW 0 admin group(exclude 0 include any 0 include all 0) priority setup 7 hold 0
   24 Jul 12 14:44:44.774 EXTCTRL LSP: Sent Path computation request and LSP status
   23 Jul 12 14:44:44.773 EXTCTRL_LSP: Computation request/lsp status contains:  signalled bw 0
req BW 0 admin group(exclude 0 include any 0 include all 0) priority setup 7 hold 0
   22 Jul 12 14:44:15.053 EXTCTRL LSP: Sent Path computation request and LSP status
   21 Jul 12 14:44:15.053 EXTCTRL_LSP: Computation request/lsp status contains:  signalled bw 0
req BW 0 admin group(exclude 0 include any 0 include all 0) priority setup 7 hold 0
   20 Jul 12 14:43:45.705 EXTCTRL LSP: Sent Path computation request and LSP status
   19 Jul 12 14:43:45.705 EXTCTRL_LSP: Computation request/lsp status contains:  signalled bw 0
req BW 0 admin group(exclude 0 include any 0 include all 0) priority setup 7 hold 0
   18 Jul 12 14:43:45.705 EXTCTRL LSP: Sent Path computation request and LSP status
   17 Jul 12 14:43:45.705 EXTCTRL_LSP: Computation request/lsp status contains:  signalled bw 0
req BW 0 admin group(exclude 0 include any 0 include all 0) priority setup 7 hold 0
   16 Jul 12 14:43:45.705 EXTCTRL_LSP: Control status became external
   15 Jul 12 14:43:42.398 CSPF failed: no route toward 128.102.177.16
   14 Jul 12 14:43:13.009 EXTCTRL_LSP: Control status became local
   13 Jul 12 14:43:13.009 EXTCTRL LSP: Sent Path computation request and LSP status
   12 Jul 12 14:43:13.008 EXTCTRL_LSP: Computation request/lsp status contains:  signalled bw 0
req BW 0 admin group(exclude 0 include any 0 include all 0) priority setup 7 hold 0
   11 Jul 12 14:42:43.343 EXTCTRL LSP: Sent Path computation request and LSP status
   10 Jul 12 14:42:43.343 EXTCTRL_LSP: Computation request/lsp status contains:  signalled bw 0
req BW 0 admin group(exclude 0 include any 0 include all 0) priority setup 7 hold 0
    9 Jul 12 14:42:43.343 EXTCTRL LSP: Sent Path computation request and LSP status
    8 Jul 12 14:42:43.343 EXTCTRL_LSP: Computation request/lsp status contains:  signalled bw 0
req BW 0 admin group(exclude 0 include any 0 include all 0) priority setup 7 hold 0
    7 Jul 12 14:42:43.342 EXTCTRL LSP: Sent Path computation request and LSP status
    6 Jul 12 14:42:43.342 EXTCTRL_LSP: Computation request/lsp status contains:  signalled bw 0
req BW 0 admin group(exclude 0 include any 0 include all 0) priority setup 7 hold 0
    5 Jul 12 14:42:43.341 EXTCTRL_LSP: Control status became external
    4 Jul 12 14:42:43.337 EXTCTRL_LSP: Control status became local
    3 Jul 12 14:42:43.323 EXTCTRL LSP: Sent Path computation request and LSP status
    2 Jul 12 14:42:43.323 EXTCTRL_LSP: Computation request/lsp status contains:  signalled bw 0
req BW 0 admin group(exclude 0 include any 0 include all 0) priority setup 7 hold 0
    1 Jul 12 14:42:43.258 EXTCTRL LSP: Awaiting external controller connection
  Created: Tue Jul 12 14:42:43 2016
```

```
Total 2 displayed, Up 2, Down 0

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Meaning

The output displays the lsp2-pcc LSP as the PCE-controlled LSP.

Verifying PCE Configuration on the PCC

Purpose

Verify the PCE parameters configuration and PCE state.

Action

From operational mode, run the `show path-computation-client active-pce` command.

```
user@PCC> show path-computation-client active-pce
PCE pce1
-----------------------------------------
General
    PCE IP address           : 10.102.180.246
    Local IP address         : 10.102.180.228
    Priority                 : 0
    PCE status               : PCE_STATE_UP
    Session type             : PCE_TYPE_STATEFULACTIVE
    LSP provisioning allowed : On
    P2MP LSP report allowed  : On
    P2MP LSP update allowed  : Off
    P2MP LSP init allowed    : Off
    PCE-mastership           : main

Counters
    PCReqs              Total: 0           last 5min: 0           last hour: 0
    PCReps              Total: 0           last 5min: 0           last hour: 0
    PCRpts              Total: 12          last 5min: 0           last hour: 12
```

```
    PCUpdates           Total: 1            last 5min: 0            last hour: 1
    PCCreates           Total: 0            last 5min: 0            last hour: 0

Timers
    Local  Keepalive timer:   30 [s]  Dead timer:  120 [s]  LSP cleanup timer:    0 [s]
    Remote Keepalive timer:   30 [s]  Dead timer:  120 [s]  LSP cleanup timer:    0 [s]

Errors
    PCErr-recv
    PCErr-sent
            Type: 1             Value: 2            Count: 1
    PCE-PCC-NTFS
    PCC-PCE-NTFS
```

Meaning

The output displays the active PCE that Router PCC is connected to, and the pce1 PCE parameters and state.

Understanding Path Computation Element Protocol for MPLS RSVP-TE with Support for PCE-Initiated Point-to-Multipoint LSPs

IN THIS SECTION

- Benefits of PCE-Initiated Point-to-Multipoint LSPs | **2188**
- Signaling of PCE-Initiated Point-to-Multipoint LSPs | **2188**
- Behavior of PCE-Initiated Point-to-Multipoint LSPs After PCEP Session Failure | **2188**
- Configuring PCE-Initiated Point-to-Multipoint LSP Capability | **2188**
- Supported and Unsupported Features for PCE-Initiated Point-to-Multipoint LSPs | **2189**
- Mapping PCE-initiated Point-To-Multipoint LSPs to MVPN | **2190**

With the introduction of point-to-multipoint PCE-initiated LSPs, a PCE can initiate and provision a point-to-multipoint LSP dynamically without the need for local LSP configuration on the PCC. This enables the PCE to control the timing and sequence of the point-to-multipoint path computations within and across Path Computation Element Protocol (PCEP) sessions, thereby creating a dynamic network that is centrally controlled and deployed.

Benefits of PCE-Initiated Point-to-Multipoint LSPs

Meets the requirements of point-to-multipoint traffic engineering LSP placement in response to application demands through dynamic creation and tear down of point-to-multipoint LSPs, thereby creating a dynamic network that is centrally controlled and deployed.

Signaling of PCE-Initiated Point-to-Multipoint LSPs

The signaling of PCE-initiated point-to-multipoint LSPs is as follows:

- **When a new branch is added (Grafting)**—Only the new branch sub-LSP is signaled and does not result in re-signaling of the entire point-to-multipoint tree.

  If any topology changes occurred before provisioning of the new sub-LSP, then the Path Computation Server (PCS) re-computes the entire point-to-multipoint tree and updates the point-to-multipoint LSP using a PC update message.

- **When a branch is deleted (Pruning)**—The deleted branch sub-LSP is torn down and does not result in re-signaling of the entire point-to-multipoint tree.

- **When a branch sub-LSP parameter is changed**—Change in sub-LSP parameters, such as Explicit Route Object (ERO), bandwidth, or priority, can happen either because of optimization, or on user request. If there is a re-signaling request for a sub-LSP, the entire point-to-multipoint tree is re-signaled, and then the switchover to the new instance happens once the new instances of all the branches are up.

- **When a branch sub-LSP path fails**—An error is reported to the PCS for the failed branch sub-LSP. On receiving the new ERO from the PCS, the entire point-to-multipoint tree is re-signaled along with the failed branch sub-LSP, and the switchover to the new instance happens in a make-before-break (MBB) fashion.

Behavior of PCE-Initiated Point-to-Multipoint LSPs After PCEP Session Failure

When a PCEP session fails, the PCE-initiated point-to-multipoint LSPs are orphaned until the expiration of the `state timeout` timer. After the `state timeout` timer expires, the PCE-initiated LSPs are cleaned up.

To obtain control of a PCE-initiated point-to-multipoint LSP after a PCEP session failure, the primary or secondary PCE sends a `PCInitiate` message before the `state timeout` timer expires.

Configuring PCE-Initiated Point-to-Multipoint LSP Capability

By default, the creation and provisioning of point-to-multipoint LSPs by a PCE is not supported on a PCC. To enable this capability, include the `p2mp-lsp-init-capability` and `p2mp-lsp-update-capability` statements at the `[edit protocols pcep pce` *`pce-name`*`]` or `[edit protocols pcep pce-group group-id]` hierarchy levels.

The p2mp-lsp-init-capability statement provides the capability to provision point-to-multipoint RSVP-TE LSPs by a PCE. The p2mp-lsp-update-capability statement provides the capability to update point-to-multipoint RSVP-TE LSP parameters by a PCE.

Supported and Unsupported Features for PCE-Initiated Point-to-Multipoint LSPs

The following features are supported with PCE-initiated point-to-multipoint LSPs:

- Partial compliance with the Internet draft draft-ietf-pce-stateful-pce-p2mp (expires October 2018), *Path Computation Element (PCE) Protocol Extensions for Stateful PCE usage for Point-to-Multipoint Traffic Engineering Label Switched Paths*.
- Starting in Junos OS Release 21.1R1, we support nonstop active routing (NSR) for PCE-initiated RSVP-based point-to-multipoint LSPs. Only the primary Routing Engine maintains the PCEP session with the controller. It synchronizes all RSVP LSPs initiated by PCEs, including multicast flow specifications for any PCE-initiated P2MP LSPs, with the backup Routing Engine. During a switchover, the PCEP session goes down and re-establishes when the backup Routing Engine becomes the primary Routing Engine. This reduces traffic loss for the traffic carried over PCE-initiated RSVP LSPs during Routing Engine switchovers. This feature is enabled when NSR is configured.

The following features are not supported with PCE-initiated point-to-multipoint LSPs:

- Delegation of point-to-multipoint locally controlled LSP.
- LSP control delegation.
- Interior gateway protocol (IGP) extension for PCE discovery within an IGP routing domain.
- Request/response messaging.
- Direct movement of branch sub-LSP from one point-to-multipoint tree to another.

  The same can be achieved by deleting a branch sub-LSP from the first point-to-multipoint tree and re-adding it to another after the PCReport message indicates LSP removal from the device.
- IPv6 is not supported.
- SERO based signalling is not supported.
- Empty-ERO feature is not supported.
- Link protection is not supported.

Mapping PCE-initiated Point-To-Multipoint LSPs to MVPN

You can associate a single or range of MVPN multicast flows (S,G) to a dynamically created PCE-initiated point-to-multipoint label-switched path (LSP). You can specify only selective types of flows for this feature to work. This includes:

- Route distinguisher (RD) which is mapped to the MVPN routing-instance.
- (S,G) which is the source of a multicast packet and destination multicast group address. This is used to filter incoming traffic for mapping it to the tunnel.
- Point-to-multipoint LSP that is used to send traffic that matches the above-mentioned flow specification.

For more details, see Internet draft draft-ietf-pce-pcep-flowspec-05 (expires February 16, 2020) *PCEP Extension for Flow Specification*.

The current implementation of this feature does not implement the following sections of the draft:

- Section 3.1.2—Advertising PCE capabilties in IGP
- Section 3.2—PCReq and PCRep message
- Section 7—Most of the flow specifications, except route distinguThe current implementation of this feature does not supporisher and IPv4 multicast flow specifications, are not supported.

To enable the mapping of PCE-initiated point-to-multipoint LSPs to MVPN:

- Include the `pce_traffic_steering` statement at the `[edit protocols pcep pce` *`pce-id`*`]` hierarchy level to indicate the support for flow specification capability (also called traffic steering) by the PCC.
- Include the `external-controller` statement at the `[edit routing-instances` *`routing-instance-name`* `provider-tunnel]` hierarchy level.

  The presence of `external-controller` in the provider-tunnel configuration for MVPN indicates that the point-to-multipoint LSP and (S,G) for this MVPN instance can be provided by the external controller. This enables the external controller to dynamically configure (S,G) and point-to-multipoint LSP for MVPN.

Take the following into consideration for mapping of PCE-initiated point-to-multipoint LSPs to MVPN:

- If you do not enable the `external-controller` *`pccd`* statement for a particular MVPN instance, then the PCCD process does not configure (S,G) dynamically.
- If you disable the `external-controller` *`pccd`* configuration from the CLI, then the dynamically learned multicast flows (S,G) for that particular MVPN instance is deleted and reported to the external controller.

- When (S,G) is already configured from the CLI, the PCC cannot configure (S,G) dynamically as local configuration has a higher priority.
- If any particular (S,G) is learned from the external controller dynamically and then you configure the same (S,G) for the same MVPN instance, then the dynamically learned (S,G) is deleted and reported to the external controller through the PCC.
- If the routing protocol process reboots, then the PCCD process reconfigures all the (S,G) again.
- If the PCCD process reboots, then MVPN reports all PCCD configured (S,G) to the external controller.
- If user enables `external-controller` *pccd* for a particular MVPN instance, then MVPN requests the PCCD process to configure (S,G), if any present.
- If there are major configuration changes to a particular MPVN instance, then MVPN requests the PCCD process to reconfigure all (S,G) for that particular MVPN instance.
- All flow specifications associated with any PCE-initiated point-to-multipoint LSP must have the same RD. During PC initiation if all flow specifications do not have the same RD, then the PC initiate message is dropped with an error.
- You can associate a point-to-multipoint LSP only with selective type of flow specifications, otherwise the PC initiate message is dropped with an error.
- During PC update if all flow specifications do not have same the RD either due to a new flow specification addition, or due to existing flow specification update, then the PCC drops the update message.
- During PC update if all flow specifications do not meet the selective condition either due to new flow specification addition, or due to existing flow specifications update, then the PCC drops the update message.
- Behavior for mapping of PCE-initiated point-to-multipoint LSP with MVPN routing-instance and mapping of static (locally configured) point-to-multipoint LSP with MVPN instance is the same at user level.
- A flow specification ID can be associated with only one point-to-multipoint LSP. To associate the same RD and (S,G) to multiple point-to-multipoint LSPs, you can add multiple flow specifications with different IDs and same RD & (S,G).
- For PCEP-mapped dynamic (S,G), the threshold value is always the default value of *0*.
- There is no limit on the number of flow specifications mapped to a single PCE-initiated point-to-multipoint LSP.
- The current implementation of this feature does not support:

- Reporting of forwarding states that are associated with the point-to-multipoint LSP.
- Inclusive provider tunnel dynamic configuration
- Mapping for MVPN ingress replication tunnel
- Programmable routing protocol process (prpd)
- Reporting of CLI configured point-to-multipoint LSP which is mapped to MVPN multicast flows (S,G).

SEE ALSO

Configuring Nonstop Active Routing

Enable Segment Routing for the Path Computation Element Protocol

SUMMARY

You can enable segment routing or Source Packet Routing in Networking (SPRING) traffic-engineering (SR-TE) with the Path Computation Element Protocol (PCEP) for traffic steering. With this support, the advantages of segment routing are extended to the label-switched paths (LSPs) that are externally controlled by a Path Computation Element (PCE).

IN THIS SECTION

- Segment Routing for the Path Computation Element Protocol Overview | **2192**
- Example: Configure Segment Routing for the Path Computation Element Protocol | **2200**

Segment Routing for the Path Computation Element Protocol Overview

IN THIS SECTION

- Benefits of Segment Routing for PCEP | **2193**
- Segment Routing for Traffic Engineering | **2193**
- Junos OS Implementation of Segment Routing for PCEP | **2194**
- Segment Routing for PCEP Limitations and Unsupported Features | **2199**

Benefits of Segment Routing for PCEP

- Setting up of LSPs through an external controller provides a global view of per-LSP and per-device bandwidth demand on the network, enabling online and real-time constraint-based path computation.

  The advantages of segment routing are extended to the LSPs initiated by an external controller, also known as a Path Computation Element (PCE), augmenting the benefits of external path computing in an MPLS network.

- A Path Computation Client (PCC, which is an ingress MX Series router) with delegation capability can take back control of the delegated segment routing LSPs from the PCE when the PCEP session goes down; the LSPs would otherwise be deleted from the PCC. You can thus ensure LSP data protection by averting a situation where packets are silently discarded or dropped (also known as a null route condition).

Segment Routing for Traffic Engineering

Segment routing can operate over an IPv4 or IPv6 data plane, and supports equal-cost multipath (ECMP). With the IGP extensions built into it, segment routing integrates with the rich multiservice capabilities of MPLS, including Layer 3 VPN, Virtual Private Wire Service (VPWS), Virtual Private LAN Service (VPLS), and Ethernet VPN (EVPN).

Some of the high-level components of the segment routing–traffic engineering (SR–TE) solution include:

- Use of an IGP for advertising link characteristics. This functionality is similar to RSVP-TE.
- Use of Constrained Shortest Path First (CSPF) on the ingress device or the PCE.
- Use of an IGP for advertising labels for links.

  In SR-TE functionality:

  1. The ingress device constructs an LSP by stacking the labels of the links that it wants to traverse.
  2. The per-link IGP advertisement is combined with label stacking to create source-routed LSPs on the ingress device, so the transit devices are not aware of the end-to-end LSPs.
  3. LSPs are created between edge nodes without placing any per-LSP memory requirements on the transit devices. (Creation of such LSPs is enabled as there is no per-LSP signaling in SR-TE.)
  4. Per-neighbor labels are stacked, which results in the management of a large number of labels, leading to control plane scaling.

Junos OS Implementation of Segment Routing for PCEP

Junos OS implements segment routing for PCEP for two types of LSPs—PCE-initiated LSPs and PCE-delegated LSPs.

PCE-Initiated Segment Routing LSPs

The PCE-initiated segment routing LSPs are those LSPs that the PCE creates for the adjacency and node segments

The PCE performs the following functions:

1. Computes the path of the segment routing LSP.
2. Provisions the LSP on the Path Computation Client (PCC) using PCEP segment routing extensions.
3. Parses the PCEP segment routing extensions.
4. Creates a tunnel route on the PCC that has its own preference value and is made available in the inet.3 routing table to resolve IP traffic and services like any other tunnel route.

The PCC performs the following functions:

1. Selects the outgoing interface based on the first network access identifier (NAI) in the source Explicit Route Object (S-ERO).

   Junos OS supports S-EROs that contain the first hop as a strict hop; Junos OS doesn't support selection of the outgoing interface on the PCC based on a loose-hop node segment ID (SID). However, the remaining hops can be loose. No specific processing is done for the S-EROs that are beyond the first hop, other than to simply use the label for next-hop creation.

2. Rejects the S-ERO if:

   - The S-ERO does not have labels in it.
   - The S-ERO carries more than six hops.

   The PCC creates an equal-cost multipath (ECMP) route when there are multiple LSPs to the same destination with the same metric.

3. Waits for the PCE to process any event that leads to a change in the segment routing LSP after it is provisioned—for example, if the label is changed or withdrawn, or if one of the interfaces traversed by the LSP goes down.

When the PCEP session goes down, the PCE-initiated segment routing LSP:

1. Remains up for 300 seconds.

2. Is deleted from the PCC after 300 seconds.

For more details, see Internet drafts draft-ietf-pce-lsp-setup-type-03.txt (expires December 25, 2015), *Conveying path setup type in PCEP messages*; and draft-ietf-pce-segment-routing-06.txt (expires February 10, 2016), *PCEP Extensions for Segment Routing*.

PCE-Delegated Segment Routing LSPs

The PCE-delegated segment routing LSPs are those LSPs that the PCC configures locally and then delegates to a PCE controller.

NOTE: Junos OS Release 20.1R1 supports:

- PCE delegation capability only for noncolored segment routing LSPs with IPv4 destinations.
- Delegation and reporting of only the first segment of a segment list to an external controller. Multiple segments are not supported for PCE delegation.

The PCC can delegate a segment routing LSP to an external controller (the PCE) in the following ways:

- **Initial delegation**—The local LSPs are yet to be configured on the PCC, and the delegation of the LSP happens at the time the LSP is configured.
- **Delegation of existing LSP**—The local LSPs are configured on the PCC, and the delegation of the LSP happens after the source-routing path is configured. That is, the delegation capability is enabled on existing segment routing LSPs.

After delegating a segment routing LSP, the PCE controls the delegated LSPs and can modify the LSP attributes for path computation. The LSP control reverts to the PCC when the PCEP session between the PCC and the PCE goes down. The PCE-delegated LSPs have an advantage over PCE-initiated LSPs in case the PCEP session goes down. In the case of PCE-initiated LSPs, when the PCEP session is down, the LSPs are deleted from the PCC. However, in the case of PCE-delegated LSPs, when the PCEP session goes down, the PCC takes back control of the delegated LSPs from the PCE. As a result, with PCE-delegated LSPs, we avert a situation where packets are silently discarded (also known as a null route condition) when the session goes down.

The following types of segment routing LSPs support the PCE-delegation capability:

- **Static LSPs**—Statically configured source-routing paths that have the entire label stack statically configured.
- **Auto-translated LSPs**—Statically configured source-routing paths that are automatically translated.
- **Computed LSPs**—Statically configured source-routing paths that are computed with distributed Constrained Shortest Path First (CSPF).

- **Dynamic LSPs**—Dynamically created tunnels triggered through the Dynamic Tunnel Module that have last-hop ERO resolution.

Depending on the source of the segment routing LSP, you can configure the delegation capability on the PCC. To enable delegation of segment routing LSPs, include the `lsp-external-controller` *pccd* statement at the appropriate level under the `[edit protocols source-packet-routing]` hierarchy.

Table 2 shows a mapping of the LSP source to the corresponding configuration hierarchy level at which the delegation capability is enabled.

NOTE: You must include the `lsp-external-controller` *pccd* statement at the `[edit protocols source-packet-routing]` and `[edit protocols mpls]` hierarchy levels before configuring the delegation capability on the PCC.

Table 47: Mapping of Segment Routing LSP Source with Configuration Hierarchy

Source of Segment Routing LSP	Configuration Hierarchy
• Auto-translated LSPs • Static LSPs	Primary segment list at `[edit protocols source-packet-routing source-routing-path` *lsp-name* `primary` *path-name*`]`
Computed LSPs (distributed CSPF)	Primary segment list of the source-routing path at: • `[edit protocols source-packet-routing source-routing-path` *lsp-name* `primary` *path-name* `compute` *profile-name*`]` • `[edit protocols source-packet-routing source-routing-path` *lsp-name* `primary` *path-name*`]`
Dynamic LSPs	Primary segment list of the source-routing path template at: • `[edit protocols source-packet-routing source-routing-path-template` *template-name* `primary` *primary-segment-list-name*`]` • `[edit protocols source-packet-routing source-routing-path-template` *template-name*`]`

You can view the control status of the SR-TE LSPs from the *show spring-traffic-engineering* command output.

Table 3 displays the PCEP interaction when the `lsp-external-controller` statement is configured for a source-routing path.

Table 48: PCEP Interaction LSP Delegation

lsp-external-controller Configuration Hierarchy	source-routing-path Delegation State	PCEP Interaction Between PCC and PCE
Primary segment list of source-routing path	Initial delegation	1. A PCReport message is sent to the PCE for delegation. The PCReport contains only constraints and path details (such as ERO). 2. PCE calculates the path for LSP and reports path to be in the down state. 3. No route is programmed by the local LSP until the controller computes the ERO and notifies the result to the PCC through PCUpdate. The same behavior is seen when the routing protocol process (rpd) restarts or a Routing Engine switchover happens.

Table 48: PCEP Interaction LSP Delegation *(Continued)*

lsp-external-controller Configuration Hierarchy	source-routing-path Delegation State	PCEP Interaction Between PCC and PCE
Primary segment list of source-routing path	Delegation of existing path	1. A PCReport is sent to the PCE for delegation. The PCReport contains only constraints and path details (such as ERO). 2. A corresponding primary segment is delegated to the PCE. 3. PCE calculates the path for the LSP. 4. The primary segment continues to contribute to the route as determined by the local configuration or computation until a PCUpdate is received from the PCE. • If Seamless BFD (S-BFD) is not configured for the primary segment, then there is no further update to the route and the LSP state is also not monitored and reported to the PCE. The LSP state at this point is reported as up or down depending on whether path computation had succeeded at that point. • If S-BFD is configured for the primary segment, then the state of the primary segment is tracked and reported to the PCE. If BFD detects the primary segment to be down, the corresponding primary path is removed from the route. The same route that was previously calculated is reprogrammed if that path is up now. 5. If a PCUpdate message is received from the PCE, SR-TE uses the received parameter to set up the path for which the PCReport message was sent. The programmed path then includes only the segment list received from the PCE, and all the other segment lists that were previously programmed are removed. This reprogramming of the route happens in a make-before-break fashion.

Table 48: PCEP Interaction LSP Delegation *(Continued)*

lsp-external-controller Configuration Hierarchy	source-routing-path Delegation State	PCEP Interaction Between PCC and PCE
Primary segment of source-routing path	Delegation is not configured or has been deleted.	The segment list from the PCE (if available) is no longer used and the computation result from the local configuration is used. When the local result for the segment list is available, the corresponding segment list is used to program the route in a make-before-break fashion.
Segment list of source-routing path	Delegation is enabled after LSP is configured.	Delegation functionality is triggered for the primary segment list under the source-routing path.
Segment list of source-routing path	Delegation is not configured or has been deleted.	Delegation functionality is removed from the primary segment list under the source-routing path.
Primary segment list of source-routing path template	Delegation is enabled after LSP is configured.	• Under the source-routing path template—Delegation functionality is triggered for the entire source-routing path. Template configurations can be applied only to the Dynamic Tunnel Module. • Under the primary path in the source-routing path template—Delegation functionality is triggered for that particular primary path according to the configuration.
Primary segment list of source-routing path template	Delegation is not configured or has been deleted.	Delegation functionality is removed from all the source-routing paths and primary paths that match the template configuration.

Segment Routing for PCEP Limitations and Unsupported Features

The support of segment routing for PCEP does not add to the performance burden on the system. However, it has the following limitations:

- An SR-TE LSP is not locally protected on the PCC. When the LSP is more than six hops, no service is provided on the LSP other than to carry plain IP traffic.

- Graceful Routing Engine switchover (GRES) and unified in-service software upgrade (unified ISSU) are not supported.
- Nonstop active routing (NSR) is not supported.
- IPv6 is not supported.
- PCE-delegated LSPs do not support the following:
  - Colored SR-TE LSPs
  - IPv6 LSPs
  - Secondary segment list of the source-routing path. Only one path of the segment list can be delegated.
  - Multisegment standard. Only the first segment of the segment list is delegated and reported to the controller.

Example: Configure Segment Routing for the Path Computation Element Protocol

IN THIS SECTION

- Requirements | **2200**
- Overview | **2201**
- Configuration | **2203**
- Verification | **2212**

This example shows how to configure segment routing or Source Packet Routing in Networking (SPRING) traffic-engineering (SR-TE) for the Path Computation Element Protocol (PCEP). In the configuration, we leverage the advantages of segment routing with the benefits of external path computing for efficient traffic engineering.

Requirements

This example uses the following hardware and software components:

- Four MX Series 5G Universal Routing Platforms, where the ingress MX Series router is the Path Computation Client (PCC).
- A TCP connection from the PCC to an external stateful Path Computation Element (PCE).
- Junos OS Release 17.2 or later running on the PCC for the implementation of PCE-initiated LSPs.

For PCE-delegation functionality, you must run Junos OS Release 20.1R1 or a later release.

Before you begin:

- Configure the device interfaces.
- Configure MPLS.
- Configure IS-IS.

Overview

IN THIS SECTION

Topology | 2202

The Junos OS implementation of segment routing for PCEP includes PCE-initiated and PCE-delegated SR-TE LSPs.

- The implementation of PCE-initiated LSPs is introduced in Junos OS Release 17.2R1, where the traffic-engineering capabilities of segment routing are supported in PCEP sessions for LSPs initiated by a PCE. The PCE creates the LSPs for the adjacency and node segments. Tunnel routes are created in the inet.3 routing table of the PCC corresponding to the PCE-initiated SR-TE LSPs.
- The implementation of PCE-delegated LSPs is introduced in Junos OS Release 20.1R1, where the locally configured IPv4 noncolored segment routing LSPs on the PCC can be delegated to a PCE controller. The PCE then controls the LSP and can modify LSP attributes for path computation.

The PCE-delegated LSPs have an advantage over PCE-initiated LSPs at the time the PCEP session goes down. In the case of PCE-initiated LSPs, when the PCEP session is down, the LSPs are deleted from the PCC. However, in the case of PCE-delegated LSPs, when the PCEP session goes down, the PCC takes back control of the delegated LSPs from the PCE. As a result, with PCE-delegated LSPs, we avert a situation where packets are silently discarded (also known as a null route condition) when the PCEP session goes down.

To enable segment routing for PCEP:

For PCE-initiated segment routing LSPs:

1. Enable external path computing for MPLS by including the `lsp-external-controller` statement at the `[edit protocols mpls]` hierarchy level.

   This configuration is required for PCEP with RSVP-TE extensions as well. You cannot disable PCEP with RSVP-TE when segment routing for PCEP is enabled.

2. Enable external path computing for SR-TE by including the `lsp-external-controller` *`pccd`* statement at the `[edit protocols spring-traffic-engineering]` hierarchy level.

3. Enable segment routing for the PCE by including the `spring-capability` statement at the `[edit protocols pcep pce` *`pce-name`*`]` hierarchy level.

4. Optionally, configure the maximum SID depth for the PCE by including the `max-sid-depth` *`number`* statement at the `[edit protocols pcep pce` *`pce-name`*`]` hierarchy level.

   The maximum SID depth is the number of SIDs supported by a node or a link on a node. When not configured, a default maximum SID value of 5 is applied.

5. Optionally, configure the preference value for segment routing by including the `preference` *`preference-value`* at the `[edit protocol spring-te]` hierarchy level.

   The preference value indicates the order in which a path is selected as the active path form among candidate paths, where a higher value has a higher preference. When not configured, a default preference value of 8 is applied.

6. Optionally, configure segment routing logging for troubleshooting purpose by including the `traceoptions` statement at the `[edit protocols spring-te]` hierarchy level.

For PCE-delegation of segment routing LSPs, in addition to the aforementioned steps, do the following:

1. Define a segment list with label parameters. This creates a segment routing LSP locally on the PCC.

2. Enable delegation capability of the locally configured LSP on the PCC by including the `lsp-external-controller` *`pccd`* statement at any of the following hierarchies depending on the segment routing LSP source:

   - For statically configured source-routing paths that are computed with distributed CSPF—`[edit protocols source-packet-routing source-routing-path` *`lsp-name`* `primary` *`path-name`* `compute` *`profile-name`*`]` and `[edit protocols source-packet-routing source-routing-path` *`lsp-name`* `primary` *`path-name`*`]` hierarchy levels.

   - For statically configured source-routing paths that have the entire label stack statically configured and source-routing paths that are automatically translated—`[edit protocols source-packet-routing source-routing-path` *`lsp-name`* `primary` *`path-name`*`]` hierarchy level.

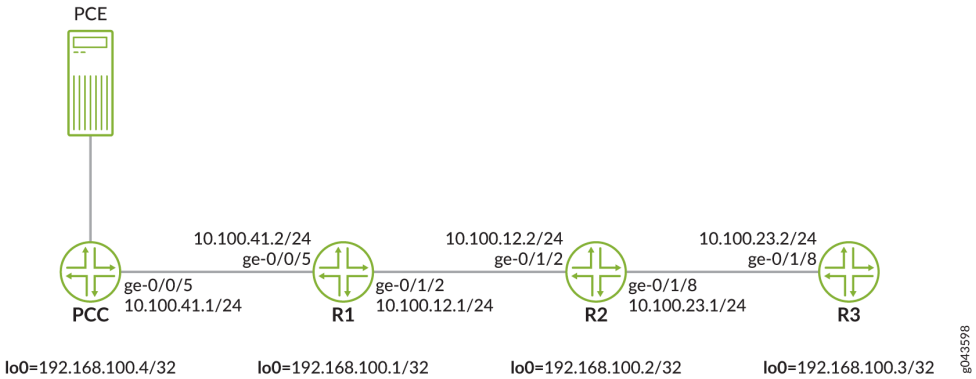
   - For dynamically created tunnels triggered through the Dynamic Tunnel Module that have last-hop ERO resolution—`[edit protocols source-packet-routing source-routing-path-template` *`template-name`* `primary` *`primary-segment-list-name`*`]` and `[edit protocols source-packet-routing source-routing-path-template` *`template-name`*`]` hierarchy levels.

Topology

Figure 155 on page 2203 illustrates a sample network topology that has a PCEP session running between the PCE and the PCC (the ingress MX Series router). Routers R1, R2, and R3 are the other MX

Series routers in the network. In this example, we configure segment routing for PCEP on the PCC. We also configure a static route on the PCC to Router R3 to verify the use of SR-TE tunnel routes when routing traffic for the static route.

Figure 155: Segment Routing for PCEP



Configuration

IN THIS SECTION

- CLI Quick Configuration | **2203**
- Procedure | **2206**
- Results | **2210**

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

Although we present the configuration of all the devices (PCC and the three routers) in this section, the Step-by-Step procedure documents only the configuration of the PCC.

PCC

```
set interfaces ge-0/0/5 unit 0 family inet address 10.100.41.1/24
set interfaces ge-0/0/5 unit 0 family iso
set interfaces ge-0/0/5 unit 0 family mpls
```

```
set interfaces lo0 unit 0 family inet address 192.168.100.4/32 primary
set interfaces lo0 unit 0 family iso address 49.0011.0110.0000.0101.00
set interfaces lo0 unit 0 family mpls
set routing-options static route 100.1.1.1/32 next-hop 192.168.100.3
set routing-options router-id 192.168.100.4
set routing-options autonomous-system 64496
set protocols rsvp interface fxp0.0 disable
set protocols rsvp interface all
set protocols mpls lsp-external-controller pccd
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols isis source-packet-routing srgb start-label 800000
set protocols isis source-packet-routing srgb index-range 4000
set protocols isis source-packet-routing node-segment ipv4-index 101
set protocols isis source-packet-routing node-segment ipv6-index 11
set protocols isis level 1 disable
set protocols isis level 2 wide-metrics-only
set protocols isis interface all point-to-point
set protocols isis interface all level 2 metric 10
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0 passive
set protocols source-packet-routing segment-list static_seg_list_1 hop1 label 800102
set protocols source-packet-routing segment-list static_seg_list_1 hop2 label 800103
set protocols source-packet-routing source-routing-path static_srte_lsp_1 to 192.168.100.3
set protocols source-packet-routing source-routing-path static_srte_lsp_1 primary
static_seg_list_1 lsp-external-controller pccd
set protocols spring-traffic-engineering lsp-external-controller pccd
set protocols source-packet-routing source-routing-path static1 lsp-external-controller pccd
set protocols pcep pce pce1 local-address 192.168.100.4
set protocols pcep pce pce1 destination-ipv4-address 10.102.180.232
set protocols pcep pce pce1 destination-port 4189
set protocols pcep pce pce1 pce-type active
set protocols pcep pce pce1 pce-type stateful
set protocols pcep pce pce1 lsp-provisioning
set protocols pcep pce pce1 spring-capability
```

Router R1

```
set interfaces ge-0/0/5 unit 0 family inet address 10.100.41.2/24
set interfaces ge-0/0/5 unit 0 family iso
set interfaces ge-0/0/5 unit 0 family mpls
set interfaces ge-0/1/2 unit 0 family inet address 10.100.12.1/24
```

```
set interfaces ge-0/1/2 unit 0 family iso
set interfaces ge-0/1/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.100.1/32 primary
set interfaces lo0 unit 0 family iso address 49.0011.0110.0000.0102.00
set interfaces lo0 unit 0 family mpls
set routing-options router-id 192.168.100.1
set routing-options autonomous-system 64496
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols isis source-packet-routing srgb start-label 800000
set protocols isis source-packet-routing srgb index-range 4000
set protocols isis source-packet-routing node-segment ipv4-index 102
set protocols isis level 1 disable
set protocols isis level 2 wide-metrics-only
set protocols isis interface all point-to-point
set protocols isis interface all level 2 metric 10
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0 passive
```

Router R2

```
set interfaces ge-0/1/2 unit 0 family inet address 10.100.12.2/24
set interfaces ge-0/1/2 unit 0 family iso
set interfaces ge-0/1/2 unit 0 family mpls
set interfaces ge-0/1/8 unit 0 family inet address 10.100.23.1/24
set interfaces ge-0/1/8 unit 0 family iso
set interfaces ge-0/1/8 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.100.2/32
set interfaces lo0 unit 0 family iso address 49.0011.0110.0000.0105.00
set interfaces lo0 unit 0 family mpls
set routing-options router-id 192.168.100.2
set routing-options autonomous-system 64496
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols isis source-packet-routing srgb start-label 800000
set protocols isis source-packet-routing srgb index-range 4000
set protocols isis source-packet-routing node-segment ipv4-index 105
set protocols isis level 1 disable
```

```
set protocols isis level 2 wide-metrics-only
set protocols isis interface all point-to-point
set protocols isis interface all level 2 metric 10
set protocols isis interface all level 1 disable
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0 passive
```

Router R3

```
set interfaces ge-0/1/8 unit 0 family inet address 10.100.23.2/24
set interfaces ge-0/1/8 unit 0 family iso
set interfaces ge-0/1/8 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.100.3/32 primary
set interfaces lo0 unit 0 family iso address 49.0011.0110.0000.0103.00
set interfaces lo0 unit 0 family mpls
set routing-options static route 100.1.1.1/32 receive
set routing-options router-id 192.168.100.3
set routing-options autonomous-system 64496
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols isis source-packet-routing srgb start-label 800000
set protocols isis source-packet-routing srgb index-range 4000
set protocols isis source-packet-routing node-segment ipv4-index 103
set protocols isis level 1 disable
set protocols isis level 2 wide-metrics-only
set protocols isis interface all point-to-point
set protocols isis interface all level 2 metric 10
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0 passive
```

Procedure

Step-by-Step Procedure

In this example, we configure only the PCC.

The following steps require that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the PCC:

1. Configure the interfaces of the PCC.

```
[edit interfaces]
user@PCC# set ge-0/0/5 unit 0 family inet address 10.100.41.1/24
user@PCC# set ge-0/0/5 unit 0 family iso
user@PCC# set ge-0/0/5 unit 0 family mpls
user@PCC# set lo0 unit 0 family inet address 192.168.100.4/32 primary
user@PCC# set lo0 unit 0 family iso address 49.0011.0110.0000.0101.00
user@PCC# set lo0 unit 0 family mpls
```

2. Configure the router ID and assign an autonomous system number for the PCC.

```
[edit routing-options]
user@PCC# set router-id 192.168.100.4
user@PCC# set autonomous-system 64496
```

3. Configure a static route from the PCC to Router R3.

   The static route is created for verification purpose only and does not affect the feature functionality.

```
[edit routing-options]
user@PCC# set static route 100.1.1.1/32 next-hop 192.168.100.3
```

4. Configure RSVP on all the interfaces of the PCC, excluding the management interface.

```
[edit protocols]
user@PCC# set rsvp interface fxp0.0 disable
user@PCC# set rsvp interface all
```

5. Configure MPLS on all the interfaces of the PCC, excluding the management interface.

```
[edit protocols]
user@PCC# set mpls interface all
user@PCC# set mpls interface fxp0.0 disable
```

6. Enable external path computing capability for MPLS.

```
[edit protocols]
user@PCC# set mpls lsp-external-controller pccd
```

7. Configure IS-IS level 2 on all the interfaces of the PCC, excluding the management and loopback interfaces.

```
[edit protocols]
user@PCC# set isis level 1 disable
user@PCC# set isis level 2 wide-metrics-only
user@PCC# set isis interface all point-to-point
user@PCC# set isis interface all level 2 metric 10
user@PCC# set isis interface fxp0.0 disable
user@PCC# set isis interface lo0.0 passive
```

8. Configure segment routing global block (SRGB) attributes for segment routing.

```
[edit protocols]
user@PCC# set isis source-packet-routing srgb start-label 800000
user@PCC# set isis source-packet-routing srgb index-range 4000
user@PCC# set isis source-packet-routing node-segment ipv4-index 101
user@PCC# set isis source-packet-routing node-segment ipv6-index 11
```

9. Enable external path computing capability for SR-TE.

```
[edit protocols]
user@PCC# set spring-traffic-engineering lsp-external-controller pccd
```

10. Configure the PCE parameters and enable provisioning of the LSP by the PCE and the segment routing capability.

```
[edit protocols]
user@PCC# set pcep pce pce1 local-address 192.168.100.4
user@PCC# set pcep pce pce1 destination-ipv4-address 10.102.180.232
user@PCC# set pcep pce pce1 destination-port 4189
```

```
user@PCC# set pcep pce pce1 pce-type active
user@PCC# set pcep pce pce1 pce-type stateful
```

11. Enable provisioning of segment routing LSPs by the PCE.

```
[edit protocols]
user@PCC# set pcep pce pce1 lsp-provisioning
```

12. Enable segment routing capability for the PCE.

```
[edit protocols]
user@PCC# set pcep pce pce1 spring-capability
```

13. Define the static segment list `static_seg_list_1` parameters.

```
[edit protocols]
user@PCC# set source-packet-routing segment-list static_seg_list_1 hop1 label 800102
user@PCC# set source-packet-routing segment-list static_seg_list_1 hop2 label 800103
```

14. Configure a static segment routing LSP from the PCC to Router R3 for PCE delegation.

```
[edit protocols]
user@PCC# set source-packet-routing source-routing-path static_srte_lsp_1 to 192.168.100.3
```

15. Enable delegation capability for the `static_srte_lsp_1` source-routing path.

```
[edit protocols]
user@PCC# set source-packet-routing source-routing-path static_srte_lsp_1 primary
static_seg_list_1 lsp-external-controller pccd
```

By completing Steps 13, 14, and 15, you enable the PCC to delegate the segment routing LSPs to the PCE.

16. Commit the configuration.

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@PCC# show interfaces
ge-0/0/5 {
    unit 0 {
        family inet {
            address 10.100.41.1/24;
        }
        family iso;
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 192.168.100.4/32 {
                primary;
            }
        }
        family iso {
            address 49.0011.0110.0000.0101.00;
        }
        family mpls;
    }
}
```

```
user@PCC# show routing-options
static {
    route 100.1.1.1/32 next-hop 192.168.100.3;
}
router-id 192.168.100.4;
autonomous-system 64496;
```

```
user@PCC# show protocols
rsvp {
```

```
    interface fxp0.0 {
        disable;
    }
    interface all;
}
mpls {
    lsp-external-controller pccd;
    interface all;
    interface fxp0.0 {
        disable;
    }
}
isis {
    source-packet-routing {
        srgb start-label 800000 index-range 4000;
        node-segment {
            ipv4-index 101;
            ipv6-index 11;
        }
    }
    level 1 disable;
    level 2 wide-metrics-only;
    interface all {
        point-to-point;
        level 2 metric 10;
    }
    interface fxp0.0 {
        disable;
    }
    interface lo0.0 {
        passive;
    }
}
spring-traffic-engineering {
    lsp-external-controller pccd;
}
source-packet-routing {
    segment-list static_seg_list_1 {
        hop1 label 800102
        hop1 label 800102
    }
    source-routing-path static_srte_lsp_1 {
        to 192.168.100.3;
```

```
        primary {
            static_seg_list_1 {
                lsp-external-controller pccd;
            }
        }
    }
}
pcep {
    pce pce1 {
        local-address 192.168.100.4;
        destination-ipv4-address 10.102.180.232;
        destination-port 4189;
        pce-type active stateful;
        lsp-provisioning;
        spring-capability;
    }
}
```

If you are done configuring the device (the PCC), enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verify IS-IS Adjacency and Labels | **2212**
- Verify the Traffic Engineering Database | **2222**
- Verify SR-TE LSPs | **2226**
- Verify Tunnel Route Creation | **2229**
- Verify Forwarding Table Entries | **2231**
- Verify the Use of Tunnel Routes for Static Route Forwarding | **2233**

Confirm that the configuration is working properly.

Verify IS-IS Adjacency and Labels

Purpose

Verify the IS-IS adjacency on the PCC. Take note of the SRGB label range, adjacency and node segment values, and SPRING capability output fields.

Action

From operational mode, run the `show isis adjacency extensive`, `show isis database extensive`, and `show isis overview` commands.

```
user@PCC> show isis adjacency extensive
R1
  Interface: ge-0/0/5.0, Level: 2, State: Up, Expires in 25 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 00:37:15 ago
  Circuit type: 2, Speaks: IP, IPv6
  Topologies: Unicast
  Restart capable: Yes, Adjacency advertisement: Advertise
  IP addresses: 10.100.41.2
  Level 2 IPv4 Adj-SID: 16
  Transition log:
  When                  State        Event           Down reason
  Wed Apr  5 02:42:48   Up           Seenself

PCE
  Interface: gre.0, Level: 2, State: Up, Expires in 25 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 00:27:00 ago
  Circuit type: 2, Speaks: IP, IPv6
  Topologies: Unicast
  Restart capable: Yes, Adjacency advertisement: Advertise
  IP addresses: 11.105.199.2
  Level 2
  Transition log:
  When                  State        Event           Down reason
  Wed Apr  5 02:53:03   Up           Seenself
```

```
user@PCC> show isis database extensive
IS-IS level 1 link-state database:

IS-IS level 2 link-state database:

PCC.00-00 Sequence: 0x2a6, Checksum: 0x1a4f, Lifetime: 1150 secs
  IPV4 Index: 101
  Node Segment Blocks Advertised:
    Start Index : 0, Size : 4000, Label-Range: [ 800000, 803999 ]
   IS neighbor: R1.00                          Metric:       10
     Two-way fragment: R1.00-00, Two-way first fragment: R1.00-00
```

```
 IS neighbor: PCE.00                        Metric: 16777215
 IP prefix: 192.168.100.4/32                   Metric:        0 Internal Up
 IP prefix: 11.101.102.0/30                 Metric:       10 Internal Up
 IP prefix: 11.105.199.0/30                 Metric: 16777215 Internal Up

Header: LSP ID: PCC.00-00, Length: 243 bytes
  Allocated length: 1492 bytes, Router ID: 192.168.100.4
  Remaining lifetime: 1150 secs, Level: 2, Interface: 0
  Estimated free bytes: 1084, Actual free bytes: 1249
  Aging timer expires in: 1150 secs
  Protocols: IP, IPv6

Packet: LSP ID: PCC.00-00, Length: 243 bytes, Lifetime : 1198 secs
  Checksum: 0x1a4f, Sequence: 0x2a6, Attributes: 0x3 L1 L2
  NLPID: 0x83, Fixed length: 27 bytes, Version: 1, Sysid length: 0 bytes
  Packet type: 20, Packet version: 1, Max area: 0

TLVs:
  Area address: 49.0011 (3)
  LSP Buffer Size: 1492
  Speaks: IP
  Speaks: IPV6
  IP router id: 192.168.100.4
  IP address: 192.168.100.4
  Hostname: PCC
  IS extended neighbor: R1.00, Metric: default 10
    IP address: 10.100.41.1
    Neighbor's IP address: 10.100.41.2
    Local interface index: 334, Remote interface index: 333
    Current reservable bandwidth:
      Priority 0 : 10Mbps
      Priority 1 : 10Mbps
      Priority 2 : 10Mbps
      Priority 3 : 10Mbps
      Priority 4 : 10Mbps
      Priority 5 : 10Mbps
      Priority 6 : 10Mbps
      Priority 7 : 10Mbps
    Maximum reservable bandwidth: 10Mbps
    Maximum bandwidth: 10Mbps
    Administrative groups:  0 none
    P2P IPV4 Adj-SID - Flags:0x30(F:0,B:0,V:1,L:1,S:0), Weight:0, Label: 16
  IS extended neighbor: PCE.00, Metric: default 16777215
```

```
      IP address: 11.105.199.1
      Neighbor's IP address: 11.105.199.2
      Local interface index: 329, Remote interface index: 329
    IP extended prefix: 11.101.102.0/30 metric 10 up
    IP extended prefix: 11.105.199.0/30 metric 16777215 up
    IP extended prefix: 192.168.100.4/32 metric 0 up
      8 bytes of subtlvs
      Node SID, Flags: 0x40(R:0,N:1,P:0,E:0,V:0,L:0), Algo: SPF(0), Value: 101
    Router Capability:  Router ID 192.168.100.4, Flags: 0x00
      SPRING Capability - Flags: 0xc0(I:1,V:1), Range: 4000, SID-Label: 800000
      SPRING Algorithm - Algo: 0
  No queued transmissions

R1.00-00 Sequence: 0x297, Checksum: 0x1615, Lifetime: 839 secs
  IPV4 Index: 102
  Node Segment Blocks Advertised:
    Start Index : 0, Size : 4000, Label-Range: [ 800000, 803999 ]
   IS neighbor: PCC.00                         Metric:       10
     Two-way fragment: PCC.00-00, Two-way first fragment: PCC.00-00
   IS neighbor: R2.00                         Metric:       10
     Two-way fragment: R2.00-00, Two-way first fragment: R2.00-00
   IP prefix: 192.168.100.1/32                  Metric:        0 Internal Up
   IP prefix: 11.101.102.0/30                 Metric:       10 Internal Up
   IP prefix: 11.102.105.0/30                 Metric:       10 Internal Up

  Header: LSP ID: R1.00-00, Length: 302 bytes
    Allocated length: 302 bytes, Router ID: 192.168.100.1
    Remaining lifetime: 839 secs, Level: 2, Interface: 334
    Estimated free bytes: 0, Actual free bytes: 0
    Aging timer expires in: 839 secs
    Protocols: IP, IPv6

  Packet: LSP ID: R1.00-00, Length: 302 bytes, Lifetime : 1196 secs
    Checksum: 0x1615, Sequence: 0x297, Attributes: 0x3 L1 L2
    NLPID: 0x83, Fixed length: 27 bytes, Version: 1, Sysid length: 0 bytes
    Packet type: 20, Packet version: 1, Max area: 0

  TLVs:
    Area address: 49.0011 (3)
    LSP Buffer Size: 1492
    Speaks: IP
    Speaks: IPV6
    IP router id: 192.168.100.1
```

```
    IP address: 192.168.100.1
    Hostname: R1
    IP extended prefix: 192.168.100.1/32 metric 0 up
      8 bytes of subtlvs
      Node SID, Flags: 0x40(R:0,N:1,P:0,E:0,V:0,L:0), Algo: SPF(0), Value: 102
    IP extended prefix: 11.101.102.0/30 metric 10 up
    IP extended prefix: 11.102.105.0/30 metric 10 up
    Router Capability:  Router ID 192.168.100.1, Flags: 0x00
      SPRING Capability - Flags: 0xc0(I:1,V:1), Range: 4000, SID-Label: 800000
      SPRING Algorithm - Algo: 0
    IS extended neighbor: R2.00, Metric: default 10
      IP address: 10.100.12.1
      Neighbor's IP address: 10.100.12.2
      Local interface index: 334, Remote interface index: 333
      Current reservable bandwidth:
        Priority 0 : 10Mbps
        Priority 1 : 10Mbps
        Priority 2 : 10Mbps
        Priority 3 : 10Mbps
        Priority 4 : 10Mbps
        Priority 5 : 10Mbps
        Priority 6 : 10Mbps
        Priority 7 : 10Mbps
      Maximum reservable bandwidth: 10Mbps
      Maximum bandwidth: 10Mbps
      Administrative groups:  0 none
      P2P IPV4 Adj-SID - Flags:0x30(F:0,B:0,V:1,L:1,S:0), Weight:0, Label: 17
    IS extended neighbor: PCC.00, Metric: default 10
      IP address: 10.100.41.2
      Neighbor's IP address: 10.100.41.1
      Local interface index: 333, Remote interface index: 334
      Current reservable bandwidth:
        Priority 0 : 10Mbps
        Priority 1 : 10Mbps
        Priority 2 : 10Mbps
        Priority 3 : 10Mbps
        Priority 4 : 10Mbps
        Priority 5 : 10Mbps
        Priority 6 : 10Mbps
        Priority 7 : 10Mbps
      Maximum reservable bandwidth: 10Mbps
      Maximum bandwidth: 10Mbps
      Administrative groups:  0 none
```

```
      P2P IPV4 Adj-SID - Flags:0x30(F:0,B:0,V:1,L:1,S:0), Weight:0, Label: 16
  No queued transmissions

R3.00-00 Sequence: 0x95, Checksum: 0xd459, Lifetime: 895 secs
  IPV4 Index: 103
  Node Segment Blocks Advertised:
    Start Index : 0, Size : 4000, Label-Range: [ 800000, 803999 ]
   IS neighbor: R2.00                         Metric:       10
     Two-way fragment: R2.00-00, Two-way first fragment: R2.00-00
   IP prefix: 192.168.100.3/32                  Metric:        0 Internal Up
   IP prefix: 11.102.1.0/24                   Metric:       10 Internal Up
   IP prefix: 11.103.107.0/30                 Metric:       10 Internal Up

  Header: LSP ID: R3.00-00, Length: 209 bytes
    Allocated length: 284 bytes, Router ID: 192.168.100.3
    Remaining lifetime: 895 secs, Level: 2, Interface: 334
    Estimated free bytes: 75, Actual free bytes: 75
    Aging timer expires in: 895 secs
    Protocols: IP, IPv6

  Packet: LSP ID: R3.00-00, Length: 209 bytes, Lifetime : 1192 secs
    Checksum: 0xd459, Sequence: 0x95, Attributes: 0x3 L1 L2
    NLPID: 0x83, Fixed length: 27 bytes, Version: 1, Sysid length: 0 bytes
    Packet type: 20, Packet version: 1, Max area: 0

  TLVs:
    Area address: 49.0011 (3)
    LSP Buffer Size: 1492
    Speaks: IP
    Speaks: IPV6
    IP router id: 192.168.100.3
    IP address: 192.168.100.3
    Hostname: R3
    IS extended neighbor: R2.00, Metric: default 10
      IP address: 10.100.23.2
      Neighbor's IP address: 10.100.23.1
      Local interface index: 336, Remote interface index: 334
      Current reservable bandwidth:
        Priority 0 : 10Mbps
        Priority 1 : 10Mbps
        Priority 2 : 10Mbps
        Priority 3 : 10Mbps
        Priority 4 : 10Mbps
```

```
        Priority 5 : 10Mbps
        Priority 6 : 10Mbps
        Priority 7 : 10Mbps
      Maximum reservable bandwidth: 10Mbps
      Maximum bandwidth: 10Mbps
      Administrative groups:  0 none
      P2P IPV4 Adj-SID - Flags:0x30(F:0,B:0,V:1,L:1,S:0), Weight:0, Label: 16
    IP extended prefix: 192.168.100.3/32 metric 0 up
      8 bytes of subtlvs
      Node SID, Flags: 0x40(R:0,N:1,P:0,E:0,V:0,L:0), Algo: SPF(0), Value: 103
    IP extended prefix: 11.103.107.0/30 metric 10 up
    IP extended prefix: 11.102.1.0/24 metric 10 up
    Router Capability:  Router ID 192.168.100.3, Flags: 0x00
      SPRING Capability - Flags: 0xc0(I:1,V:1), Range: 4000, SID-Label: 800000
      SPRING Algorithm - Algo: 0
  No queued transmissions

R2.00-00 Sequence: 0x2aa, Checksum: 0xf8f4, Lifetime: 1067 secs
  IPV4 Index: 105
  Node Segment Blocks Advertised:
    Start Index : 0, Size : 4000, Label-Range: [ 800000, 803999 ]
   IS neighbor: R1.00                         Metric:       10
     Two-way fragment: R1.00-00, Two-way first fragment: R1.00-00
   IS neighbor: R3.00                         Metric:       10
     Two-way fragment: R3.00-00, Two-way first fragment: R3.00-00
   IP prefix: 192.168.100.2/32                  Metric:        0 Internal Up
   IP prefix: 11.102.105.0/30                 Metric:       10 Internal Up
   IP prefix: 11.103.107.0/30                 Metric:       10 Internal Up

  Header: LSP ID: R2.00-00, Length: 302 bytes
    Allocated length: 302 bytes, Router ID: 192.168.100.2
    Remaining lifetime: 1067 secs, Level: 2, Interface: 334
    Estimated free bytes: 0, Actual free bytes: 0
    Aging timer expires in: 1067 secs
    Protocols: IP, IPv6

  Packet: LSP ID: R2.00-00, Length: 302 bytes, Lifetime : 1194 secs
    Checksum: 0xf8f4, Sequence: 0x2aa, Attributes: 0x3 L1 L2
    NLPID: 0x83, Fixed length: 27 bytes, Version: 1, Sysid length: 0 bytes
    Packet type: 20, Packet version: 1, Max area: 0

  TLVs:
    Area address: 49.0011 (3)
```

```
    LSP Buffer Size: 1492
    Speaks: IP
    Speaks: IPV6
    IP router id: 192.168.100.2
    IP address: 192.168.100.2
    Hostname: R2
    IP extended prefix: 192.168.100.2/32 metric 0 up
      8 bytes of subtlvs
      Node SID, Flags: 0x40(R:0,N:1,P:0,E:0,V:0,L:0), Algo: SPF(0), Value: 105
    IP extended prefix: 11.102.105.0/30 metric 10 up
    IP extended prefix: 11.103.107.0/30 metric 10 up
    Router Capability:  Router ID 192.168.100.2, Flags: 0x00
      SPRING Capability - Flags: 0xc0(I:1,V:1), Range: 4000, SID-Label: 800000
      SPRING Algorithm - Algo: 0
    IS extended neighbor: R3.00, Metric: default 10
      IP address: 10.100.23.1
      Neighbor's IP address: 10.100.23.2
      Local interface index: 334, Remote interface index: 336
      Current reservable bandwidth:
        Priority 0 : 10Mbps
        Priority 1 : 10Mbps
        Priority 2 : 10Mbps
        Priority 3 : 10Mbps
        Priority 4 : 10Mbps
        Priority 5 : 10Mbps
        Priority 6 : 10Mbps
        Priority 7 : 10Mbps
      Maximum reservable bandwidth: 10Mbps
      Maximum bandwidth: 10Mbps
      Administrative groups:  0 none
      P2P IPV4 Adj-SID - Flags:0x30(F:0,B:0,V:1,L:1,S:0), Weight:0, Label: 16
    IS extended neighbor: R1.00, Metric: default 10
      IP address: 10.100.12.2
      Neighbor's IP address: 10.100.12.1
      Local interface index: 333, Remote interface index: 334
      Current reservable bandwidth:
        Priority 0 : 10Mbps
        Priority 1 : 10Mbps
        Priority 2 : 10Mbps
        Priority 3 : 10Mbps
        Priority 4 : 10Mbps
        Priority 5 : 10Mbps
        Priority 6 : 10Mbps
```

```
        Priority 7 : 10Mbps
      Maximum reservable bandwidth: 10Mbps
      Maximum bandwidth: 10Mbps
      Administrative groups:  0 none
      P2P IPV4 Adj-SID - Flags:0x30(F:0,B:0,V:1,L:1,S:0), Weight:0, Label: 17
  No queued transmissions

PCE.00-00 Sequence: 0x277, Checksum: 0x64a5, Lifetime: 533 secs
   IS neighbor: PCC.00                         Metric: 16777215
   IP prefix: 11.0.0.199/32                   Metric:        0 Internal Up
   IP prefix: 11.105.199.0/30                 Metric: 16777215 Internal Up

  Header: LSP ID: PCE.00-00, Length: 120 bytes
    Allocated length: 284 bytes, Router ID: 11.0.0.199
    Remaining lifetime: 533 secs, Level: 2, Interface: 329
    Estimated free bytes: 164, Actual free bytes: 164
    Aging timer expires in: 533 secs
    Protocols: IP, IPv6

  Packet: LSP ID: PCE.00-00, Length: 120 bytes, Lifetime : 1196 secs
    Checksum: 0x64a5, Sequence: 0x277, Attributes: 0x3 L1 L2
    NLPID: 0x83, Fixed length: 27 bytes, Version: 1, Sysid length: 0 bytes
    Packet type: 20, Packet version: 1, Max area: 0

  TLVs:
    Area address: 11.0007 (3)
    LSP Buffer Size: 1492
    Speaks: IP
    Speaks: IPV6
    IP router id: 11.0.0.199
    IP address: 11.0.0.199
    Hostname: PCE
    Router Capability:  Router ID 11.0.0.199, Flags: 0x00
    IP extended prefix: 11.105.199.0/30 metric 16777215 up
    IP extended prefix: 11.0.0.199/32 metric 0 up
    IS extended neighbor: PCC.00, Metric: default 16777215
      IP address: 11.105.199.2
      Neighbor's IP address: 11.105.199.1
```

```
      Local interface index: 329, Remote interface index: 329
  No queued transmissions
```

```
user@PCC> show isis overview
Instance: master
  Router ID: 192.168.100.4
  Hostname: PCC
  Sysid: 0110.0000.0101
  Areaid: 49.0011
  Adjacency holddown: enabled
  Maximum Areas: 3
  LSP life time: 1200
  Attached bit evaluation: enabled
  SPF delay: 200 msec, SPF holddown: 5000 msec, SPF rapid runs: 3
  IPv4 is enabled, IPv6 is enabled, SPRING based MPLS is enabled
  Traffic engineering: enabled
  Restart: Disabled
    Helper mode: Enabled
  Layer2-map: Disabled
  Source Packet Routing (SPRING): Enabled
    SRGB Config Range:
      SRGB Start-Label : 800000, SRGB Index-Range : 4000
    SRGB Block Allocation: Success
      SRGB Start Index : 800000, SRGB Size : 4000, Label-Range: [ 800000, 803999 ]
    Node Segments: Enabled
      Ipv4 Index : 101, Ipv6 Index : 11
  Level 1
    Internal route preference: 15
    External route preference: 160
    Prefix export count: 0
    Wide metrics are enabled, Narrow metrics are enabled
    Source Packet Routing is enabled
  Level 2
    Internal route preference: 18
    External route preference: 165
    Prefix export count: 0
    Wide metrics are enabled
    Source Packet Routing is enabled
```

Meaning

The IS-IS adjacency between the PCC and PCE and that between the PCC and Router R1 are up and operational. The output also displays the label assignments for the adjacent and node segments.

Verify the Traffic Engineering Database

Purpose

Verify the traffic engineering database entries on the PCC.

Action

From operational mode, run the `show ted database extensive` command.

```
user@PCC# show ted database extensive
TED database: 5 ISIS nodes 5 INET nodes
NodeID: PCC.00(192.168.100.4)
  Type: Rtr, Age: 403 secs, LinkIn: 1, LinkOut: 1
  Protocol: IS-IS(2)
   192.168.100.4
    To: R1.00(192.168.100.1), Local: 10.100.41.1, Remote: 10.100.41.2
      Local interface index: 334, Remote interface index: 333
      Color: 0 none
      Metric: 10
      IGP metric: 10
      Static BW: 10Mbps
      Reservable BW: 10Mbps
      Available BW [priority] bps:
          [0] 10Mbps       [1] 10Mbps      [2] 10Mbps      [3] 10Mbps
          [4] 10Mbps       [5] 10Mbps      [6] 10Mbps      [7] 10Mbps
      Interface Switching Capability Descriptor(1):
        Switching type: Packet
        Encoding type: Packet
        Maximum LSP BW [priority] bps:
          [0] 10Mbps       [1] 10Mbps      [2] 10Mbps      [3] 10Mbps
          [4] 10Mbps       [5] 10Mbps      [6] 10Mbps      [7] 10Mbps
      P2P Adjacency-SID:
        IPV4, SID: 16, Flags: 0x30, Weight: 0
   Prefixes:
     192.168.100.4/32
      Metric: 0, Flags: 0x00
```

```
      Prefix-SID:
        SID: 101, Flags: 0x40, Algo: 0
   SPRING-Capabilities:
     SRGB block [Start: 800000, Range: 4000, Flags: 0xc0]
   SPRING-Algorithms:
     Algo: 0
NodeID: R1.00(192.168.100.1)
  Type: Rtr, Age: 712 secs, LinkIn: 2, LinkOut: 2
  Protocol: IS-IS(2)
   192.168.100.1
    To: PCC.00(192.168.100.4), Local: 10.100.41.2, Remote: 10.100.41.1
      Local interface index: 333, Remote interface index: 334
      Color: 0 none
      Metric: 10
      IGP metric: 10
      Static BW: 10Mbps
      Reservable BW: 10Mbps
      Available BW [priority] bps:
          [0] 10Mbps       [1] 10Mbps      [2] 10Mbps      [3] 10Mbps
          [4] 10Mbps       [5] 10Mbps      [6] 10Mbps      [7] 10Mbps
      Interface Switching Capability Descriptor(1):
        Switching type: Packet
        Encoding type: Packet
        Maximum LSP BW [priority] bps:
          [0] 10Mbps       [1] 10Mbps      [2] 10Mbps      [3] 10Mbps
          [4] 10Mbps       [5] 10Mbps      [6] 10Mbps      [7] 10Mbps
      P2P Adjacency-SID:
        IPV4, SID: 16, Flags: 0x30, Weight: 0
    To: R2.00(192.168.100.2), Local: 10.100.12.1, Remote: 10.100.12.2
      Local interface index: 334, Remote interface index: 333
      Color: 0 none
      Metric: 10
      IGP metric: 10
      Static BW: 10Mbps
      Reservable BW: 10Mbps
      Available BW [priority] bps:
          [0] 10Mbps       [1] 10Mbps      [2] 10Mbps      [3] 10Mbps
          [4] 10Mbps       [5] 10Mbps      [6] 10Mbps      [7] 10Mbps
      Interface Switching Capability Descriptor(1):
        Switching type: Packet
        Encoding type: Packet
        Maximum LSP BW [priority] bps:
          [0] 10Mbps       [1] 10Mbps      [2] 10Mbps      [3] 10Mbps
```

```
          [4] 10Mbps       [5] 10Mbps      [6] 10Mbps      [7] 10Mbps
      P2P Adjacency-SID:
        IPV4, SID: 17, Flags: 0x30, Weight: 0
   Prefixes:
     192.168.100.1/32
      Metric: 0, Flags: 0x00
      Prefix-SID:
        SID: 102, Flags: 0x40, Algo: 0
   SPRING-Capabilities:
     SRGB block [Start: 800000, Range: 4000, Flags: 0xc0]
   SPRING-Algorithms:
     Algo: 0
NodeID: R3.00(192.168.100.3)
  Type: Rtr, Age: 435 secs, LinkIn: 1, LinkOut: 1
  Protocol: IS-IS(2)
   192.168.100.3
    To: R2.00(192.168.100.2), Local: 10.100.23.2, Remote: 10.100.23.1
      Local interface index: 336, Remote interface index: 334
      Color: 0 none
      Metric: 10
      IGP metric: 10
      Static BW: 10Mbps
      Reservable BW: 10Mbps
      Available BW [priority] bps:
          [0] 10Mbps       [1] 10Mbps      [2] 10Mbps      [3] 10Mbps
          [4] 10Mbps       [5] 10Mbps      [6] 10Mbps      [7] 10Mbps
      Interface Switching Capability Descriptor(1):
        Switching type: Packet
        Encoding type: Packet
        Maximum LSP BW [priority] bps:
          [0] 10Mbps       [1] 10Mbps      [2] 10Mbps      [3] 10Mbps
          [4] 10Mbps       [5] 10Mbps      [6] 10Mbps      [7] 10Mbps
      P2P Adjacency-SID:
        IPV4, SID: 16, Flags: 0x30, Weight: 0
   Prefixes:
     192.168.100.3/32
      Metric: 0, Flags: 0x00
      Prefix-SID:
        SID: 103, Flags: 0x40, Algo: 0
   SPRING-Capabilities:
     SRGB block [Start: 800000, Range: 4000, Flags: 0xc0]
   SPRING-Algorithms:
     Algo: 0
```

```
NodeID: R2.00(192.168.100.2)
  Type: Rtr, Age: 456 secs, LinkIn: 2, LinkOut: 2
  Protocol: IS-IS(2)
   192.168.100.2
    To: R1.00(192.168.100.1), Local: 10.100.12.2, Remote: 10.100.12.1
      Local interface index: 333, Remote interface index: 334
      Color: 0 none
      Metric: 10
      IGP metric: 10
      Static BW: 10Mbps
      Reservable BW: 10Mbps
      Available BW [priority] bps:
          [0] 10Mbps       [1] 10Mbps      [2] 10Mbps      [3] 10Mbps
          [4] 10Mbps       [5] 10Mbps      [6] 10Mbps      [7] 10Mbps
      Interface Switching Capability Descriptor(1):
        Switching type: Packet
        Encoding type: Packet
        Maximum LSP BW [priority] bps:
          [0] 10Mbps       [1] 10Mbps      [2] 10Mbps      [3] 10Mbps
          [4] 10Mbps       [5] 10Mbps      [6] 10Mbps      [7] 10Mbps
      P2P Adjacency-SID:
        IPV4, SID: 17, Flags: 0x30, Weight: 0
    To: R3.00(192.168.100.3), Local: 10.100.23.1, Remote: 10.100.23.2
      Local interface index: 334, Remote interface index: 336
      Color: 0 none
      Metric: 10
      IGP metric: 10
      Static BW: 10Mbps
      Reservable BW: 10Mbps
      Available BW [priority] bps:
          [0] 10Mbps       [1] 10Mbps      [2] 10Mbps      [3] 10Mbps
          [4] 10Mbps       [5] 10Mbps      [6] 10Mbps      [7] 10Mbps
      Interface Switching Capability Descriptor(1):
        Switching type: Packet
        Encoding type: Packet
        Maximum LSP BW [priority] bps:
          [0] 10Mbps       [1] 10Mbps      [2] 10Mbps      [3] 10Mbps
          [4] 10Mbps       [5] 10Mbps      [6] 10Mbps      [7] 10Mbps
      P2P Adjacency-SID:
        IPV4, SID: 16, Flags: 0x30, Weight: 0
   Prefixes:
     192.168.100.2/32
      Metric: 0, Flags: 0x00
```

```
      Prefix-SID:
        SID: 105, Flags: 0x40, Algo: 0
   SPRING-Capabilities:
     SRGB block [Start: 800000, Range: 4000, Flags: 0xc0]
   SPRING-Algorithms:
     Algo: 0
NodeID: PCE.00(11.0.0.199)
  Type: Rtr, Age: 267 secs, LinkIn: 0, LinkOut: 0
  Protocol: IS-IS(2)
   11.0.0.199
```

Meaning

The traffic engineering database includes entries advertised from Routers R1, R2, and R3, which the PCE uses for external path computing for the PCC.

Verify SR-TE LSPs

Purpose

Verify the creation of SR-TE LSPs on the PCC.

Action

From operational mode, run the `show path-computation-client lsp`, `show spring-traffic-engineering lsp detail`, and `show route protocol spring-te` commands.

```
user@PCC> show path-computation-client lsp

  Name                              Status            PLSP-Id  LSP-Type       Controller
Path-Setup-Type       Template
  adj_sid_lsp                       (Up)              3        ext-provised
pce1              spring-te
  node_sid_lsp                      (Up)              5        ext-provised
pce1              spring-te
```

```
user@PCC> show spring-traffic-engineering lsp detail
Name: adj_sid_lsp
To: 192.168.100.3
```

```
State: Up, Outgoing interface: ge-0/0/5.0
Delegation info:
   Control-status: Externally controlled
   Routing-status: Externally routed
SR-ERO hop count: 3
  Hop 1 (Strict):
    NAI: IPv4 Adjacency ID, 10.100.41.1 -> 10.100.41.2
    SID type: 20-bit label, Value: 16
  Hop 2 (Strict):
    NAI: IPv4 Adjacency ID, 10.100.12.1 -> 10.100.12.2
    SID type: 20-bit label, Value: 17
  Hop 3 (Strict):
    NAI: IPv4 Adjacency ID, 10.100.23.1 -> 10.100.23.2
    SID type: 20-bit label, Value: 16

Name: node_sid_lsp
To: 192.168.100.3
State: Up, Outgoing interface: ge-0/0/5.0
Delegation info:
   Control-status: Externally controlled
   Routing-status: Externally routed
SR-ERO hop count: 3
  Hop 1 (Strict):
    NAI: IPv4 Adjacency ID, 10.100.41.1 -> 10.100.41.2
    SID type: 20-bit label, Value: 16
  Hop 2 (Strict):
    NAI: IPv4 Node ID, Node address: 192.168.100.1
    SID type: 20-bit label, Value: 800105
  Hop 3 (Strict):
    NAI: IPv4 Node ID, Node address: 192.168.100.2
    SID type: 20-bit label, Value: 800103

Name: static_srte_lsp_1
Tunnel-source: Static configuration
To: 192.168.100.3
State: Up
  Path: static_seg_list_1
  Outgoing interface: NA
  Delegation info:
    Control-status: Externally controlled
    Routing-status: Externally routed
  Auto-translate status: Disabled Auto-translate result: N/A
  BFD status: Up BFD name: V4-srte_bfd_session-4
```

```
  SR-ERO hop count: 2
    Hop 1 (Strict):
      NAI: IPv4 Adjacency ID, 13.1.1.2 -> 36.12.16.1
      SID type: None
    Hop 2 (Strict):
      NAI: IPv4 Node ID, Node address: 192.168.100.3
      SID type: 20-bit label, Value: 804000

Total displayed LSPs: 3 (Up: 3, Down: 0)
```

```
user@PCC> show route protocol spring-te
inet.0: 17 destinations, 17 routes (17 active, 0 holddown, 0 hidden)

inet.3: 3 destinations, 4 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.100.3/32      *[SPRING-TE/8] 00:23:32, metric 0
                       to 10.100.41.2 via ge-0/0/5.0, Push 16, Push 17(top)
                     > to 10.100.41.2 via ge-0/0/5.0, Push 800103, Push 800105(top)

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

mpls.0: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)

inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
```

Meaning

The outputs show that two SR-TE LSPs—`adj_sid_lsp` and `node_sid_lsp`—have been created by the PCE for the adjacency and node segments, respectively.

The segment routing LSP, `static_srte_lsp_1`, is enabled with the delegation capability. The `Delegation info` field shows the control and routing status of PCE-delegated LSPs. `Externally controlled` signifies that the PCE has control over the LSPs. `Externally routed` signifies that the PCE has provided the ERO for the source-routing path.

Verify Tunnel Route Creation

Purpose

Verify the tunnel routes created for the SR-TE LSPs that are included in the inet.3 routing table on the PCC.

Action

From operation mode, run the `show route table inet.3 extensive` command.

```
user@PCC> show route table inet.3 extensive
inet.3: 3 destinations, 4 routes (3 active, 0 holddown, 0 hidden)
192.168.100.1/32 (1 entry, 1 announced)
        *L-ISIS Preference: 14
                Level: 2
                Next hop type: Router, Next hop index: 581
                Address: 0xb7a23b0
                Next-hop reference count: 13
                Next hop: 10.100.41.2 via ge-0/0/5.0, selected
                Session Id: 0x172
                State: Active Int
                Local AS:    64496
                Age: 45:51      Metric: 10
                Validation State: unverified
                ORR Generation-ID: 0
                Task: IS-IS
                Announcement bits (2): 0-Resolve tree 1 2-Resolve tree 3
                AS path: I

192.168.100.3/32 (2 entries, 1 announced)
        *SPRING-TE Preference: 8
                Next hop type: Router, Next hop index: 0
                Address: 0xb61c190
                Next-hop reference count: 7
                Next hop: 10.100.41.2 via ge-0/0/5.0 weight 0x1
                Label operation: Push 16, Push 17(top)
                Label TTL action: prop-ttl, prop-ttl(top)
                Load balance label: Label 16: None; Label 17: None;
                Label element ptr: 0xb7a2a60
                Label parent element ptr: 0x0
                Label element references: 5
```

```
                Label element child references: 0
                Label element lsp id: 0
                Session Id: 0x0
                Next hop: 10.100.41.2 via ge-0/0/5.0 weight 0x1, selected
                Label operation: Push 800103, Push 800105(top)
                Label TTL action: prop-ttl, prop-ttl(top)
                Load balance label: Label 800103: None; Label 800105: None;
                Label element ptr: 0xb7a2c40
                Label parent element ptr: 0x0
                Label element references: 2
                Label element child references: 0
                Label element lsp id: 0
                Session Id: 0x0
                State: Active Int
                Local AS:    64496
                Age: 9:44       Metric: 0
                Validation State: unverified
                Task: SPRING-TE
                Announcement bits (2): 0-Resolve tree 1 2-Resolve tree 3
                AS path: I
        L-ISIS Preference: 14
                Level: 2
                Next hop type: Router, Next hop index: 0
                Address: 0xb7a28f0
                Next-hop reference count: 1
                Next hop: 10.100.41.2 via ge-0/0/5.0, selected
                Label operation: Push 800103
                Label TTL action: prop-ttl
                Load balance label: Label 800103: None;
                Label element ptr: 0xb7a2880
                Label parent element ptr: 0x0
                Label element references: 1
                Label element child references: 0
                Label element lsp id: 0
                Session Id: 0x0
                State: Int
                Inactive reason: Route Preference
                Local AS:    64496
                Age: 45:40      Metric: 30
                Validation State: unverified
                ORR Generation-ID: 0
                Task: IS-IS
                AS path: I
```

```
192.168.100.2/32 (1 entry, 1 announced)
        *L-ISIS Preference: 14
                Level: 2
                Next hop type: Router, Next hop index: 0
                Address: 0xb7a29b0
                Next-hop reference count: 1
                Next hop: 10.100.41.2 via ge-0/0/5.0, selected
                Label operation: Push 800105
                Label TTL action: prop-ttl
                Load balance label: Label 800105: None;
                Label element ptr: 0xb7a2940
                Label parent element ptr: 0x0
                Label element references: 1
                Label element child references: 0
                Label element lsp id: 0
                Session Id: 0x0
                State: Active Int
                Local AS:    64496
                Age: 45:40      Metric: 20
                Validation State: unverified
                ORR Generation-ID: 0
                Task: IS-IS
                Announcement bits (2): 0-Resolve tree 1 2-Resolve tree 3
                AS path: I
```

Meaning

Tunnel routes have been created for the PCE-controlled LSP destination with SR-TE as the protocol label.

Verify Forwarding Table Entries

Purpose

Verify that the SR-TE LSP destination to Router R3 is installed in the forwarding table of the PCC.

Action

From operation mode, run the `show route forwarding-table destination` *ip-address* `extensive` command.

```
user@PCC> show route forwarding-table destination 192.168.100.3 extensive
Routing table: default.inet [Index 0]
Internet:
Enabled protocols: Bridging,

Destination:  192.168.100.3/32
  Route type: user
  Route reference: 0                   Route interface-index: 0
  Multicast RPF nh index: 0
  P2mpidx: 0
  Flags: sent to PFE, rt nh decoupled
  Nexthop: 10.100.41.2
  Next-hop type: unicast               Index: 581      Reference: 14
  Next-hop interface: ge-0/0/5.0

Routing table: __pfe_private__.inet [Index 3]
Internet:
Enabled protocols: Bridging,

Destination:  default
  Route type: permanent
  Route reference: 0                   Route interface-index: 0
  Multicast RPF nh index: 0
  P2mpidx: 0
  Flags: sent to PFE
  Next-hop type: discard               Index: 517      Reference: 2

Routing table: __juniper_services__.inet [Index 5]
Internet:
Enabled protocols: Bridging,

Destination:  default
  Route type: permanent
  Route reference: 0                   Route interface-index: 0
  Multicast RPF nh index: 0
  P2mpidx: 0
  Flags: sent to PFE
  Next-hop type: discard               Index: 530      Reference: 2
```

```
Routing table: __master.anon__.inet [Index 6]
Internet:
Enabled protocols: Bridging, Dual VLAN,

Destination:  default
  Route type: permanent
  Route reference: 0                    Route interface-index: 0
  Multicast RPF nh index: 0
  P2mpidx: 0
  Flags: sent to PFE
  Next-hop type: reject                 Index: 545      Reference: 1
```

Meaning

The SR-TE LSP destination IP address to Router R3 is installed as a forwarding entry.

Verify the Use of Tunnel Routes for Static Route Forwarding

Purpose

Verify that the static route is taking the tunnel route created for the SR-TE LSPs.

Action

From operational mode, run the `show route` *ip-address* and `show route forwarding-table destination` *ip-address* commands.

```
user@PCC> show route 100.1.1.1
inet.0: 17 destinations, 17 routes (17 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

100.1.1.1/32       *[Static/5] 00:33:36, metric2 0
                    > to 10.100.41.2 via ge-0/0/5.0, Push 16, Push 17(top)
                      to 10.100.41.2 via ge-0/0/5.0, Push 800103, Push 800105(top)
```

```
user@PCC> show route forwarding-table destination 100.1.1.1
Routing table: default.inet
Internet:
```

```
Enabled protocols: Bridging,
Destination        Type RtRef Next hop           Type Index    NhRef Netif
100.1.1.1/32       user     0                    indr  1048575     2
                              10.100.41.2      Push 16, Push 17(top)      590     2 ge-0/0/5.0

Routing table: __pfe_private__.inet
Internet:
Enabled protocols: Bridging,
Destination        Type RtRef Next hop           Type Index    NhRef Netif
default            perm     0                    dscd      517     2

Routing table: __juniper_services__.inet
Internet:
Enabled protocols: Bridging,
Destination        Type RtRef Next hop           Type Index    NhRef Netif
default            perm     0                    dscd      530     2

Routing table: __master.anon__.inet
Internet:
Enabled protocols: Bridging, Dual VLAN,
Destination        Type RtRef Next hop           Type Index    NhRef Netif
default            perm     0                    rjct      545     1
```

Meaning

The outputs show that the static route to Router R3 uses the tunnel route created for the SR-TE LSP.

Static Segment Routing Label Switched Path

IN THIS SECTION

- Static Segment Routing LSP in MPLS Networks | **2235**
- Example: Configuring Static Segment Routing Label Switched Path | **2253**

The segment routing architecture enables the ingress devices in a core network to steer traffic through explicit paths. You can configure these paths using segment lists to define the paths that the incoming traffic should take. The incoming traffic may be labeled or IP traffic, causing the forwarding operation at the ingress device to be either a label swap, or a destination-based lookup.

Static Segment Routing LSP in MPLS Networks

IN THIS SECTION

- Introduction to Segment Routing LSPs | 2235
- Benefits of using Segment Routing LSPs | 2237
- Colored Static Segment Routing LSP | 2237
- Non-Colored Static Segment Routing LSP | 2237
- Static Segment Routing LSP Provisioning | 2243
- Static Segment Routing LSP Limitations | 2243
- Color-Based Mapping of VPN Services | 2244
- Tunnel Templates for PCE-Initiated Segment Routing LSPs | 2252

Source packet routing or segment routing is a control-plane architecture that enables an ingress devices in a core network to steer traffic through a specific set of nodes and links in the network without relying on the intermediate nodes in the network to determine the actual path it should take. You can configure these paths using segment lists to define the paths that the incoming traffic should take. The incoming traffic may be labeled or IP traffic, causing the forwarding operation at the ingress device to be either a label swap, or a destination-based lookup.

Introduction to Segment Routing LSPs

Segment routing leverages the source routing paradigm. A device steers a packet through an ordered list of instructions, called segments. A segment can represent any instruction, topological or service-based. A segment can have a local semantic to a segment routing node or to a global node within a segment routing domain. Segment routing enforces a flow through any topological path and service chain while maintaining per-flow state only at the ingress device to the segment routing domain. Segment routing can be directly applied to the MPLS architecture with no change on the forwarding plane. A segment is encoded as an MPLS label. An ordered list of segments is encoded as a stack of labels. The segment to process is on the top of the stack. Upon completion of a segment, the related label is popped from the stack.

Segment routing LSPs can either be dynamic or static in nature.

Dynamic segment routing LSPs—When a segment routing LSP is created either by an external controller and downloaded to an ingress device through Path Computation Element Protocol (PCEP) extensions, or from a BGP segment routing policy through BGP segment routing extensions, the LSP is dynamically provisioned. The segment list of the dynamic segment routing LSP is contained in the PCEP Explicit Route Object (ERO), or the BGP segment routing policy of the LSP.

Static segment routing LSPs—When a segment routing LSP is created on the ingress device through local configuration, the LSP is statically provisioned.

A static segment routing LSP can further be classified as colored and non-colored LSPs based on the configuration of the `color` statement at the `[edit protocols source-packet-routing source-routing-path` *`lsp-name`*`]` hierarchy level.

For example:

```
[edit protocols]
    source-packet-routing {
    source-routing-path lsp_name {
        to destination_address;
        color color_value;
        binding-sid binding-label;
        primary segment_list_1_name weight weight;
        ...
        primary segment_list_n_name weight weight;
        secondary segment_list_n_name;
        sr-preference sr_preference_value;
    }
}
```

Here, each primary and secondary statement refers to a segment list.

```
[edit protocols]
source-packet-routing {
    segment-list segment_list_name {
        hop_1_name label sid_label;
        ...
        hop_n_name label sid_label;
    }
}
```

Benefits of using Segment Routing LSPs

- Static segment routing does not rely on per LSP forwarding state on transit routers. Hence, removing the need of provisioning and maintaining per LSP forwarding state in the core.
- Provide higher scalability to MPLS networks.

Colored Static Segment Routing LSP

A static segment routing LSP configured with the `color` statement is called a colored LSP.

Understanding Colored Static Segment Routing LSPs

Similar to a BGP segment routing policy, the ingress route of the colored LSP is installed in the `inetcolor.0` or `inet6color.0` routing tables, with `destination-ip-address, color` as key for mapping IP traffic.

A static colored segment routing LSP may have a binding SID, for which a route is installed in the `mpls.0` routing table. This binding SID label is used to map labeled traffic to the segment routing LSP. The gateways of the route are derived from the segment list configurations under the primary and secondary paths.

Segment List of Colored Segment Routing LSPs

The colored static segment routing LSPs already provide support for first hop label mode of resolving an LSP. However, first hop IP mode is not supported for colored segment routing LSPs. A commit check feature is introduced to ensure that all the segment lists contributing to the colored routes have the minimum label present for all hops. If this requirement is not met, the commit is blocked.

Non-Colored Static Segment Routing LSP

A static segment routing LSP that is configured without the `color` statement is a non-colored LSP. Similar to PCEP segment routing tunnels, the ingress route is installed in the `inet.3` or `inet6.3` routing tables.

Junos OS supports non-colored static segment routing LSPs on ingress routers. You can provision non-colored static segment routing LSP by configuring one source routed path and one or more segment lists. These segment lists can be used by multiple non-colored segment routing LSPs.

Understanding Non-Colored Segment Routing LSPs

The non-colored segment routing LSP has a unique name and a destination IP address. An ingress route to the destination is installed in the inet.3 routing table with a default preference of 8 and a metric of 1. This route allows non-colored services to be mapped to the segment routing LSP pertaining to the destination. In case the non-colored segment routing LSP does not require an ingress route then the

ingress route can be disabled. A non-colored segment routing LSP uses binding SID label to achieve segment routing LSP stitching. This label that can be used to model the segment routing LSP as a segment that may be further used to construct other segment routing LSPs in a hierarchical manner. The transit of the binding SID label, by default, has a preference of 8 and a metric of 1.

Statically configured non-colored segment routing LSPs on the ingress device are reported to the Path Computation Element (PCE) through a Path Computation Element Protocol (PCEP) session. These non-colored segment routing LSPs may have binding service identifier (SID) labels associated with them. With this feature, the PCE can use this binding SID label in the label stack to provision PCE-initiated segment routing LSP paths.

A non-colored segment routing LSP can have a maximum of 8 primary paths. If there are multiple operational primary paths then the packet forwarding engine (PFE) distributes traffic over the paths based on the load balancing factors like the weight configured on the path. This is equal cost multi path (ECMP) if none of the paths have a weight configured on them or weighted ECMP if at least one of the paths has a non-zero weight configured on the paths. In both the cases, when one or some of the paths fail, the PFE rebalances the traffic over the remaining paths that automatically leads to achieving path protection. A non-colored segment routing LSP can have a secondary path for dedicated path protection. Upon failure of a primary path, the PFE rebalances the traffic to the remaining functional primary paths. Otherwise, the PFE switches the traffic to the backup path, hence achieving path protection. A non-colored segment routing LSP may specify a metric at `[edit protocols source-packet-routing source-routing-path` *`lsp-name`*`]` for its ingress and binding-SID routes. Multiple non-colored segment routing LSPs have the same destination address that contribute to the next hop of the ingress route.

Multiple non-colored segment routing LSPs have the same destination address that contribute to the next hop of the ingress route. Each path ,either primary or secondary, of each segment routing LSP is considered as a gateway candidate, if the path is functional and the segment routing LSP has the best preference of all these segment routing LSPs. However, the maximum number of gateways that the next-hop can hold cannot exceed the RPD multi-path limit, which is 128 by default. Extra paths are pruned, firstly secondary paths and then primary paths. A given segment list may be referred multiple times as primary or secondary paths by these segment routing LSPs. In this case, there are multiple gateways, each having a unique segment routing LSP tunnel ID. These gateways are distinct, although they have identical outgoing label stack and interface. A non-colored segment routing LSP and a colored segment routing LSP may also have the same destination address. However, they correspond to different destination addresses for ingress routes, as the colored segment routing LSP's destination address is constructed with both its destination address and color.

NOTE: In the case where a static non-colored segment routing LSP and a PCEP-created segment routing LSP co-exist and have the same to address that contributes to the same ingress route, if they also have the same preference. Otherwise, the segment routing LSP with the best preference is installed for the route.

Segment List of Non-Colored Segment Routing LSPs

A segment list consists of a list of hops. These hops are based on the SID label or an IP address. The number of SID labels in the segment list should not exceed the maximum segment list limit. Maximum segment-list binding to a LSP tunnel is increased from 8 to 128, with maximum 1000 tunnels per system. A maximum of 128 primary paths are supported per static segment routing LSP. You can configure the maximum segment list limit at the `[edit protocols source-packet-routing]` hierarchy level.

The first hop of the non-colored static LSPs provides support for SID labels, in addition to IP addresses. With the first hop label support, MPLS fast reroute (FRR) and weighted equal-cost multipath is enabled for resolving the static non-colored segment routing LSPs, similar to colored static LSPs.

For the first-hop label mode to take effect, you must include the `inherit-label-nexthops` statement globally or individually for a segment list, and the first hop of the segment list must include both IP address and label. If the first hop includes only IP address, the `inherit-label-nexthops` statement does not have any effect.

You can configure `inherit-label-nexthops` at any one of the following hierarchies. The `inherit-label-nexthops` statement takes effect only if the segment list first hop includes both IP address and label.

- **Segment list level**—At the `[edit protocols source-packet-routing segment-list` *`segment-list-name`*`]` hierarchy level.
- **Globally**—At the `[edit protocols source-packet-routing]` hierarchy level.

When the `inherit-label-nexthops` statement is configured globally, it takes precedence over the segment-list level configuration, and the `inherit-label-nexthops` configuration is applied to all the segment lists. When the `inherit-label-nexthops` statement is not configured globally, only segment lists with both labels and IP address present in the first hop, and configured with `inherit-label-nexthops` statement are resolved using SID labels.

For dynamic non-colored static LSPs, that is the PCEP-driven segment routing LSPs, the `inherit-label-nexthops` statement must be enabled globally, as the segment-level configuration is not applied.

Table 4 describes the mode of segment routing LSP resolution based on the first hop specification.

Table 49: Non-Colored Static LSP Resolution Based on First Hop Specification

First Hop Specification	Mode of LSP Resolution

IP address only
 	
For example:
 	
`segment-list path-1 {`
`  hop-1 ip-address 172.16.12.2;`
`  hop-2 label 1000012;`
`  hop-3 label 1000013;`
`  hop-4 label 1000014;`
`}`
	The segment list is resolved using the IP address.

SID only
 	
For example:
 	
`segment-list path-2 {`
`  hop-1 label 1000011;`
`  hop-2 label 1000012;`
`  hop-3 label 1000013;`
`  hop-4 label 1000014;`
`}`
	The segment list is resolved using SID labels.

IP address and SID (without the `inherit-label-nexthops` configuration)
 	
For example:
 	
`segment-list path-3 {`
`  hop1 {`
`  label 801006;`
`  ip-address 172.16.1.2;`
`  }`
`  hop-2 label 1000012;`
`  hop-3 label 1000013;`
`  hop-4 label 1000014;`
`}`
	By default, the segment list is resolved using IP address.

Table 49: Non-Colored Static LSP Resolution Based on First Hop Specification *(Continued)*

First Hop Specification	Mode of LSP Resolution

IP address and SID (with the `inherit-label-nexthops` configuration)
 	
For example:
 	
`segment-list path-3 {`
`  inherit-label-nexthops;`
`  hop1 {`
`  label 801006;`
`  ip-address 172.16.1.2;`
`  }`
`  hop-2 label 1000012;`
`  hop-3 label 1000013;`
`  hop-4 label 1000014;`
`}`
	The segment list is resolved using SID labels.

You can use the `show route` *ip-address* `protocol spring-te active-path table inet.3` command to view the non-colored segment routing traffic-engineered LSPs having multiple segment lists installed in the inet.3 routing table.

For example:

```
user@host> show route 10.7.7.7 protocol spring-te active-path table inet.3
inet.3: 42 destinations, 59 routes (41 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

10.7.7.7/32         *[SPRING-TE/8] 00:01:25, metric 1, metric2 0
                    > to 10.11.1.2 via et-0/0/0.1, Push 801007
                      to 10.21.1.2 via et-0/0/2.1, Push 801007
                      to 10.102.1.2 via et-0/0/0.2, Push 801007, Push 801002(top)
                      to 10.21.1.2 via et-0/0/2.2, Push 801007, Push 801005(top)
                      to 10.103.1.2 via et-0/0/0.3, Push 801007, Push 801003(top)
                      to 10.203.1.2 via et-0/0/2.3, Push 801007, Push 801006(top)
                      to 10.104.1.2 via et-0/0/0.4, Push 801007, Push 801003, Push 801002(top)
                      to 10.204.1.2 via et-0/0/2.4, Push 801007, Push 801006, Push 801005(top)
```

NOTE: The first hop type of segment lists of a static segment routing LSP can cause a commit to fail, if:

- Different segment lists of a tunnel have different first hop resolution types. This is applicable to both colored and non-colored static segment routing LSPs. However, this does not apply for PCEP-driven LSPs; a system log message is generated for the mismatch in the first hop resolution type at the time of computing the path.

  For example:

```
segment-list path-1 {
    hop-1 ip-address 172.16.12.2;
    hop-2 label 1000012;
    hop-3 label 1000013;
    hop-4 label 1000014;
}
segment-list path-2 {
    hop-1 label 1000011;
    hop-2 label 1000012;
    hop-3 label 1000013;
    hop-4 label 1000014;
}
source-routing-path lsp1 {
    to 172.16.10.1;
    primary {
        path-1;
        path-2;
    }
}
```

  The commit of tunnel *lsp1* fails, as path-1 is of IP address mode and path-2 is of label mode.

- The binding SID is enabled for the static non-colored LSP whose segment list type is SID label.

  For example:

```
segment-list path-3 {
    hop-1 label 1000011;
```

```
            hop-2 label 1000012;
            hop-3 label 1000013;
            hop-4 label 1000014;
        }
        source-routing-path lsp1 {
            to 172.16.10.1;
            binding-sid 333;
            primary {
                path-3;
            }
        }
```

Static Segment Routing LSP Provisioning

Segment provisioning is performed on per-router basis. For a given segment on a router, a unique service identifier (SID) label is allocated from a desired label pool which may be from the dynamic label pool for an adjacency SID label or from the segment routing global block (SRGB) for a prefix SID or node SID. The adjacency SID label can be dynamically allocated, which is the default behavior, or be allocated from a local static label pool (SRLB). A route for the SID label is then installed in the mpls.0 table.

Junos OS allows static segment routing LSPs by configuring the `segment` statement at the `[edit protocols mpls static-label-switched-path` *`static-label-switched-path`*`]` hierarchy level. A static segment LSP is identified by a unique SID label that falls under Junos OS static label pool. You can configure the Junos OS static label pool by configuring the `static-label-range` *`static-label-range`* statement at the `[edit protocols mpls label-range]` hierarchy level.

Static Segment Routing LSP Limitations

- Junos OS currently has a limitation that the next hop cannot be built to push more than the maximum segment list depth labels. So, a segment list with more than the maximum SID labels (excluding the SID label of the first hop which is used to resolve forwarding next-hop) is not usable for colored or non-colored segment routing LSPs. Also, the actual number allowed for a given segment routing LSP may be even lower than the maximum limit, if an MPLS service is on the segment routing LSP or the segment routing LSP is on a link or a node protection path. In all cases, the total number of service labels, SID labels, and link or node protection labels must not exceed the maximum segment list depth. You can configure the maximum segment list limit at `[edit protocols source-packet-routing]` hierarchy level. Multiple non-colored segment routing LSPs with less than or equal to the maximum SID labels can be stitched together to construct a longer segment routing LSP. This is called segment routing LSP stitching. It can be achieved using binding-SID label.

- The segment routing LSP stitching is actually performed at path level. If a non-colored segment routing LSP has multiple paths that is multiple segment lists, each path can be independently stitched to another non-colored segment routing LSP at a stitching point. A non-colored segment routing LSP which is dedicated to stitching may disable ingress route installation by configuring `no-ingress` statement at `[edit protocols source-packet-routing source-routing-path lsp-name]` hierarchy level.
- A maximum of 128 primary paths and 1 secondary path are supported per non-colored static segment routing LSP. If there is a violation in configuration, commit check fails with an error.
- Maximum segment-list binding to a LSP tunnel is increased from 8 to 128, with maximum 1000 tunnels per system. A maximum of 128 primary paths are supported per static segment routing LSP. As a limitation, the maximum sensor support for LSP path is 32000 only.
- If any segment-list is configured with more labels than the maximum segment list depth, the configuration commit check fails with an error.

Color-Based Mapping of VPN Services

You can specify color as a protocol next hop constraint (in addition to the IPv4 or IPv6 address) for resolving transport tunnels over static colored and BGP segment routing traffic-engineered (SR-TE) LSPs. This is called the color-IP protocol next hop resolution, where you are required to configure a resolution-map and apply to the VPN services. With this feature, you can enable color-based traffic steering of Layer 2 and Layer 3 VPN services.

Junos OS supports colored SR-TE LSPs associated with a single color. The color-based mapping of VPN services feature is supported on static colored LSPs and BGP SR-TE LSPs.

VPN Service Coloring

In general, a VPN service may be assigned a color on the egress router where the VPN NLRI is advertised, or on an ingress router where the VPN NLRI is received and processed.

You can assign a color to the VPN services at different levels:

- Per routing instance.
- Per BGP group.
- Per BGP neighbor.
- Per prefix.

Once you assign a color, the color is attached to a VPN service in the form of BGP color extended community.

You can assign multiple colors to a VPN service, referred to as multi-color VPN services. In such cases, the last color attached is considered as the color of the VPN service, and all other colors are ignored.

Multiple colors are assigned by egress and/or ingress devices through multiple policies in the following order:

- BGP export policy on the egress device.
- BGP import policy on the ingress device.
- VRF import policy on the ingress device.

The two modes of VPN service coloring are:

Egress Color Assignment

In this mode, the egress device (that is, the advertiser of the VPN NLRI) is responsible for coloring the VPN service. To enable this mode, you can define a routing policy, and apply it in the VPN service's routing-instance `vrf-export`, group export, or group neighbor export at the `[edit protocols bgp]` hierarchy level. The VPN NLRI is advertised by BGP with the specified color extended community.

For example:

```
[edit policy-options]
community red-comm {
    members color:0:50;
}
```

```
[edit policy-options]
policy-statement pol-color {
    term t1 {
        from {
            [any match conditions];
        }
        then {
            community add red-comm;
            accept;
        }
```

```
    }
}
```

```
[edit routing-instances]
vpn-X {
    ...
    vrf-export pol-color ...;
}
```

Or

NOTE: When you apply the routing policy as an export policy of a BGP group or BGP neighbor, you must include the `vpn-apply-export` statement at the BGP, BGP group, or BGP neighbor level in order for the policy to take an effect on the VPN NLRI.

```
[edit protocols bgp]
group PEs {
    ...
    neighbor PE-A {
        export pol-color ...;
        vpn-apply-export;
    }
}
```

The routing policies are applied to Layer 3 VPN prefix NLRIs, Layer 2 VPN NRLIs, and EVPN NLRIs. The color extended community is inherited by all the VPN routes, imported, and installed in the target VRFs on one or multiple ingress devices.

Ingress Color Assignment

In this mode, the ingress device (that is, the receiver of the VPN NLRI) is responsible for coloring the VPN service. To enable this mode, you can define a routing policy, and apply it to the VPN service's routing-instance `vrf-import`, group import, or group neighbor import at the `[edit protocols bgp]` hierarchy level. All the VPN routes matching the routing policy is attached with the specified color extended community.

For example:

```
[edit routing-options]
community red-comm {
    members color:0:50;
}
```

```
[edit policy-options]
policy-statement pol-color {
    term t1 {
        from {
            [any match conditions];
        }
        then {
            community add red-comm;
            accept;
        }
    }
}
```

```
[edit routing-instances]
vpn-Y {
    ...
    vrf-import pol-color ...;
}
```

Or

```
[edit protocols bgp]
group PEs {
    ...
    neighbor PE-B {
        import pol-color ...;
    }
}
```

Specifying VPN Service Mapping Mode

To specify flexible VPN service mapping modes, you must define a policy using the `resolution-map` statement, and refer the policy in a VPN service's routing-instance `vrf-import`, group import, or group neighbor import at the `[edit protocols bgp]` hierarchy level. All the VPN routes matching the routing policy are attached with the specified resolution-map.

For example:

```
[edit policy-options]
resolution-map map-A {
    <mode-1>;
    <mode-2>;
    ...
}
policy-statement pol-resolution {
    term t1 {
        from {
            [any match conditions];
        }
        then {
            resolution-map map-A;
            accept;
        }
    }
}
```

You can apply import policy to the VPN service's routing-instance.

```
[edit routing-instances]
vpn-Y {
    ...
    vrf-import pol-resolution ...;
}
```

You can also apply the import policy to a BGP group or BGP neighbor.

```
[edit protocols bgp]
group PEs {
    ...
    neighbor PE-B {
```

```
        import pol-resolution ...;
    }
}
```

NOTE: Each VPN service mapping mode should have a unique name defined in the resolution-map. Only a single entry of IP-color is supported in the resolution-map, where the VPN route(s) are resolved using a colored-IP protocol next hop in the form of `ip-address:color`.

Color-IP Protocol Next Hop Resolution

The protocol next hop resolution process is enhanced to support colored-IP protocol next hop resolution. For a colored VPN service, the protocol next hop resolution process takes a color and a resolution-map, builds a colored-IP protocol next hop in the form of *IP-address:color*, and resolves the protocol next hop in the inet6color.0 routing table.

You must configure a policy to support multipath resolution of colored Layer 2 VPN, Layer 3 VPN, or EVPN services over colored LSPs. The policy must then be applied with the relevant RIB table as the resolver import policy.

For example:

```
[edit policy-options]
policy-statement mpath {
    then multipath-resolve;
}
```

```
[edit routing-options]
resolution {
    rib bgp.l3vpn.0 {
        inetcolor-import mpath;
    }
}

resolution {
    rib bgp.l3vpn-inet6.0 {
        inet6color-import mpath;
    }
}
```

```
resolution {
    rib bgp.l2vpn.0 {
        inetcolor-import mpath;
    }
}

resolution {
    rib mpls.0 {
        inetcolor-import mpath;
    }
}

resolution {
    rib bgp.evpn.0 {
        inetcolor-import mpath;
    }
}
```

Fallback to IP Protocol Next Hop Resolution

If a colored VPN service does not have a resolution-map applied to it, the VPN service ignores its color and falls back to the IP protocol next hop resolution. Conversely, if a non-colored VPN service has a resolution-map applied to it, the resolution-map is ignored, and the VPN service uses the IP protocol next hop resolution.

The fallback is a simple process from colored SR-TE LSPs to LDP LSPs, by using a RIB group for LDP to install routes in inet{6}color.0 routing tables. A longest prefix match for a colored-IP protocol next hop ensures that if a colored SR-TE LSP route does not exist, an LDP route with a matching IP address should be returned.

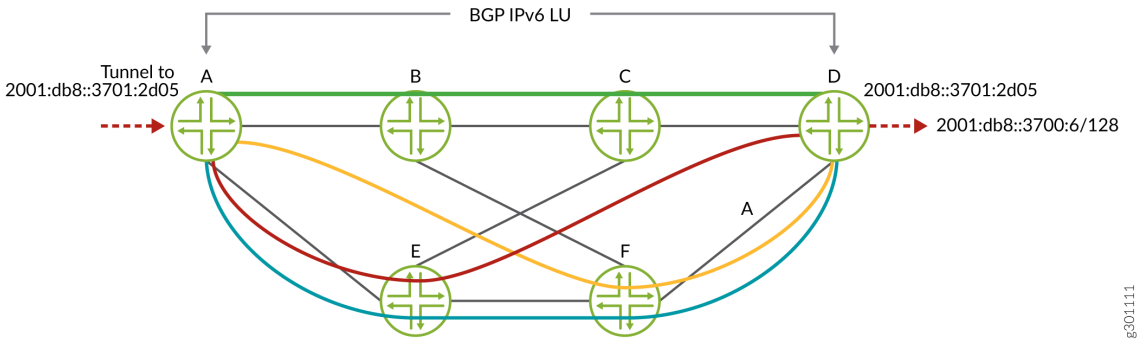
BGP Labeled Unicast Color-based Mapping over SR-TE

BGP Labeled Unicast (BGP-LU) can resolve IPv4 or IPv6 routes over segment routing–traffic engineering (SR-TE) for both IPv4 and IPv6 address families. BGP-LU supports mapping a BGP community color and defining a `resolution map` for SR-TE. A colored protocol next hop is constructed and it is resolved on a colored SR-TE tunnel in the `inetcolor.0` or `inet6color.0` table. BGP uses `inet.3` and `inet6.3` tables for non-color based mapping. This enables you to advertise BGP-LU IPv6 and IPv4 prefixes with an IPv6 next-hop address in IPv6-only networks where routers do not have any IPv4 addresses configured. With this feature, Currently we support BGP IPv6 LU over SR-TE with IS-IS underlay.

In Figure 9, the controller configures 4 colored tunnels in an IPv6 core network configured with SR-TE. Each colored tunnel takes a different path to the destination router D depending on the defined

resolution map. The controller configures a colored SR-TE tunnel to 2001:db8::3701:2d05 interface in router D . BGP imports policies to assign a color and resolution map to the received prefix 2001:db8::3700:6/128. Based on the assigned community color, BGP-LU resolves the colored next hop for BGP IPv6 LU prefix according to the assigned resolution map policy.

Figure 156: BGP IPv6 LU over colored IPv6 SR-TE



BGP-LU supports the following scenarios:

- BGP IPv4 LU over colored BGP IPv4 SR-TE, with ISIS/OSPF IPv4 SR extensions.
- BGP IPv4 LU over static colored and non-colored IPv4 SR-TE, with ISIS/OSPF IPv4 SR extensions.
- BGP IPv6 LU over colored BGP IPv6 SR-TE, with ISIS IPv6 SR extensions.
- BGP IPv6 LU over static colored and non-colored IPv6 SR-TE, with ISIS IPv6 SR extensions.
- IPv6 Layer 3 VPN services with IPv6 local address and IPv6 neighbor address.
- IPv6 Layer 3 VPN services over BGP IPv6 SR-TE, with ISIS IPv6 SR extensions.
- IPv6 Layer 3 VPN services over static-colored and non-colored IPv6 SR-TE, with ISIS IPv6 SR extensions.

Supported and Unsupported Features for Color-Based Mapping of VPN Services

The following features and functionality are supported with color-based mapping of VPN services:

- BGP Layer 2 VPN (Kompella Layer 2 VPN)
- BGP EVPN
- Resolution-map with a single IP-color option.
- Colored IPv4 and IPv6 protocol next hop resolution.

- Routing information base (also known as routing table) group based fallback to LDP LSP in inetcolor.0 routing table.
- Colored SR-TE LSP.
- Virtual platforms.
- 64-bit Junos OS.
- Logical systems.
- BGP labeled unicast.

The following features and functionality are not supported with color-based mapping of VPN services:

- Colored MPLS LSPs, such as RSVP, LDP, BGP-LU, static.
- Layer 2 circuit
- FEC-129 BGP auto-discovered and LDP-signaled Layer 2 VPN.
- VPLS
- MVPN
- IPv4 and IPv6 using resolution-map.

Tunnel Templates for PCE-Initiated Segment Routing LSPs

You can configure a tunnel template for PCE-initiated segment routing LSPs to pass down two additional parameters for these LSPs - Bidirectional forwarding detection (BFD) and LDP tunneling.

When a PCE-Initiated segment routing LSP is being created, the LSP is checked against policy statements (if any) and if there is a match, the policy applies the configured template for that LSP. The template configuration is inherited only if it is not provided by the LSP source (PCEP); for example, metric.

To configure a template:

1. Include the *source-routing-path-template* statement at the `[edit protocols source-packet-routing]` hierarchy level. You can configure the additional BFD and LDP tunneling parameters here.
2. Include the source-routing-path-template-map statement at the `[edit protocols source-packet-routing]` hierarchy level to list the policy statements against which the PCE-initiated LSP should be checked.
3. Define a policy to list the LSPs on which the template should be applied.

The `from` statement can include either the LSP name or LSP regular expression using the `lsp` and `lsp-regex` match conditions. These options are mutually exclusive, so you can specify only one option at a given point in time.

The `then` statement must include the `sr-te-template` option with an accept action. This applies the template to the PCE-initiated LSP.

Take the following into consideration when configuring a template for PCE-initiated LSPs:

- Template configuration is not applicable to staticalyy configured segment routing LSPs, or any other client's segment routing LSP.
- PCEP-provided configuration has precedence over template configuration.
- PCEP LSP does not inherit template segment-list configuration.

Example: Configuring Static Segment Routing Label Switched Path

IN THIS SECTION

- Requirements | **2253**
- Overview | **2254**
- Configuration | **2254**
- Verification | **2268**

This example shows how to configure static segment routing label switched paths (LSPs) in MPLS networks. This configuration helps to bring higher scalability to MPLS networks.

Requirements

This example uses the following hardware and software components:

- Seven MX Series 5G Universal Routing Platforms
- Junos OS Release 18.1 or later running on all the routers

Before you begin, be sure you configure the device interfaces.

Overview

IN THIS SECTION

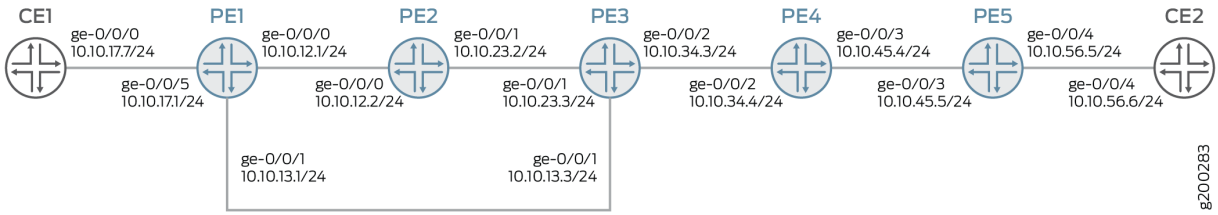
- Topology | **2254**

Junos OS a set of explicit segment routing paths are configured on the ingress router of a non-colored static segment routing tunnel by configuring the `segment-list` statement at the `[edit protocols source-packet-routing]` hierarchy level. You can configure segment routing tunnel by configuring the `source-routing-path` statement at `[edit protocols source-packet-routing]` hierarchy level. The segment routing tunnel has a destination address and one or more primary paths and optionally secondary paths that refer to the segment list. Each segment list consists of a sequence of hops. For non-colored static segment routing tunnel, the first hop of the segment list specifies an immediate next hop IP address and the second to Nth hop specifies the segment identifies (SID) labels corresponding to the link or node which the path traverses. The route to the destination of the segment routing tunnel is installed in inet.3 table.

Topology

In this example, configure layer 3 VPN on the provider edge routers PE1 and PE5. Configure the MPLS protocol on all the routers. The segment routing tunnel is configured from router PE1 to router PE5 with a primary path configured on router PE1 and router PE5 . Router PE1 is also configured with secondary path for path protection. The transit routers PE2 to PE4 are configured with adjacency SID labels with label pop and an outgoing interface.

Figure 157: Static Segment Routing Label Switched Path



Configuration

IN THIS SECTION

- CLI Quick Configuration | **2255**

- Configuring Device PE1 | 2259
- Configuring Device PE2 | 2265
- Results | 2267

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

PE1

```
set interfaces ge-0/0/0 unit 0 family inet address 10.10.12.1/24
set interfaces ge-0/0/0 unit 0 family mpls maximum-labels 5
set interfaces ge-0/0/1 unit 0 family inet address 10.10.13.1/24
set interfaces ge-0/0/1 unit 0 family mpls maximum-labels 5
set interfaces ge-0/0/5 unit 0 family inet address 10.10.17.1/24
set routing-options autonomous-system 65000
set routing-options forwarding-table export load-balance-policy
set routing-options forwarding-table chained-composite-next-hop ingress l3vpn
set protocols mpls interface ge-0/0/0.0
set protocols mpls interface ge-0/0/1.0
set protocols mpls label-range static-label-range 1000000 1000999
set protocols bgp group pe type internal
set protocols bgp group pe local-address 192.168.147.211
set protocols bgp group pe family inet-vpn unicast
set protocols bgp group pe neighbor 192.168.146.181
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols source-packet-routing segment-list sl-15-primary hop-1 ip-address 10.10.13.3
set protocols source-packet-routing segment-list sl-15-primary hop-2 label 1000134
set protocols source-packet-routing segment-list sl-15-primary hop-3 label 1000145
set protocols source-packet-routing segment-list sl-15-backup hop-1 ip-address 10.10.12.2
set protocols source-packet-routing segment-list sl-15-backup hop-2 label 1000123
set protocols source-packet-routing segment-list sl-15-backup hop-3 label 1000134
set protocols source-packet-routing segment-list sl-15-backup hop-4 label 1000145
set protocols source-packet-routing source-routing-path lsp-15 to 192.168.146.181
set protocols source-packet-routing source-routing-path lsp-15 binding-sid 1000999
```

```
set protocols source-packet-routing source-routing-path lsp-15 primary sl-15-primary
set protocols source-packet-routing source-routing-path lsp-15 secondary sl-15-backup
set policy-options policy-statement VPN-A-export term a from protocol ospf
set policy-options policy-statement VPN-A-export term a from protocol direct
set policy-options policy-statement VPN-A-export term a then community add VPN-A
set policy-options policy-statement VPN-A-export term a then accept
set policy-options policy-statement VPN-A-export term b then reject
set policy-options policy-statement VPN-A-import term a from protocol bgp
set policy-options policy-statement VPN-A-import term a from community VPN-A
set policy-options policy-statement VPN-A-import term a then accept
set policy-options policy-statement VPN-A-import term b then reject
set policy-options policy-statement bgp-to-ospf from protocol bgp
set policy-options policy-statement bgp-to-ospf from route-filter 10.10.0.0/16 orlonger
set policy-options policy-statement bgp-to-ospf then accept
set policy-options policy-statement load-balance-policy then load-balance per-packet
set policy-options community VPN-A members target:65000:1
set routing-instances VRF1 instance-type vrf
set routing-instances VRF1 interface ge-0/0/5.0
set routing-instances VRF1 route-distinguisher 192.168.147.211:1
set routing-instances VRF1 vrf-import VPN-A-import
set routing-instances VRF1 vrf-export VPN-A-export
set routing-instances VRF1 vrf-table-label
set routing-instances VRF1 protocols ospf export bgp-to-ospf
set routing-instances VRF1 protocols ospf area 0.0.0.0 interface ge-0/0/5.0
```

PE2

```
set interfaces ge-0/0/0 unit 0 family inet address 10.10.12.2/24
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 10.10.23.2/24
set interfaces ge-0/0/1 unit 0 family mpls
set protocols mpls static-label-switched-path adj-23 segment 1000123
set protocols mpls static-label-switched-path adj-23 segment next-hop 10.10.23.3
set protocols mpls static-label-switched-path adj-23 segment pop
set protocols mpls static-label-switched-path adj-21 segment 1000221
set protocols mpls static-label-switched-path adj-21 segment next-hop 10.10.12.1
set protocols mpls static-label-switched-path adj-21 segment pop
set protocols mpls interface ge-0/0/0.0
set protocols mpls interface ge-0/0/1.0
set protocols mpls label-range static-label-range 1000000 1000999
```

```
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
```

PE3

```
set interfaces ge-0/0/0 unit 0 family inet address 10.10.13.3/24
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 10.10.23.3/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 family inet address 10.10.34.3/24
set interfaces ge-0/0/2 unit 0 family mpls
set protocols mpls static-label-switched-path adj-34 segment 1000134
set protocols mpls static-label-switched-path adj-34 segment next-hop 10.10.34.4
set protocols mpls static-label-switched-path adj-34 segment pop
set protocols mpls static-label-switched-path adj-32 segment 1000232
set protocols mpls static-label-switched-path adj-32 segment next-hop 10.10.23.2
set protocols mpls static-label-switched-path adj-32 segment pop
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols mpls label-range static-label-range 1000000 1000999
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
```

PE4

```
set interfaces ge-0/0/2 unit 0 family inet address 10.10.34.4/24
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 family inet address 10.10.45.4/24
set interfaces ge-0/0/3 unit 0 family mpls
set protocols mpls static-label-switched-path adj-45 segment 1000145
set protocols mpls static-label-switched-path adj-45 segment next-hop 10.10.45.5
set protocols mpls static-label-switched-path adj-45 segment pop
set protocols mpls static-label-switched-path adj-43 segment 1000243
set protocols mpls static-label-switched-path adj-43 segment next-hop 10.10.34.3
set protocols mpls static-label-switched-path adj-43 segment pop
set protocols mpls interface ge-0/0/2.0
set protocols mpls interface ge-0/0/3.0
set protocols mpls label-range static-label-range 1000000 1000999
set protocols ospf area 0.0.0.0 interface ge-0/0/2.0
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0
```

PE5

```
set interfaces ge-0/0/3 unit 0 family inet address 10.10.45.5/24
set interfaces ge-0/0/3 unit 0 family mpls maximum-labels 5
set interfaces ge-0/0/4 unit 0 family inet address 10.10.56.5/24
set routing-options autonomous-system 65000
set protocols mpls interface ge-0/0/3.0
set protocols mpls label-range static-label-range 1000000 1000999
set protocols bgp group pe type internal
set protocols bgp group pe local-address 192.168.146.181
set protocols bgp group pe family inet-vpn unicast
set protocols bgp group pe neighbor 192.168.147.211
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols bfd sbfd local-discriminator 0.0.0.32 minimum-receive-interval 1000
set protocols source-packet-routing segment-list sl-51 hop-1 ip-address 10.10.45.4
set protocols source-packet-routing segment-list sl-51 hop-2 label 1000243
set protocols source-packet-routing segment-list sl-51 hop-3 label 1000232
set protocols source-packet-routing segment-list sl-51 hop-4 label 1000221
set protocols source-packet-routing source-routing-path lsp-51 to 192.168.147.211
set protocols source-packet-routing source-routing-path lsp-51 primary sl-51
set policy-options policy-statement VPN-A-export term a from protocol ospf
set policy-options policy-statement VPN-A-export term a from protocol direct
set policy-options policy-statement VPN-A-export term a then community add VPN-A
set policy-options policy-statement VPN-A-export term a then accept
set policy-options policy-statement VPN-A-export term b then reject
set policy-options policy-statement VPN-A-import term a from protocol bgp
set policy-options policy-statement VPN-A-import term a from community VPN-A
set policy-options policy-statement VPN-A-import term a then accept
set policy-options policy-statement VPN-A-import term b then reject
set policy-options policy-statement bgp-to-ospf from protocol bgp
set policy-options policy-statement bgp-to-ospf from route-filter 10.10.0.0/16 orlonger
set policy-options policy-statement bgp-to-ospf then accept
set policy-options community VPN-A members target:65000:1
set routing-instances VRF1 instance-type vrf
set routing-instances VRF1 interface ge-0/0/4.0
set routing-instances VRF1 route-distinguisher 192.168.146.181:1
set routing-instances VRF1 vrf-import VPN-A-import
set routing-instances VRF1 vrf-export VPN-A-export
set routing-instances VRF1 vrf-table-label
```

```
set routing-instances VRF1 protocols ospf export bgp-to-ospf
set routing-instances VRF1 protocols ospf area 0.0.0.0 interface ge-0/0/4.0
```

CE1

```
set interfaces ge-0/0/0 unit 0 family inet address 10.10.17.7/24
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
```

CE2

```
set interfaces ge-0/0/4 unit 0 family inet address 10.10.56.6/24
set protocols ospf area 0.0.0.0 interface ge-0/0/4.0
```

Configuring Device PE1

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure Device PE1:

1. Configure the interfaces.

```
[edit interfaces]
set ge-0/0/0 unit 0 family inet address 10.10.12.1/24
set ge-0/0/0 unit 0 family mpls maximum-labels 5
set ge-0/0/1 unit 0 family inet address 10.10.13.1/24
set ge-0/0/1 unit 0 family mpls maximum-labels 5
set ge-0/0/5 unit 0 family inet address 10.10.17.1/24
```

2. Configure autonomous system number and options to control packet forwarding routing options.

```
[edit routing-options]
set autonomous-system 65000
set forwarding-table export load-balance-policy
set forwarding-table chained-composite-next-hop ingress l3vpn
```

3. Configure the interfaces with the MPLS protocol and configure the MPLS label range.

```
[edit protocols mpls]
set interface ge-0/0/0.0
set interface ge-0/0/1.0
set label-range static-label-range 1000000 1000999
```

4. Configure the type of peer group, local address, protocol family for NLRIs in updates, and IP address of a neighbor for the peer group.

```
[edit protocols bgp]
set group pe type internal
set group pe local-address 192.168.147.211
set group pe family inet-vpn unicast
set group pe neighbor 192.168.146.181
```

5. Configure the protocol area interfaces.

```
[edit protocols ospf]
set area 0.0.0.0 interface ge-0/0/0.0
set area 0.0.0.0 interface lo0.0
set area 0.0.0.0 interface ge-0/0/1.0
```

6. Configure the IPv4 address and labels of primary and secondary paths for source routing-traffic engineering (TE) policies of protocol source packet routing (SPRING).

```
[edit protocols source-packet-routing segment-list]
set sl-15-primary hop-1 ip-address 10.10.13.3
set sl-15-primary hop-2 label 1000134
set sl-15-primary hop-3 label 1000145
set sl-15-backup hop-1 ip-address 10.10.12.2
set sl-15-backup hop-2 label 1000123
set sl-15-backup hop-3 label 1000134
set sl-15-backup hop-4 label 1000145
```

7. Configure destination IPv4 address, binding SID label, primary, and secondary source routing path for protocol SPRING.

```
[edit protocols source-packet-routing source-routing-path]
set lsp-15 to 192.168.146.181
set lsp-15 binding-sid 1000999
set lsp-15 primary sl-15-primary
set lsp-15 secondary sl-15-backup
```

8. Configure policy options.

```
[edit policy-options policy-statement]
set VPN-A-export term a from protocol ospf
set VPN-A-export term a from protocol direct
set VPN-A-export term a then community add VPN-A
set VPN-A-export term a then accept
set VPN-A-export term b then reject
set VPN-A-import term a from protocol bgp
set VPN-A-import term a from community VPN-A
set VPN-A-import term a then accept
set VPN-A-import term b then reject
set bgp-to-ospf from protocol bgp
set bgp-to-ospf from route-filter 10.10.0.0/16 orlonger
set bgp-to-ospf then accept
set load-balance-policy then load-balance per-packet
```

9. Configure BGP community information.

```
[edit policy-options]
set community VPN-A members target:65000:1
```

10. Configure routing instance VRF1 with instance type, interface, router distinguisher, VRF import, export and table label. Configure export policy and interface of area for protocol OSPF.

```
[edit routing-instances VRF1]
set instance-type vrf
set interface ge-0/0/5.0
set route-distinguisher 192.168.147.211:1
set vrf-import VPN-A-import
```

```
set vrf-export VPN-A-export
set vrf-table-label
set protocols ospf export bgp-to-ospf
set protocols ospf area 0.0.0.0 interface ge-0/0/5.0
```

Results

From configuration mode, confirm your configuration by entering the **show interfaces**, **show policy-options**, **show protocols**, **show routing-options**, and **show routing-instances** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@PE1# show
   interfaces {
    ge-0/0/0 {
        unit 0 {
            family inet {
                address 10.10.12.1/24;
            }
            family mpls {
                maximum-labels 5;
            }
        }
    }
    ge-0/0/1 {
        unit 0 {
            family inet {
                address 10.10.13.1/24;
            }
            family mpls {
                maximum-labels 5;
            }
        }
    }
    ge-0/0/5 {
        unit 0 {
            family inet {
                address 10.10.17.1/24;
            }
        }
```

```
    }
}
policy-options {
    policy-statement VPN-A-export {
        term a {
            from protocol [ ospf direct ];
            then {
                community add VPN-A;
                accept;
            }
        }
        term b {
            then reject;
        }
    }
    policy-statement VPN-A-import {
        term a {
            from {
                protocol bgp;
                community VPN-A;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    policy-statement bgp-to-ospf {
        from {
            protocol bgp;
            route-filter 10.10.0.0/16 orlonger;
        }
        then accept;
    }
    policy-statement load-balance-policy {
        then {
            load-balance per-packet;
        }
    }
    community VPN-A members target:65000:1;
}
routing-instances {
    VRF1 {
```

```
        instance-type vrf;
        protocols {
            ospf {
                area 0.0.0.0 {
                    interface ge-0/0/5.0;
                }
                export bgp-to-ospf;
            }
        }
        interface ge-0/0/5.0;
        route-distinguisher 192.168.147.211:1;
        vrf-import VPN-A-import;
        vrf-export VPN-A-export;
        vrf-table-label;
    }
}
routing-options {
    autonomous-system 65000;
    forwarding-table {
        export load-balance-policy;
        chained-composite-next-hop {
            ingress {
                l3vpn;
            }
        }
    }
}
protocols {
    bgp {
        group pe {
            type internal;
            local-address 192.168.147.211;
            family inet-vpn {
                unicast;
            }
            neighbor 192.168.146.181;
        }
    }
    mpls {
        label-range {
            static-label-range 1000000 1000999;
        }
        interface ge-0/0/0.0;
```

```
        interface ge-0/0/1.0;
    }
    ospf {
        area 0.0.0.0 {
            interface ge-0/0/0.0;
            interface lo0.0;
            interface ge-0/0/1.0;
        }
    }
    source-packet-routing {
        segment-list sl-15-primary {
            hop-1 ip-address 10.10.13.3;
            hop-2 label 1000134;
            hop-3 label 1000145;
        }
        segment-list sl-15-backup {
            hop-1 ip-address 10.10.12.2;
            hop-2 label 1000123;
            hop-3 label 1000134;
            hop-4 label 1000145;
        }
        source-routing-path lsp-15 {
            to 192.168.146.181;
            binding-sid 1000999;
            primary {
                sl-15-primary;
            }
            secondary {
                sl-15-backup;
            }
        }
    }
}
```

Configuring Device PE2

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

1. Configure the interfaces.

```
[edit interfaces]
set ge-0/0/0 unit 0 family inet address 10.10.12.2/24
set ge-0/0/0 unit 0 family mpls
set ge-0/0/1 unit 0 family inet address 10.10.23.2/24
set ge-0/0/1 unit 0 family mpls
```

2. Configure the static LSP for protocol MPLS.

```
[edit protocols mpls static-label-switched-path]
set adj-23 segment 1000123
set adj-23 segment next-hop 10.10.23.3
set adj-23 segment pop
set adj-21 segment 1000221
set adj-21 segment next-hop 10.10.12.1
set adj-21 segment pop
```

3. Configure interfaces and static label range for protocol MPLS.

```
[edit protocols mpls]
set interface ge-0/0/0.0
set interface ge-0/0/1.0
set label-range static-label-range 1000000 1000999
```

4. Configure interfaces for protocol OSPF.

```
[edit protocols ospf area 0.0.0.0]
set interface ge-0/0/0.0
set interface ge-0/0/1.0
```

Results

From configuration mode on router PE2, confirm your configuration by entering the **show interfaces** and **show protocols** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@PE2# show
interfaces {
    ge-0/0/0 {
        unit 0 {
            family inet {
                address 10.10.12.2/24;
            }
            family mpls;
        }
    }
    ge-0/0/1 {
        unit 0 {
            family inet {
                address 10.10.23.2/24;
            }
            family mpls;
        }
    }
}
protocols {
    mpls {
        label-range {
            static-label-range 1000000 1000999;
        }
        interface ge-0/0/0.0;
        interface ge-0/0/1.0;
        static-label-switched-path adj-23 {
            segment {
                1000123;
                next-hop 10.10.23.3;
                pop;
            }
        }
        static-label-switched-path adj-21 {
            segment {
```

```
                1000221;
                next-hop 10.10.12.1;
                pop;
            }
        }
    }
    ospf {
        area 0.0.0.0 {
            interface ge-0/0/0.0;
            interface ge-0/0/1.0;
        }
    }
}
```

Verification

IN THIS SECTION

- Verifying Route Entry of Routing Table inet.3 of Router PE1 | **2268**
- Verifying Route Table Entries of Routing Table mpls.0 of Router PE1 | **2269**
- Verifying SPRING Traffic Engineered LSP of Router PE1 | **2270**
- Verifying SPRING Traffic Engineered LSPs on the Ingress Router of Router PE1 | **2270**
- Verifying the Routing Table Entries of Routing Table mpls.0 of Router PE2 | **2272**
- Verifying the Status of Static MPLS LSP Segments of Router PE2 | **2272**

Confirm that the configuration is working properly.

Verifying Route Entry of Routing Table inet.3 of Router PE1

Purpose

Verify the route entry of routing table inet.3 of router PE1.

Action

From operational mode, enter the `show route table inet.3` command.

```
user@PE1> show route table inet.3
inet.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.146.181/32   *[SPRING-TE/8] 03:09:26, metric 1
                     > to 10.10.13.3 via ge-0/0/1.0, Push 1000145, Push 1000134(top)
                       to 10.10.12.2 via ge-0/0/0.0, Push 1000145, Push 1000134, Push 1000123(top)
```

Meaning

The output displays the ingress routes of segment routing tunnels.

Verifying Route Table Entries of Routing Table mpls.0 of Router PE1

Purpose

Verify the route entries of routing table mpls.0

Action

From operational mode, enter the `show route table mpls.0` command.

```
user@PE1> show route table mpls.0
mpls.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 03:25:52, metric 1
                      Receive
1                  *[MPLS/0] 03:25:52, metric 1
                      Receive
2                  *[MPLS/0] 03:25:52, metric 1
                      Receive
13                 *[MPLS/0] 03:25:52, metric 1
                      Receive
16                 *[VPN/0] 03:25:52
                    > via lsi.0 (VRF1), Pop
```

```
1000999            *[SPRING-TE/8] 03:04:03, metric 1
                    > to 10.10.13.3 via ge-0/0/1.0, Swap 1000145, Push 1000134(top)
                      to 10.10.12.2 via ge-0/0/0.0, Swap 1000145, Push 1000134, Push 1000123(top)
```

Meaning

The output displays the SID labels of segment routing tunnels.

Verifying SPRING Traffic Engineered LSP of Router PE1

Purpose

Verify SPRING traffic engineered LSPs on the ingress routers.

Action

From operational mode, enter the `show spring-traffic-engineering overview` command.

```
user@PE1> show spring-traffic-engineering overview
Overview of SPRING-TE:
  Route preference: 8
  Number of LSPs: 1 (Up: 1, Down: 0)
  External controllers:
            < Not configured >
```

Meaning

The output displays the overview of SPRING traffic engineered LSPs on the ingress router.

Verifying SPRING Traffic Engineered LSPs on the Ingress Router of Router PE1

Purpose

Verify SPRING traffic engineered LSPs on the ingress router.

Action

From operational mode, enter the `show spring-traffic-engineering lsp detail` command.

```
user@PE1# show spring-traffic-engineering lsp detail
Name: lsp-15
To: 192.168.146.181
State: Up
  Path: sl-15-primary
  Outgoing interface: ge-0/0/1.0
  BFD status: N/A (Up: 0, Down: 0)
  SR-ERO hop count: 3
    Hop 1 (Strict):
      NAI: IPv4 Adjacency ID, 0.0.0.0 -> 10.10.13.3
      SID type: None
    Hop 2 (Strict):
      NAI: None
      SID type: 20-bit label, Value: 1000134
    Hop 3 (Strict):
      NAI: None
      SID type: 20-bit label, Value: 1000145
  Path: sl-15-backup
  Outgoing interface: ge-0/0/0.0
  BFD status: N/A (Up: 0, Down: 0)
  SR-ERO hop count: 4
    Hop 1 (Strict):
      NAI: IPv4 Adjacency ID, 0.0.0.0 -> 10.10.12.2
      SID type: None
    Hop 2 (Strict):
      NAI: None
      SID type: 20-bit label, Value: 1000123
    Hop 3 (Strict):
      NAI: None
      SID type: 20-bit label, Value: 1000134
    Hop 4 (Strict):
      NAI: None
      SID type: 20-bit label, Value: 1000145


Total displayed LSPs: 1 (Up: 1, Down: 0)
```

Meaning

The output displays details of SPRING traffic engineered LSPs on the ingress router

Verifying the Routing Table Entries of Routing Table mpls.0 of Router PE2

Purpose

Verify the routing table entries of routing table mpls.0 of router PE2.

Action

From operational mode, enter the `show route table mpls.0` command.

```
user@PE2> show route table mpls.0
mpls.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 03:22:29, metric 1
                      Receive
1                  *[MPLS/0] 03:22:29, metric 1
                      Receive
2                  *[MPLS/0] 03:22:29, metric 1
                      Receive
13                 *[MPLS/0] 03:22:29, metric 1
                      Receive
1000123            *[MPLS/6] 03:22:29, metric 1
                    > to 10.10.23.3 via ge-0/0/1.0, Pop
1000123(S=0)       *[MPLS/6] 03:22:29, metric 1
                    > to 10.10.23.3 via ge-0/0/1.0, Pop
1000221            *[MPLS/6] 03:22:29, metric 1
                    > to 10.10.12.1 via ge-0/0/0.0, Pop
1000221(S=0)       *[MPLS/6] 03:22:29, metric 1
                    > to 10.10.12.1 via ge-0/0/0.0, Pop
```

Verifying the Status of Static MPLS LSP Segments of Router PE2

Purpose

Verify the status of MPLS LSP segments of router PE2.

Action

From operational mode, enter the `show mpls static-lsp` command.

```
user@PE2> show mpls static-lsp
Ingress LSPs:
Total 0, displayed 0, Up 0, Down 0

Transit LSPs:
Total 0, displayed 0, Up 0, Down 0

Bypass LSPs:
Total 0, displayed 0, Up 0, Down 0

Segment LSPs:
LSPname                              SID-label       State
adj-21                               1000221         Up
adj-23                               1000123         Up
Total 2, displayed 2, Up 2, Down 0
```

Meaning

The output displays the status of static MPLS LSP segments of router PE2.

Enabling Distributed CSPF for Segment Routing LSPs

IN THIS SECTION

- Distributed CSPF Computation Constraints | **2274**
- Distributed CSPF Computation Algorithm | **2275**
- Distributed CSPF Computation Database | **2275**
- Configuring Distributed CSPF Computation Constraints | **2275**
- Distributed CSPF Computation | **2277**
- Interaction Between Distributed CSPF Computation and SR-TE Features | **2277**
- Distributed CSPF Computation Sample Configurations | **2278**

With the distributed Constrained Shortest Path First (CSPF) for segment routing LSP feature, you can compute a segment routing LSP locally on the ingress device according to the constraints you have configured. With this feature, the LSPs are optimized based on the configured constraints and metric type (traffic-engineering or IGP). The LSPs are computed to utilize the available ECMP paths to the destination with segment routing label stack compression enabled or disabled.

Use Feature Explorer to confirm platform and release support for specific features.

Review the "Platform-Specific Segment Routing LSP Behavior" on page 925 section for notes related to your platform.

Distributed CSPF Computation Constraints

Segment routing LSP paths are computed when all the configured constraints are met.

The distributed CSPF computation feature supports the following subset of constraints specified in the Internet draft, draft-ietf-spring-segment-routing-policy-03.txt, *Segment Routing Policy for Traffic Engineering*:

- Inclusion and exclusion of administrative groups.
- Inclusion of loose or strict hop IP addresses.

  NOTE: You can specify only router IDs in the loose or strict hop constraints. Labels and other IP addresses cannot be specified as loose or strict hop constraints in Junos OS Release 19.2R1-S1.

- Maximum number of segment IDs (SIDs) in the segment list.
- Maximum number of segment lists per candidate segment routing path.

The distributed CSPF computation feature for segment routing LSPs does not support the following types of constraints and deployment scenarios:

- IPV6 addresses.
- Inter domain segment routing traffic engineering (SR-TE) LSPs.
- Unnumbered interfaces.
- Multiple protocols routing protocols such as, OSPF, ISIS, and BGP-LS, enabled at the same time.
- Computation with prefixes or anycast addresses as destinations.
- Including and excluding interface IP addresses as constraints.

Distributed CSPF Computation Algorithm

The distributed CSPF computation feature for segment routing LSPs uses the label stack compression algorithm with CSPF.

Label Stack Compression Enabled

A compressed label stack represents a set of paths from a source to a destination. It generally consists of node SIDs and adjacency SIDs. When label stack compression is enabled, the result of the computation is a set of paths that maximize ECMP to the destination, with minimum number of SIDs in the stack, while conforming to constraints.

Label Stack Compression Disabled

The multipath CSPF computation with label stack compression disabled finds up to *N* segment lists to destination, where:

- The cost of all segment lists is equal to and the same as the shortest traffic-engineering metric to reach the destination.
- Each segment list is comprised of adjacency SIDs.
- The value of *N* is the maximum number of segment lists allowed for the candidate path by configuration.
- No two segment lists are identical.
- Each segment list satisfies all the configured constraints.

Distributed CSPF Computation Database

The database used for SR-TE computation has all links, nodes, prefixes and their characteristics irrespective of whether traffic-engineering is enabled in those advertising nodes. In other words, it is the union of the traffic-engineering database (TED) and the IGP link state database of all domains that the computing node has learnt from. As a result, for CSPF to work, you must include the `igp-topology` statement at the `[edit protocols isis traffic-engineering]` hierarchy level.

Configuring Distributed CSPF Computation Constraints

You can use a compute profile to logically group the computation constraints. These compute profiles are referenced by the segment routing paths for computing the primary and secondary segment routing LSPs.

To configure a compute profile, include the *compute-profile* statement at the `[edit protocols source-packet-routing]` hierarchy level.

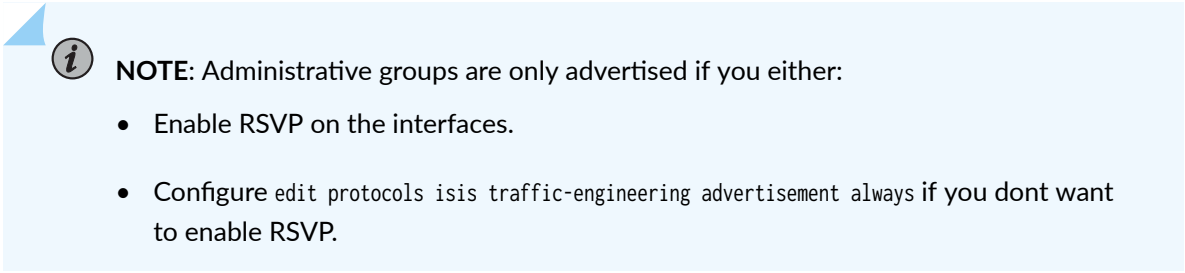
The configuration for the supported computation constraints include:

- **Administrative groups**

  You can configure *admin-groups* under the `[edit protocols mpls]` hierarchy level. Junos OS applies the administrative group configuration to the segment routing traffic-engineering (SR-TE) interfaces.

  To configure the computation constraints you can specify three categories for a set of administrative groups. The computation constraint configuration can be common to all candidate segment routing paths, or it can be under individual candidate paths.

  - `include-any`—Specifies that any link with at least one of the configured administrative groups in the list is acceptable for the path to traverse.
  - `include-all`—Specifies that any link with all of the configured administrative groups in the list is acceptable for the path to traverse.
  - `exclude`—Specifies that any link which does not have any of the configured administrative groups in the list is acceptable for the path to traverse.

  NOTE: Administrative groups are only advertised if you either:

  - Enable RSVP on the interfaces.
  - Configure `edit protocols isis traffic-engineering advertisement always` if you dont want to enable RSVP.

- **Explicit path**

  You can specify a series of router IDs in the compute profile as a constraint for computing the SR-TE candidate paths. Each hop has to be an IPv4 address and can be of type strict or loose. If the type of a hop is not configured, strict is used. You must include the `compute` option under the *segment-list* statement when specifying the explicit path constraint.

- **Maximum number of segment lists (ECMP paths)**

  You can associate a candidate path with a number of dynamic segment-lists. The paths are ECMP paths, where each segment-list translates into a next hop gateway with active weight. These paths are a result of path computation with or without compression.

  You can configure this attribute using the `maximum-computed-segment-lists` *`maximum-computed-segment-lists`* option under the *compute-profile* configuration statement. This configuration determines the maximum number of such segment lists computed for a given primary and secondary LSP.

- **Maximum segment list depth**

  The maximum segment list depth computation parameter ensures that amongst the ECMP paths that satisfy all other constraints such as administrative group, only the paths that have segment lists less than or equal to the maximum segment list depth are used. When you configure this parameter as a constraint under the compute-profile, it overrides the `maximum-segment-list-depth` configuration under the `[edit protocols source-packet-routing]` hierarchy level, if present.

  You can configure this attribute using the `maximum-segment-list-depth` *`maximum-segment-list-depth`* option under the *compute-profile* configuration statement.

- **Protected or unprotected adjacency SIDs**

  You can configure protected or unprotected adjacency SID as a constraint under the *compute-profile* to avoid links with the specified SID type.

- **Metric type**

  You can specify the type of metric on the link to be used for computation. By default, SR-TE LSPs use traffic-engineering metrics of the links for computation. The traffic-engineering metric for links is advertised by traffic-engineering extensions of IGP protocols. However, you may also choose to use the IGP-metric for computation by using the metric-type configuration in the compute profile.

  You can configure this attribute using the `metric-type (`*`igp | te`*`)` option under the *compute-profile* configuration statement.

Distributed CSPF Computation

The SR-TE candidate paths are computed locally such that they satisfy the configured constraints. When label stack compression is disabled, the multi-path CSPF computation result is a set of adjacency SID stacks. When label stack compression is enabled, the result is a set of compressed label stacks (composed of adjacent SIDs and node SIDs).

When secondary paths are computed, the links, nodes and SRLGs taken by the primary paths are not avoided for computation. For more information on primary and secondary paths, see "Configuring Primary and Secondary LSPs" on page 683.

For any LSPs with unsuccessful computation result, the computation is retried as traffic-engineering database (TED) changes.

Interaction Between Distributed CSPF Computation and SR-TE Features

Weights Associated With Paths of an SR-TE Policy

You can configure weights against computed and static SR-TE paths, which contribute to the next hops of the route. However, a single path that has computation enabled can result in multiple segment lists.

These computed segment lists are treated as ECMP amongst themselves. You can assign hierarchical ECMP weights to these segments, considering the weights assigned to each of the configured primaries.

BFD Liveliness Detection

You can configure BFD liveliness detection for the computed primary or secondary paths. Every computed primary or secondary path can result in multiple segment lists, as a result, the BFD parameters configured against the segment lists are applied to all the computed segment lists. If all the active primary paths go down, the pre-programmed secondary path (if provided) becomes active.

inherit-label-nexthops

You are not required to explicitly enable the `inherit-label-nexthops` configuration under the `[edit protocols source-packet-routing segment-list` *`segment-list-name`*`]` hierarchy for the computed primary or secondary paths, as it is a default behavior.

Auto-Translate Feature

You can configure the auto-translate feature on the segment lists, and the primary or secondary paths with the auto-translate feature reference these segment lists. On the other hand, the primary or secondary on which compute feature is enabled cannot reference any segment list. As a result, you cannot enable both the compute feature and the auto-translate feature for a given primary or secondary path. However, you could have an LSP configured with a primary path with compute type and another with auto-translate type.

Distributed CSPF Computation Sample Configurations

Example 1

In Example 1,

- The non-computed primary path references a configured segment-list. In this example, the configured segment list *static_sl1* is referenced, and it also serves as the name for this primary path.
- A computed primary should have a name configured, and this name should not reference any configured segment list. In this example, *compute_segment1* is not a configured segment list.
- The *compute_profile_red* compute-profile is applied to the primary path with the name *compute_segment1*.

- The *compute_profile_red* compute-profile includes a segment list of type `compute`, which is used to specify the explicit path constraint for the computation.

```
[edit protocols source-packet-routing]
segment-list static_sl1{
    hop1 label 80000
}
segment-list exp_path1 {
    hop1 ip-address 10.1.1.1 loose
    hop2 ip-address 10.2.2.2
    compute
}
compute-profile compute_profile_red {
    include-any red
    segment-list exp_path1
    maximum-segment-list-depth 5
}
```

The weights for computed path next-hops and static next-hops are 2 and 3, respectively. Assuming the next-hops for computed paths are *comp_nh1*, *comp_nh2*, and *comp_nh3*, and the next-hop for static path is *static_nh*, the weights are applied as follows:

Next-Hop	Weight
comp_nh1	2
comp_nh2	2
comp_nh3	2
static_nh	9

Example 2

In Example 2, both the primary and secondary paths can be of compute type and can have their own compute-profiles.

```
[edit protocols source-packet-routing]
compute-profile compute_profile_green{
    include-any green
    maximum-segment-list-depth 5
}
compute-profile compute_profile_red{
    include-any red
    maximum-segment-list-depth 8
}
```

Example 3

In Example 3, when compute is mentioned under a primary or secondary path, it results in local computation of a path to the destination without any constraints or other parameters for the computation.

```
[edit protocols source-packet-routing]
source-routing-path srte_colored_policy1 {
    to 10.5.5.5
    color 5
    binding-sid 10001
    primary {
        compute_segment1 {
            compute
        }
    }
}
```

Example: Configuring CoS-Based Forwarding and Policy-Based Routing For SR-TE LSPs

SUMMARY

CoS-based forwarding (CBF) and policy-based routing (PBR, also known as filter-based forwarding) can be enabled for non-colored segment routing-traffic engineered (SR–TE) LSPs to steer selective traffic over an explicit SR-TE path, providing you the benefit of servicing traffic based on class-of-service or a policy.

IN THIS SECTION

- CoS-Based Forwarding and Policy-Based Routing For SR-TE LSPs Overview | **2281**
- Configure CoS-Based Forwarding and Policy-Based Routing for SR-TE LSPs | **2282**

CoS-Based Forwarding and Policy-Based Routing For SR-TE LSPs Overview

IN THIS SECTION

- Benefits of CoS-Based Forwarding (CBF) and Policy-Based Routing (PBR) For SR-TE LSPs | **2281**
- Segment Routing Path Sources Supporting CBF and PBR | **2281**
- Considerations for Configuring CBF and PBR for SR-TE LSPs | **2282**

Benefits of CoS-Based Forwarding (CBF) and Policy-Based Routing (PBR) For SR-TE LSPs

With CBF and PBR you can:

- Use combinations of segment routing-traffic engineering (SR-TE) paths to steer service traffic in the core.
- Choose the supporting services to resolve over the selected SR-TE paths.

Segment Routing Path Sources Supporting CBF and PBR

The following segment routing path sources support CoS-based forwarding and policy-based routing:

- **Static SR–TE paths**—Statically configured source-routing paths that have the entire label stack statically configured.
- **PCEP**—Dynamically provisioning source-routing paths created on a controller, and downloaded to an ingress router in an ERO either through PCEP segment routing extensions, or in a BGP segment routing policy through BGP segment routing extensions.

- **Dynamic LSPs**—Dynamically created tunnels triggered through the Dynamic Tunnel Module that have last-hop ERO resolution.
- **Auto-translated paths**—Statically configured source-routing paths that are automatically translated.

Considerations for Configuring CBF and PBR for SR-TE LSPs

Remember:

- CBF and PBR is enabled only on non-colored SR-TE LSPs that are either statically or dynamically configured.
- Both CBF and PBR configurations for SR-TE LSPs can co-exist on a device; the order of configuration decides the type in which the routes are forwarded.
- For PBR, if the first-hop of the SR-TE LSP is a label, then you must include the `resolution preserve-nexthop-hiearchy` statement at the `[edit routing-options]` hierarchy level.
- The class-based forwarding of routes for CBF is visible only in the forwarding table and not on the routes.
- The policy-based forwarding of routes for PBR is done on the routes and is seen in the `show route` command output.

Configure CoS-Based Forwarding and Policy-Based Routing for SR-TE LSPs

SUMMARY

CoS-based forwarding (CBF) and policy-based routing (PBR, also known as filter-based forwarding FBF) can be used to steer selective traffic using an explicit segment routing-traffic engineered (SR-TE) label-swtiched path (LSP). Only non-colored segment routing LSPs that have the next hop configured as first hop label or IP address support CBF and PBR .

Before You Begin

- You must be running Junos OS Release 20.1 and later releases to enable CBF and PBR for non-colored SR-TE LSPs.
- Configure the device interfaces and ensure the devices are connected to the network.
- Define segment lists and configure SR-TE LSPs and their associated parameters.

To configure an SR-TE LSP, do the following:

1. Define the segment list with label parameters.

```
[edit protocol]
user@host# set source-packet-routing segment-list segment-list-name hop-name ip-address ip-address
user@host# set source-packet-routing segment-list segment-list-name hop-name label number
```

For example:

```
[edit protocol]
user@host# set source-packet-routing segment-list sr1 hop1 ip-address 11.1.1.2
user@host# set source-packet-routing segment-list sr1 hop2 label 801002
user@host# set source-packet-routing segment-list sr1 hop3 label 801003
user@host# set source-packet-routing segment-list sr1 hop4 label 801007
user@host# set source-packet-routing segment-list sr1 hop1 ip-address 11.1.1.2
user@host# set source-packet-routing segment-list sr1 hop2 label 801002
user@host# set source-packet-routing segment-list sr1 hop3 label 801003
user@host# set source-packet-routing segment-list sr1 hop4 label 801007
user@host# set source-packet-routing segment-list sr2 hop1 ip-address 11.11.1.2
user@host# set source-packet-routing segment-list sr2 hop2 label 801002
user@host# set source-packet-routing segment-list sr2 hop3 label 801003
user@host# set source-packet-routing segment-list sr2 hop4 label 801007
user@host# set source-packet-routing segment-list sr3 hop1 ip-address 11.12.1.2
user@host# set source-packet-routing segment-list sr3 hop2 label 801002
user@host# set source-packet-routing segment-list sr3 hop3 label 801003
user@host# set source-packet-routing segment-list sr3 hop4 label 801007
user@host# set source-packet-routing segment-list sr4 hop1 ip-address 11.13.1.2
user@host# set source-packet-routing segment-list sr4 hop2 label 801002
user@host# set source-packet-routing segment-list sr4 hop3 label 801003
user@host# set source-packet-routing segment-list sr4 hop4 label 801007
```

2. Configure the source-routing path for the SR-TE LSPs and specify the preference value and primary segment for the path.

```
[edit protocols]
user@host# set source-packet-routing source-routing-path source-routing-path-name to destination-ip-address
user@host# set source-packet-routing source-routing-path source-routing-path-name preference preference
```

```
user@host# set source-packet-routing source-routing-path source-routing-path-name primary
primary-segment
```

For example:

```
[edit protocols]
user@host# set source-packet-routing source-routing-path srtelsp1 to 7.7.7.7
user@host# set source-packet-routing source-routing-path srtelsp1 preference 1
user@host# set source-packet-routing source-routing-path srtelsp1 primary sr1
user@host# set source-packet-routing source-routing-path srtelsp2 to 7.7.7.7
user@host# set source-packet-routing source-routing-path srtelsp2 preference 1
user@host# set source-packet-routing source-routing-path srtelsp2 primary sr2
user@host# set source-packet-routing source-routing-path srtelsp3 to 7.7.7.7
user@host# set source-packet-routing source-routing-path srtelsp3 preference 1
user@host# set source-packet-routing source-routing-path srtelsp3 primary sr3
user@host# set source-packet-routing source-routing-path srtelsp4 to 7.7.7.7
user@host# set source-packet-routing source-routing-path srtelsp4 preference 1
user@host# set source-packet-routing source-routing-path srtelsp4 primary sr4
```

You can now configure CBF and PBR for the configured SR-TE LSPs.

To configure CBF, do the following

1. Define Differentiated Services Code Point (DSCP) classifiers to handle incoming IPv4 packets, forwarding classes, and option values.

```
[edit class-of-service]
user@host# set classifiers dscpclassifier-name forwarding-class forwarding-class-name loss-
priority level code-points [ aliases ] [ 6 bit-patterns ]
```

For example:

```
[edit class-of-service]
user@host# set classifiers dscp mydscp forwarding-class af11 loss-priority low code-points
001010
user@host# set classifiers dscp mydscp forwarding-class af11 loss-priority medium-high code-
points 001100
user@host# set classifiers dscp mydscp forwarding-class af11 loss-priority high code-points
001110
user@host# set classifiers dscp mydscp forwarding-class af21 loss-priority low code-points
010010
```

```
user@host# set classifiers dscp mydscp forwarding-class af21 loss-priority medium-high code-
points 010100
user@host# set classifiers dscp mydscp forwarding-class af21 loss-priority high code-points
010110
user@host# set classifiers dscp mydscp forwarding-class af31 loss-priority low code-points
011010
user@host# set classifiers dscp mydscp forwarding-class af31 loss-priority medium-high code-
points 011100
user@host# set classifiers dscp mydscp forwarding-class af31 loss-priority high code-points
011110
user@host# set classifiers dscp mydscp forwarding-class af41 loss-priority low code-points
100010
user@host# set classifiers dscp mydscp forwarding-class af41 loss-priority medium-high code-
points 100100
user@host# set classifiers dscp mydscp forwarding-class af41 loss-priority high code-points
100110
```

2. Define forwarding classes (FCs) for grouping packets for transmission, and assign packets to output queues.

```
[edit class-of-service]
user@host# set forwarding-classes queue queue-numner class-name
```

For example:

```
[edit class-of-service]
user@host# set forwarding-classes queue 0 af11
user@host# set forwarding-classes queue 1 af21
user@host# set forwarding-classes queue 2 af31
user@host# set forwarding-classes queue 3 af41
```

3. Assign the configured classifiers to the device interfaces.

```
[edit class-of-service]
user@host# set interfaces interface-name unit unit classifiers dscp classifier-name
```

For example:

```
[edit class-of-service]
user@host# set interfaces ge-0/0/8 unit 1 classifiers dscp mydscp
user@host# set interfaces ge-0/0/8 unit 2 classifiers dscp mydscp
```

4. Define CoS-based forwarding policy options with LSP next-hop as the SR-TE LSP.

```
[edit class-of-service]
user@host# set forwarding-policy next-hop-map map-name forwarding-classes class-name lsp-next-hop source-routing-path-name
```

For example:

```
[edit class-of-service]
user@host# set forwarding-policy next-hop-map my_cbf forwarding-class af11 lsp-next-hop srtelsp1
user@host# set forwarding-policy next-hop-map my_cbf forwarding-class af21 lsp-next-hop srtelsp2
user@host# set forwarding-policy next-hop-map my_cbf forwarding-class af41 lsp-next-hop srtelsp3
user@host# set forwarding-policy next-hop-map my_cbf forwarding-class af31 lsp-next-hop srtelsp4
```

5. Discard traffic that does not meet any forwarding class in the next-hop map.

```
[edit class-of-service]
user@host# set forwarding-policy next-hop-map map-name forwarding-class-default discard
```

For example:

```
[edit class-of-service]
user@host# set forwarding-policy next-hop-map my_cbf forwarding-class-default discard
```

6. Configure a policy statement that specifies that routes matching the route filter are subject to the CoS next-hop mapping specified by map-name.

```
[edit policy-options]
user@host# set policy-statement policy-name from route-filter destination-prefix match-type
<actions>
user@host# set policy-statement policy-name then cos-next-hop-map map-name
```

For example:

```
[edit policy-options]
user@host# set policy-statement cbf from route-filter 4.0.0.1/16 orlonger
user@host# set policy-statement cbf then cos-next-hop-map my_cbf
```

7. Apply the policy to routes being exported from the routing table into the forwarding table. This enables CBF for SR-TE LSPs.

```
[edit routing-options]
user@host# set forwarding-table export policy-name
```

For example:

```
[edit routing-options]
user@host# set forwarding-table export cbf
```

8. Commit the configuration.

```
user@host# commit
```

Verify CBF Configuration

You can verify the CBF configuration using the `show route forwarding-table destination` *ip-address* `vpn` *vpn-name* `extensive` command.

```
user@host> show route forwarding-table destination 4.0.0.1 vpn vpn1 extensive
Routing table: vpn1.inet [Index 8]
Internet:
Destination:  4.0.0.1/32
```

```
  Route type: user
  Route reference: 0
  Multicast RPF nh index: 0
  P2mpidx: 0

Flags: sent to PFE
Next-hop type: indirect
Next-hop type: indexed
Route type: idx:0
Nexthop: 11.1.1.2
Next-hop type: Push 296, Push 801007, Push 801003, Push 801002(top) Index: 807

Reference: 1

Route interface-index: 0

Index: 1048579  Reference: 10001
Index: 837      Reference: 2
Load Balance Label: None
Next-hop interface: ge-0/0/1.1
Route type: idx:1
Nexthop: 11.11.1.2
Next-hop type: Push 296, Push 801007, Push 801003, Push 801002(top) Index: 809
```

For CBF, the class-based forwarding of routes is visible only in the forwarding table, unlike PBR, where the filtered routes are visible in the `show route` command output.

To configure PBR, do the following

1. Configure a policy statement which specifies that routes matching the protocol and route filter are subject to the LSP next-hop, or are load balanced as equal-cost multipath (ECMP) in the forwarding table.

```
[edit policy-options]
user@host# set policy-statement policy-name from protocol protocol-name
user@host# set policy-statement policy-name from route-filter destination-prefix match-type
<actions>
user@host# set policy-statement policy-name then install-nexthop lsp lsp-name
user@host# set policy-statement policy-name then load-balance per-packet
```

For example:

```
[edit policy-options]
user@host# set policy-statement pbr term 1 from protocol bgp
user@host# set policy-statement pbr term 1 from route-filter 4.0.0.1/32 exact
user@host# set policy-statement pbr term 1 then install-nexthop lsp srtelsp1
user@host# set policy-statement pbr term 1 then load-balance per-packet
user@host# set policy-statement pbr term 1 then reject
```

2. Configure the device to perform custom route resolution on protocol next hops of routes.

NOTE: The `resolution preserve-nexthop-hierarchy` statement is mandatory for PBR to work when the first-hop of the SR-TE LSP is a label.

```
[edit routing-options]
user@host# set resolution preserve-nexthop-hierarchy
```

3. Apply the policy to routes being exported from the routing table into the forwarding table. This enables PBR for SR-TE LSPs.

```
[edit routing-options]
user@host# set forwarding-table export policy-name
```

For example:

```
[edit routing-options]
user@host# set forwarding-table export pbr
```

4. Commit the configuration.

```
user@host# commit
```

Verify PBR Configuration

You can verify the PBR configuration using the `show route` *`destination-prefix`* command.

```
user@host> show route 4.0.0.1
vpn1.inet.0: 10003 destinations, 10003 routes (10003 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

4.0.0.1/32         *[BGP/170] 00:24:12, localpref 100, from 7.7.7.7
                      AS path: 10 I, validation-state: unverified
                       to 11.1.1.2 via ge-0/0/1.1, Push 50983, Push 801007, Push 801003, Push
801002(top)
                    >  to 11.11.1.2 via ge-0/0/1.2, Push 50983, Push 801007, Push 801003, Push
801002(top)
                       to 11.12.1.2 via ge-0/0/1.3, Push 50983, Push 801007, Push 801003, Push
801002(top)
                       to 11.13.1.2 via ge-0/0/1.4, Push 50983, Push 801007, Push 801003, Push
801002(top)
```

```
user@host> show route 4.0.0.1 expanded-nh extensive
vpn1.inet.0: 10003 destinations, 10003 routes (10003 active, 0 holddown, 0 hidden)
4.0.0.1/32 (1 entry, 1 announced)
Installed-nexthop:
Indr (0xc7aaa54) 7.7.7.7 Push 50983 Session-ID: 0x16f
  Krt_inh (0xc745a84) Index:1048579 PNH: 7.7.7.7
    Chain (0xc7aa798) Index:823 Push 50983
      Router (0xc417034) 11.1.1.2 Push 801007, Push 801003, Push 801002(top) via ge-0/0/1.1
```

The output displays all next-hops for the destination prefix, 4.0.0.1. The `expanded-nh extensive` options displays the filtered next-hops under the `Krt_inh` output field.

```
user@host> show route 4.0.0.2
vpn1.inet.0: 10003 destinations, 10003 routes (10003 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

4.0.0.2/32         *[BGP/170] 00:30:14, localpref 100, from 7.7.7.7
                      AS path: 10 I, validation-state: unverified
                       to 11.1.1.2 via ge-0/0/1.1, Push 569, Push 801007, Push 801003, Push
801002(top)
                    >  to 11.11.1.2 via ge-0/0/1.2, Push 569, Push 801007, Push 801003, Push
801002(top)
                       to 11.12.1.2 via ge-0/0/1.3, Push 569, Push 801007, Push 801003, Push
```

```
801002(top)
                      to 11.17.1.2 via ge-0/0/1.8, Push 569, Push 801007, Push 801003, Push
801002(top)
```

```
user@host> show route 7.7.7.7 protocol spring-te
inet.0: 10082 destinations, 10119 routes (10082 active, 0 holddown, 0 hidden)

inet.3: 25 destinations, 77 routes (25 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

7.7.7.7/32         *[SPRING-TE/1] 00:00:32, metric 1, metric2 4
                    >  to 11.1.1.2 via ge-0/0/1.1, Push 801007, Push 801003, Push 801002(top)
                       to 11.11.1.2 via ge-0/0/1.2, Push 801007, Push 801003, Push 801002(top)
                       to 11.12.1.2 via ge-0/0/1.3, Push 801007, Push 801003, Push 801002(top)
                       to 11.17.1.2 via ge-0/0/1.8, Push 801007, Push 801003, Push 801002(top)
```

For PBR the `show route` command output does the policy-based filtering of routes.

Enable Multiple Paths for SR-TE LSPs in PCEP

IN THIS SECTION

- Benefits of multiple paths for PCEP SR-TE LSPs | **2291**
- Limitations | **2293**
- Configuration | **2294**

You can configure multiple paths (primary or secondary) for PCEP SR-TE LSPs (statically configured, delegated, and PCE-initiated) as defined in *draft-ietf-pce-multipath-06*. Only one secondary path configuration is supported and only for statically configured SR-TE LSPs. The PCEP extensions defined in *draft-ietf-pce-multipath-06* enable PCEP to propagate multiple paths (multipath) for the LSPs between PCEP endpoints.

Benefits of multiple paths for PCEP SR-TE LSPs

- LSPs can have multiple sets of EROs to a destination
- Provides load balancing capabilities by configuring weights for individual EROs

- Aligns with SR-TE architecture draft for defining candidate paths

The following PCEP multiple path capabilities are supported:

- When PCEP for multiple paths is enabled (default), you can configure multiple primary (or one secondary) paths in a PCC configured and controlled candidate path.
- When PCEP for multiple paths is disabled, you can configure only one primary path in a candidate path. Secondary path configuration is not allowed.

If you enable PCEP multipaths, `compute-profile` can now be configured with maximum number of segment-lists (`maximum-computed-segment-lists`) greater than 1.

NOTE: When PCEP for multiple paths is enabled, PCCD will not send constraints for PCC controlled candidate paths.

When PCEP multipath capability is enabled, secondary path configuration is allowed for a non-delegated PCC candidate path, the EXPLICIT-ROUTE object (EROs) specific to the secondary path is sent to the PCE with backup flag set for the ERO. Primary paths do not include the MULTIPATH-BACKUP-TLV in the PCRpt message. Secondary path include MULTIPATH-BACKUP-TLV with backup flag set.

The following PCEP multipath functionalities are supported:

- Multipath weight TLV (MULTIPATH-WEIGHT-TLV) in path attribute (PATH-ATTRIB) object
- MULTIPATH-BACKUP TLV in path attribute (PATH-ATTRIB) object only for PCC controlled SR-TE LSPs
- MULTIPATH-CAP TLV in PCEP LSP object
- Restricts multiple primary and secondary paths in SR candidate path when PCEP multipath is disabled
- Multiple primary paths and secondary paths in SR candidate path when PCEP multipath is enabled for PCC controlled LSPs
- Maximum computed segment lists (`max-computed-segment-lists`) more than 1 in SR-TE compute profile for delegated and PCE-initiated LSPs
- Multiple EROs for PCE-initiated candidate path in SR-TE and in PCCD
- SRv6 LSPs
- SR MPLS (IPv4)
- SR MPLS (IPv4) dynamic tunnels
- Multi-controller support

- Multiple ERO paths for PCE-initiated, PCC-configured and controlled, and delegated colored and uncolored candidate paths
- Backward compatible with earlier versions of Paragon Pathfinder. For backward compatibility, you need to configure `disable-multipath-capability` configuration statement at the [`edit protocols pcep`] hierarchy level.
- Error code support for failure in validation of PCE-initiated candidate paths
  - Total sub-candidate paths per candidate path is limited to 127. For PCE initiated LSPs, if the number of ERO paths crosses 127, SR-TE throws ERROR to PCCD (and PCCD sends PCEP error message to PCE) and the corresponding ERO paths are rejected.

The following PCEP error messages are supported:

Table 50: PCEP Error Messages

Error type	Error value	Meaning	Usage
19	20	Backup path not supported	This occurs when MULTIPATH-BACKUP TLV is received by the PCC.
24	1	Unacceptable instantiation parameters	This occurs when PCE tries to add more than 127 sub-candidate paths per candidate path.

Limitations

The following PCEP limitations apply:

- The following TLVs mentioned in the *draft-ietf-pce-multipath-06* are not supported:
  - Multipath backup TLV
  - Multipath opposite direction path TLV
  - Composite candidate path
- When multipath capability is disabled in PCEP, configuring multiple sub-candidate paths is not allowed. However, on Junos devices without multipath capability (Junos OS versions earlier than 22.4R1), multiple sub-candidate path configuration is allowed. When PCEP multi-segment is enabled (by default), multiple primary paths are allowed for PCC controlled LSPs for the purpose of reporting.

However, only one primary path is supported for delegated candidate path when PCEP multi-segment is enabled.

- Admin groups and any other constraints will not be notified to PCE for PCC configured and controlled SR-MPLS and SRv6 candidate paths (with single or multiple primary configurations). There is no impact for delegated and PCE-initiated candidate paths.
- When PCEP multipath capability is enabled, secondary path configuration is allowed for non-delegated candidate paths. When the PCEP multipath capability is disabled, secondary path configuration is not allowed.
- Candidate paths cannot have a mix of PCE-initiated and delegated LSPs.
- Multiple sub-candidate paths for a PCE-initiated colored candidate path is not supported.
- Delegate features with multiple sub-candidate paths in a candidate path is not supported.

Configuration

To enable PCCD to send multipath capability TLV in LSP object to notify the maximum computed segment list for a specific candidate path, include the `propagate-max-segmentlist` configuration statement at the [`edit protocols pcep`] hierarchy level. By default, the TLV is not sent in LSP object.

```
user@host# set protocols pcep propagate-max-computed-segmentlist;
```

To disable PCEP multiple capability session for all PCEs, include the `disable-multipath-capability` configuration statement at the [`edit protocols pcep`] hierarchy level.

```
user@host# set protocols pcep disable-multipath-capability;
```

```
[edit protocols]
pcep {
    disable-multipath-capability;
    propagate-max-segmentlist;
  }
```

You can enable the following protocol traceoptions for diagnostics:

- `user@host# set protocols pcep traceoptions ...`
- `user@host# set protocols pcep pce pce1 traceoptions ...`

- `user@host# set protocols source-packet-routing traceoptions`

You can use the following show commands to display the state of LSPs in PCC:

- `user@host> show path-computation-client lsp`—Display the state of label-switched paths (LSPs) known to the Path Computation Client (PCC).
- `user@host> show path-computation-client lsp extensive`—Display extensive level of output about each known LSP - point-to-point and point-to-multipoint LSPs.
- `user@host> show path-computation active-pce`—Display the status of multipath in the sessions.
- `user@host> show spring-traffic-engineering lsp detail`—Display ingress details of SPRING traffic engineering.

Enable Transport Layer Security for PCEP Sessions

IN THIS SECTION

- Benefits of Enabling TLS for PCEP Sessions | **2295**
- Enabling TLS in Path Computation Client (PCC) | **2296**
- Updating Certificates Using the Public Key Infrastructure (PKI) | **2297**
- Establishing TLS Connection | **2297**
- Understanding the Basic TLS Handshake Mechanism | **2298**
- Diagnosing and Validating TLS for PCEP Sessions | **2299**
- Sample Output | **2300**

Transport Layer Security (TLS) provides support for peer authentication, message encryption, and integrity. You can enable TLS in Path Computation Client (PCC) to establish TCP connection with the Path Computation Element (PCE) as defined in RFC 8253. This creates a secure PCEP session (PCEPS) to transport PCEP messages.

This document describes how to enable TLS for PCEP sessions to secure interactions with PCE, including initiation of the TLS procedures, the TLS handshake mechanism, the TLS methods for peer authentication. Secure transport for PCEP over TLS is also known as PCEPS.

Benefits of Enabling TLS for PCEP Sessions

- Protects PCEP sessions from attacks such as spoofing (PCC or PCE impersonation), snooping (message interception), falsification, and denial of service.

- Leverages TLS security benefits.

Enabling TLS in Path Computation Client (PCC)

To enable TLS in PCC and to establish PCEPS session, set the `tls-strict` CLI statement at the [`edit protocols pcep`] hierarchy level.

After enabling tls-strict configuration statement, the following events happen:

1. PCEP session flaps. Any existing TCP connection is terminated and a reconnection is done using TLS.
2. PCC establishes TCP connection with the PCE.
3. TLS procedures are initiated by the **StartTLS** message from PCE to PCC and from PCC to PCE. The **StartTLS** message is sent by PCC and the **StartTLSWait** timer is started. You can configure the **StartTLSWait** timer by configuring the `start-tls-wait-timer` *seconds* CLI statement at the [`edit protocols pcep pce` *pce-id*] hierarchy level.

   > **NOTE**: The recommended value for the **StartTLSWait** timer is 60 seconds and must not be less than the **OpenWait** timer. The default **OpenWait** timer value is set as 60 seconds.

   - If Open message is received by PCC instead of **StartTLS** message, **PCErr** message with Error-Type set to 1 (PCEP session establishment failure) and Error-value set to 1 (reception of an invalid Open message or a non-Open message), and the TCP session is closed.
   - If **StartTLS** message is not received from PCE, then after the **StartTLSWait** timer expires, PCC sends a **PCErr** message with Error-Type set to 25 (PCEP **StartTLS** failure) and Error-value set to 5 (no **StartTLS** message (nor **PCErr/Open**) before **StartTLSWait** timer expiry), and the TCP session is closed.
4. Negotiation and establishment of TLS connection happens.
5. Exchange of PCEP messages is started as per RFC5440.

> **NOTE**: If you do not enable the `tls-strict` CLI statement under the [`edit protocols pcep`] hierarchy level, then while establishing a PCEP session, if the **StartTLS** message is received by PCC instead of **Open** message, **PCErr** message with Error-Type set to 1 (PCEP session establishment failure) and Error-value set to 1 (reception of an invalid Open message or a non-Open message), then the TCP session is closed.

NOTE: To establish a successful PCEPS session, TLS must be enabled on both PCC and PCE.

Updating Certificates Using the Public Key Infrastructure (PKI)

The PKI doesn't notify PCC about certificate expiry. You must manually update the certificate using the following CLI command. In this method, you must keep track of the certificate expiry date.

```
user@host> request security pki local-certificates re-enroll certificate id
```

Establishing TLS Connection

The following steps describes how a TLS connection (using TLS v1.2) is established:

1. Generate certificates for the nodes (Junos OS devices/pce-server). You can generate the certificates using one of the following methods:
   - **Method 1**—Generate key-pair and CSR on the device and send this CSR to CA to get the certificate. Once the certificate is issued, it is copied to the box and installed.
   - **Method 2**—Generate key-pair and the certificate out of the box. Both the certificate and the private key are copied on the device and installed together.
2. Load the certification authority (CA) on the PCC so that the PCE server certificate can be validated against the loaded CA.

```
user@host# set security pki ca-profile pccd-tls ca-identity pccd-tls
user@host# commit
```

```
user@host> request security pki ca-certificate load ca-profile pccd-tls filename /var/tmp/
ca.crt
```

NOTE: CAs can be loaded in flat hierarchy as an independent CA. If a CA is a sub-CA of another CA, the chain is constructed internally by PKI.

NOTE: The server certificate should be signed by a CA. Self-signed certificates are not allowed.

3. Enable TLS on PCC.
4. PCEP session is established over TLS with TLS handshake mechanism.
5. PCE server listens to port 4189 for incoming PCC connection requests through TLS.
6. PCC initiates the connection request to destination port 4189.
7. Upon completion of a three-way handshake, the TLS handshake begins by using the certificates and the one-way authentication is done (PCC authenticates server certificate). Both the server and client waits for **StartTLSWait** time to receive the **StartTLS** message. You can configure the **StartTLSWait** timer by configuring the `start-tls-wait-timer` *seconds* CLI statement at the [`edit protocols pcep pce pce-id`] hierarchy level.

NOTE: The recommended value for the **StartTLSWait** timer is 60 seconds and must not be less than the **OpenWait** timer. The default **OpenWait** timer value is set as 60 seconds.

8. After the successful TLS handshake session, PCC and PCE initiates PCEP session establishment over TLS during which the session parameters are negotiated.
   - If the certificate validation fails, then PCC terminates the TCP connection.
9. PCEP message is sent over TLS connection as application data.
10. Encryption and decryption happen on both PCC and PCE after a successful TLS handshake.
11. When the PCEP session is closed, the TLS session is removed.

NOTE: If the certificate is expired, revoked, or reloaded during an ongoing PCEP over TLS session, the ongoing session is not affected.

Understanding the Basic TLS Handshake Mechanism

The handshake is a series of the messages exchanged between a server and a client. The exact steps in handshake varies depending on key exchange algorithm, cipher suite, etc. The following are the basic TLS handshake mechanism steps:

1. **Client Hello**—The client initiates the handshake by sending this message. This message contains TLS version, list of supported cryptographic algorithms or cipher suite and other client details.
2. **Server Hello**—The server replies to Client Hello by sending Sever Hello message. This message contains server certificate, selected cryptographic algorithm, session id and server's public key.
3. **Authentication**—The client in background verifies the server's certificate with configured Certificate Authority which had issued the certificate. On successful verification the client confirms that the server is genuine and continues with interacting.
4. **Optional Client Certificate**—If the server has requested a certificate from the client in Server Hello message, then the client sends the client certificate (only in case of mutual TLS).
5. **Client Key Exchange**—The client sends secret Key encrypted with the Server's Public key (acquired in Server Hello message).
6. **Decrypt secret key**—The server decrypts the secret key using the Private key.
7. **Client Finished**—The client sends a finish message which is encrypted with the shared secret key and signals handshake complete.
8. **Server Finished**—The server responds with finish message which is encrypted with the shared secret key and signals handshake complete.
9. **Exchange Messages**—The messages after handshake completion are symmetrically encrypted.

Diagnosing and Validating TLS for PCEP Sessions

For diagnostics, use the following traceoptions CLI statements:

```
user@host# set protocols pcep traceoptions …
user@host# set protocols pcep pce pce-id traceoptions …
user@host# set protocols source-packet-routing traceoptions
```

Enable PKI logs using the following configuration and capture the same file from */var/log/<filename>*

```
user@host# set security pki traceoptions file <filename>
user@host# set security pki traceoptions flag all
sss
```

Verify the loaded CA certificate using the following command:

```
user@host> show security pki ca-certificate detail
```

Sample Output

The following is a sample output of `show path-computation-client statistics` command:

```
user@host> show path-computation-client statistics

Warning: License key missing; requires 'PCEP' license

PCE ns1
-----------------------------------------
General
    PCE IP address           : 192.168.18.1
    Local IP address         : 190.168.18.101
    Priority                 : 0
    PCE status               : PCE_STATE_UP
    Session type             : PCE_TYPE_STATEFULACTIVE
    LSP provisioning allowed : On
    P2MP LSP report allowed  : On
    P2MP LSP update allowed  : On
    P2MP LSP init allowed    : On
    Session SRv6 Capable     : No
    PCE-mastership           : main
    PCE Traffic Steering     : Off
    PCC TLS Enabled        : Yes
    PCE TLS Enabled        : Yes
    Session TLS Enabled        : Yes

Counters
    PCReqs              Total: 0           last 5min: 0           last hour: 0
    PCReps              Total: 0           last 5min: 0           last hour: 0
    PCRpts              Total: 0           last 5min: 0           last hour: 0
    PCUpdates           Total: 0           last 5min: 0           last hour: 0
    PCCreates           Total: 0           last 5min: 0           last hour: 0

Timers
    Local  Keepalive timer:    0 [s]  Dead timer:    0 [s]  LSP cleanup timer:    - [s]
```

```
    Remote Keepalive timer:    0 [s]  Dead timer:    0 [s]  LSP cleanup timer:    - [s]

Errors
    PCErr-recv
    PCErr-sent
    PCE-PCC-NTFS
    PCC-PCE-NTFS

Pcupdate empty ero action counters
    Send-err               : 0
    Tear down path         : 0
    Routing decision       : 0
    Routing decision failed: 0
```

This sample output provides the following information:

- TLS is enabled at PCC.
- PCE is TLS capable.
- TLS session is established. This also indicates that the PCE server certificate is valid.
- PCEPS session status is up and running.

Reporting Path Optimization and Computed Metrics in PCEP

IN THIS SECTION

- Benefits of Reporting Path Optimization and Computed Metrics in PCEP | **2302**
- Understanding Optimization Metrics | **2302**
- Configuring Optimization Metrics for LSPs | **2306**
- Sample Output | **2306**

Metric object in PCEP is used for several purposes. Metric object indicates the metric type that is used for path optimization. Metric object also indicates a bound on the path cost that must not be exceeded for the path to be considered as acceptable. Metric object also indicates the computed metric.

We support metric object for path optimization (interior gateway protocol, traffic engineering, and path delay) and reporting of computed metric for RSVP and SR-TE LSPs.

NOTE: Metric object for path optimization and reporting of computed metric is not applicable for SRv6-TE LSPs.

Benefits of Reporting Path Optimization and Computed Metrics in PCEP

- Reporting of path optimization metrics configured in PCC helps PCE to be aware of the constraints that is used for path computation.
- Reporting of computed metrics to the PCE. This helps PCE to analyze if the LSP requires further optimization.

Understanding Optimization Metrics

The following section describes the intended and actual optimization metrics for RSVP and and SR-TE (SR MPLS) LSPs in PCEP.

Locally Created RSVP LSP

To optimize the locally created RSVP LSPs with metrics, configure the optimization metrics (IGP, TE, and path delay) so that the configured metric is reported through PCEP. The computed metric is sent as an actual metric in PCEP through the PCRpt message.

Delegated RSVP LSP

To report the optimization metrics for delegated RSVP LSPs, configure the optimization metrics (IGP, TE, and path delay).

Intended Metric:

- When the optimization metric is configured at the time of delegation of LSP, the information is sent to PCE through PCRpt message.
- When the optimization metric is configured after the delegation of LSP, the change is applied on the LSP/communicated to the PCE when the LSP control status becomes locally controlled.
- When PCUpd message is received, if optimization metric is present in the message, the metric is used as intended metric in subsequent PCRpt messages till the LSP control status is externally controlled.
- When PCUpd message is received, if optimization metric is not present in the message, subsequent PCRpt messages do not contain intended metric.
- When the LSP control status changes to locally controlled, optimization metric configured from Junos CLI will be the intended metric in the PCRpt message.

Actual Metric:

- While delegating the LSP, PCRpt message do not contain actual metric.
- When PCUpd message is received, if computed metric is present in the message, the metric is used as actual metric in subsequent PCRpt messages till the LSP control status is externally controlled.
- When PCUpd message is received, if computed metric is not present in the message, subsequent PCRpt messages do not contain actual metric.
- When the LSP control status changes to locally controlled, metric computed by PCC is sent as actual metric in the PCRpt message.

PCE-initiated RSVP LSP

To report the optimization metrics for PCE-initiated RSVP LSPs, configure the optimization metrics (IGP, TE, and path delay) in a template. The template is then applied to the PCE-initated LSP when the LSP control status becomes locally controlled.

Intended Metric:

- When a PCE-initiated LSP is mapped to a template with optimization metric, the configuration is applied to the LSP and sent to the PCE when the LSP control status changes to locally controlled.
- When PCInit/PCUpd message is received, if optimization metric is present in the message, the metric is used as intended metric in subsequent PCRpt messages till the LSP control status is externally controlled.
- When PCInit/PCUpd message is received, if optimization metric is not present in the message, subsequent PCRpt messages do not contain intended metric.
- When the LSP control status becomes locally controlled, optimization metric present in the template is used as intended metric in the PCRpt message.

Actual Metric:

- When PCInit/PCUpd message is received, if the computed metric is present in the message, the metric is used as actual metric in subsequent PCRpt messages till the LSP control status is externally controlled.
- When PCInit/PCUpd message is received, if the computed metric is not present in the message, subsequent PCRpt messages do not contain actual metric.
- When the LSP control status changes to locally controlled, metric computed by PCC is sent as actual metric in the PCRpt message.

Delegated SR-TE LSP

To report the optimization metrics for delegated SR-TE (SR MPLS) LSPs, configure the optimization metrics (IGP, TE, and path delay). You can also report bandwidth and reservation priority for delegated SR-TE LSPs.

Intended Metric:

- When the optimization metric is configured at the time of delegation of LSP, the information is sent to the PCE through the PCRpt message.
- When the optimization metric is configured after the delegation of LSP, the change is applied on the LSP/communicated to the PCE when the LSP control status becomes locally controlled.
- When PCUpd message is received, if optimization metric is present in the message, the metric is used as intended metric in subsequent PCRpt messages till the LSP control status is externally controlled.
- When PCUpd message is received, if optimization metric is not present in the message, subsequent PCRpt messages do not contain intended metric.
- When the LSP control status changes to locally controlled, optimization metric configured from Junos CLI will be the intended metric in the PCRpt message.
- Bandwidth and reservation priority configured in the compute profile are also reported in PCRpt messages when the LSP is delegated.
- If bandwidth or priority values are updated in a PCUpd message, the PCC reports the received values as intended metrics in subsequent PCRpt messages while the LSP is externally controlled.
- If bandwidth or priority values are not present in a PCUpd message, subsequent PCRpt messages carry the values as `0`.
- When the LSP control status changes to locally controlled, bandwidth and priority are reset to `0`.

Actual Metric:

- When LSP is delegated after the creation, at the time of LSP delegation if LSP has 1 ERO, computed values of IGP, TE and delay metrics are sent as actual metrics in the PCRpt message.
- When LSP is delegated after the creation, at the time of LSP delegation if LSP has multiple EROs, computed metric/actual metric is not sent in the PCRpt message as actual metric needs to be sent per LSP (not per ERO) in PCEP.
- When PCUpd message is received, if computed metric is present in the message, the metric is used as actual metric in subsequent PCRpt messages till the LSP control status is externally controlled.
- When PCUpd message is received, if computed metric is not present in the message, subsequent PCRpt messages do not contain actual metric.

- When the LSP control status changes to locally controlled, IGP, TE and delay metrics computed in PCC are sent as actual metrics in the PCRpt message.
- If bandwidth is received in a PCUpd message, the received value is reported as actual bandwidth in subsequent PCRpt messages while the LSP is externally controlled. If not present, the last reported value is retained until control changes. At delegation, actual bandwidth is set to `0`. When the LSP becomes locally controlled, actual bandwidth is reset to `0`.
- If setup and hold priority values are received in a PCUpd message, the received values are reported as actual metrics in subsequent PCRpt messages while the LSP remains externally controlled. If not present, the values are reported as `0`. When the LSP becomes locally controlled, both priorities are reset to `0`.

PCE-initiated SR-TE LSP

Intended metrics or actual metric sent by PCE in PCInit/PCUpd messages are reported back to PCE through PCRpt message till the LSP is externally controlled. Junos OS also reports bandwidth and reservation priority for PCE-initiated SR-TE LSPs.

Intended Metric:

- When PCInit/PCUpd message is received, if optimization metric is present in the message, the metric is used as intended metric in subsequent PCRpt messages till the LSP control status is externally controlled.
- When PCInit/PCUpd message is received, if optimization metric is not present in the message, subsequent PCRpt messages do not contain intended metric.
- When the LSP control status becomes locally controlled, intended metric will not be sent.
- If bandwidth or reservation priority values are included in a PCInit/PCUpd message, the PCC reports those values as intended metrics in subsequent PCRpt messages while the LSP remains externally controlled.
- If bandwidth or reservation priority values are not present in the PCInit/PCUpd message, the PCC reports both values as `0`.
- When the LSP becomes locally controlled, both values are reset to `0`.

Actual Metric:

- When PCInit/PCUpd message is received, if computed metric is present in the message, the metric is used as actual metric in subsequent PCRpt messages till the LSP control status is externally controlled.
- When PCInit/PCUpd message is received, if computed metric is not present in the message, subsequent PCRpt messages do not contain actual metric.

- When the LSP control status changes to locally controlled, subsequent PCRpt messages do not contain actual metric.
- If bandwidth values are present in a PCInit/PCUpd message, they are reported as actual bandwidth in PCRpt messages while the LSP remains externally controlled. If not present, existing bandwidth remains unchanged until the LSP becomes locally controlled, at which point it is reset to `0`.
- If setup and hold priority values are included in a PCInit/PCUpd message, the PCC reports them as actual metrics in subsequent PCRpt messages while the LSP remains externally controlled. If absent, values are reported as `0`. When the LSP becomes locally controlled, both are reset to `0`.

Sending Optimization Metric in PCRpt Message

Optimization metric is sent to the PCE through the `intended-attributes-list` in the PCRpt message. Metric value is set to 0 and B, C flags are set to 0. Metric type indicates the metric to be optimized.

Sending Computed Metric in PCRpt Message

Computed metric is sent to the PCE through the `actual-attributes-list` in the PCRpt message. Metric value is the computed metric value and metric type indicates the computed metric type. B flag is set to 0, C flag is set to 1.

Backward Incompatibility for Route Metric

As route metric is supported using vendor TLV, PCC will not process route metric sent in metric object by Juniper PCE supporting Northstar and older releases of paragon pathfinder.

Configuring Optimization Metrics for LSPs

You can configure optimization metrics (IGP, TE, and path delay) for RSVP LSPs and SR-TE LSP.

To configure the IGP, TE, and path delay optimization metrics for RSVP LSPs, include the `metric-type <igp/te/delay/delay minimum>` CLI statement at the [`edit protocols mpls label-switched-path <lsp-name>`] hierarchy level.

To configure the IGP, TE, and path delay optimization metrics for SR-TE LSPs, include the `metric-type <igp/te/delay/delay minimum>` CLI statement at the [`edit protocols source-packet-routing compute-profile <compute-profile-name>`] hierarchy level.

Sample Output

You can use the `show path-computation-client lsp` and `show path-computation-client lsp extensive` CLI commands to display the state of label-switched paths (LSPs) known to the Path Computation Client (PCC).

The following is a sample output of `show path-computation-client lsp extensive`:

```
user@host> show path-computation-client lsp extensive name sr_lsp


LSP Name                  : sr_lsp
PathName                  : -
From                      : 192.168.1.101
To                        : 192.168.1.106
Path Setup Type           : spring-te
State                     : Up
Active Path               : Yes
Link Protection           : none
LSP Type                  : ext-provised
P2mp tree                 : NULL
Path cspf status          : external_cspf
Controller                : ns1
Template                  : NULL
PLSP-ID                   : 31
LSP-ID                    : 0
RSVP Error                : 0x0
Requested AutoBw          : 0bps
Record Route              : (Label=299792)
From PCE ERO (received)   : (Label=299792)
From RPD ERO (reported)   : (Label=299792)
Configured ERO on PCC     : Not Supported
Bandwidth:
        Intended          : 98.76Kbps
        Actual            : 0bps
Intended Metric:
        Metric type                     Bound             Optimization
        IGP                             0                 TRUE
Actual Metric:
        Metric type                     Computed value
        IGP                        50
Route Metric              : 50
Mapped Flowspec (FS-Ids)  : -
LSP Attributes:
       Exclude-Any: 0, Include-Any: 0, Include-All: 0
       Setup Priority: 0, Hold-Priority: 0
       Local Protection Bit: FALSE
Last Rpt/Pcreqest received from RPD at        : 22:15:32.000
```

```
Last Update sent to PCE at                    : 16:00:00.000
Last PcUpdate/PcCreate received from PCE at   : 22:15:32.000
Last error sent to PCE at                     : 16:00:00.000
Last 5 reasons to send Report/Pcrequest       : , , , ,
```

The output shows that the LSP is optimized with the metric type IGP. The computed value of IGP metric is 50. The Route metric installed in the route table is 50.

SRv6-TE Tunnels with micro-SIDs in PCEP

SUMMARY

IN THIS SECTION

- Benefits of SRv6-TE Tunnels with Micro-SIDs Support in PCEP | **2308**
- Overview | **2308**

Support for SRv6-TE tunnels with micro-SIDs in PCEP enhances traffic engineering and network optimization by enabling the reporting, delegation, and creation of these tunnels. You can report and delegate static SRv6-TE tunnels with micro-SID configurations to a PCE and initiate these tunnels through PCE, improving control and management. Key functionalities include reporting static SRv6-TE tunnels with micro-SIDs to the PCE, delegating their management, and creating them with proper SID structure and endpoint behavior checks. Existing CLI commands are extended to support these features, facilitating effective configuration and monitoring.

Benefits of SRv6-TE Tunnels with Micro-SIDs Support in PCEP

- Enhance traffic engineering by allowing the PCE to create and manage SRv6-TE tunnels with micro-SIDs, optimizing network performance and resource utilization.
- Provide better network control and visibility through the reporting and delegation of static SRv6-TE tunnels with micro-SID configurations to the PCE.

Overview

With the integration of SRv6-TE tunnels with micro-SIDs support in PCEP, you can significantly enhance your network's traffic engineering capabilities. This feature allows you to report, delegate, and create SRv6-TE tunnels with micro-SIDs, leveraging the Path Computation Element (PCE) for better network optimization and management. When you report static SRv6-TE tunnels with micro-SID configurations to the PCE, it includes comprehensive details such as the SID structure and endpoint behavior, enabling the PCE to manage these tunnels effectively.

Delegation of SRv6-TE tunnels with micro-SIDs to the PCE allows for enhanced control, as the PCE can manage the tunnel configurations and optimize routing paths dynamically. This delegation can be configured to occur post-creation, or you can set it up to combine creation and delegation in a single commit, streamlining your configuration process. Moreover, the PCE can initiate SRv6-TE tunnels with micro-SIDs, ensuring that proper SID structure and endpoint behavior checks are in place, thus maintaining the integrity and performance of your network routing.

You can use the following show commands to monitor SRv6-TE tunnels:

```
user@host> show spring-traffic-engineering lsp detail
user@host> show ted spring-te-policy extensive
user@host> show route table lsdist.0 protocol spring-te extensive
user@host> show path-computation-client lsp extensive name <lsp-name>
```

These commands provide detailed insights into the status and configuration of your SRv6-TE tunnels, allowing you to troubleshoot and optimize as needed. You can fully leverage the enhanced capabilities of SRv6-TE tunnels with micro-SIDs support in PCEP.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
21.R1	Starting in Junos OS Release 21.1R1, Junos OS supports nonstop active routing (NSR) for PCE-initiated RSVP-based point-to-point and point-to-multipoint LSPs.
21.R1	Starting in Junos OS Release 21.1R1, Junos OS supports nonstop active routing (NSR) for PCE-initiated RSVP-based point-to-multipoint LSPs.
19.4R1	You can associate a single or range of MVPN multicast flows (S,G) to a dynamically created PCE-initiated point-to-multipoint label-switched path (LSP).
19.4R1	You can configure a tunnel template for PCE-initiated segment routing LSPs to pass down two additional parameters for these LSPs - Bidirectional forwarding detection (BFD) and LDP tunneling.
17.2R1	Starting in Junos OS Release 17.2, in addition to `external cspf`, two new path computation types are introduced for the PCE-controlled LSPs: `local cspf` and `no cspf`.
16.1	Starting with Junos OS Release 16.1, you can secure a PCEP session using TCP-MD5 authentication as per RFC 5440.

16.1	Junos OS Release 16.1 introduces the feature of securing a PCEP session using TCP-MD5 authentication as per RFC 5440.
14.2R4	Starting in Junos OS Release 14.2R4, support of auto-bandwidth is provided for PCE-controlled LSPs.

RELATED DOCUMENTATION

- NorthStar Controller User Guide

PART 10

MPLS Troubleshooting

- Troubleshooting MPLS | **2312**

CHAPTER 20

Troubleshooting MPLS

IN THIS CHAPTER

- Troubleshooting MPLS | **2312**

Troubleshooting MPLS

IN THIS SECTION

- Verify MPLS Interfaces | **2313**
- Verify Protocol Families | **2316**
- Verify the MPLS Configuration | **2320**
- Checking the MPLS Layer | **2323**
- Verify That Node-Link Protection Is Up | **2344**
- Verify That Link Protection Is Up | **2352**
- Verify One-to-One Backup | **2358**
- Verify That the Primary Path Is Operational | **2366**
- Verify That the Secondary Path Is Established | **2368**
- Verifying the Physical Layer | **2371**
- Checking the Data Link Layer | **2382**
- Verifying the IP and IGP Layers | **2399**
- Verifying the IP Layer | **2402**
- Verify the LSP Again | **2417**
- Checking the RSVP Layer | **2421**
- Determining LSP Statistics | **2441**
- Verifying LSP Use in Your Network | **2443**
- Verify That Load Balancing Is Working | **2448**

- Verify the Operation of Uneven Bandwidth Load Balancing | **2453**
- Use the traceroute Command to Verify MPLS Labels | **2455**
- Troubleshooting GMPLS and GRE Tunnel | **2457**
- Determining LSP Status | **2482**
- Checking That RSVP Path Messages Are Sent and Received | **2489**

Verify MPLS Interfaces

IN THIS SECTION

- Purpose | **2313**
- Action | **2313**
- Meaning | **2315**

Purpose

If the MPLS protocol is not configured correctly on the routers in your network, the interfaces are not able to perform MPLS switching.

NOTE: For a labeled route to be resolved over an interface, `family mpls` must be configured at the `[edit interfaces]` hierarchy level for the route to be successfully resolved. When the interface is not configured with `family mpls`, labelled routes do not get resolved.

Action

To verify MPLS interfaces, enter the following Junos OS command-line interface (CLI) operational mode command:

```
user@host>                          show mpls interface
```

Sample Output 1

command-name

The following sample output is for all routers in the network shown in MPLS Network Topology.

```
user@R1> show mpls interface
Interface        State       Administrative groups
so-0/0/0.0       Up          <none>
so-0/0/1.0       Up          <none>
so-0/0/2.0       Up          <none>

user@R2> show mpls interface
Interface        State       Administrative groups
so-0/0/0.0       Up          <none>
so-0/0/1.0       Up          <none>
so-0/0/2.0       Up          <none>
so-0/0/3.0       Up          <none>

user@R3> show mpls interface
Interface        State       Administrative groups
so-0/0/0.0       Up          <none>
so-0/0/1.0       Up          <none>
so-0/0/2.0       Up          <none>
so-0/0/3.0       Up          <none>

user@R4> show mpls interface
Interface        State       Administrative groups
so-0/0/0.0       Up          <none>
so-0/0/1.0       Up          <none>
so-0/0/2.0       Up          <none>
so-0/0/3.0       Up          <none>

user@R5> show mpls interface
Interface        State       Administrative groups
so-0/0/0.0       Up          <none>
so-0/0/1.0       Up          <none>
so-0/0/2.0       Up          <none>

user@R6> show mpls interface
Interface        State       Administrative groups
so-0/0/0.0       Up          <none>
```

```
so-0/0/1.0       Up         <none>
so-0/0/2.0       Up         <none>
so-0/0/3.0       Up         <none>
```

Sample Output 2

command-name

```
user@R6>  show mpls interface
Interface        State       Administrative groups
so-0/0/0.0       Up         <none>
so-0/0/1.0       Up         <none>
so-0/0/3.0       Up         <none>       # so-0/0/2.0 is missing
```

Sample Output 3

command-name

```
user@host> show mpls interface
MPLS not configured
```

Meaning

Sample Output 1 shows that all MPLS interfaces on all routers in the network are enabled (**Up**) and can perform MPLS switching. If you fail to configure the correct interface at the [**edit protocols mpls**] hierarchy level or include the `family mpls` statement at the [**edit interfaces** ***type-fpc/pic/port*** **unit** ***number***] hierarchy level, the interface cannot perform MPLS switching, and does not appear in the output for the `show mpls interface` command.

Administrative groups are not configured on any of the interfaces shown in the example network in MPLS Network Topology. However, if they were, the output would indicate which affinity class bits are enabled on the router.

Sample Output 2 shows that interface **so-0/0/2.0** is missing and therefore might be incorrectly configured. For example, the interface might not be included at the [**edit protocols mpls**] hierarchy level, or the `family mpls` statement might not be included at the [**edit interfaces** ***type-fpc/pic/port*** **unit** ***number***] hierarchy level. If the interface is configured correctly, RSVP might not have signaled over this interface

yet. For more information on determining which interface is incorrectly configured, see "Verify Protocol Families" on page 2316.

Sample Output 3 shows that the MPLS protocol is not configured at the [**edit protocols mpls**] hierarchy level.

Verify Protocol Families

IN THIS SECTION

- Purpose | **2316**
- Action | **2316**
- Meaning | **2320**

Purpose

If a logical interface does not have MPLS enabled, it cannot perform MPLS switching. This step allows you to quickly determine which interfaces are configured with MPLS and other protocol families.

Action

To verify the protocol families configured on the routers in your network, enter the following Junos OS CLI operational mode command:

```
user@host>                          show interfaces terse
```

Sample Output 1

command-name

```
user@R1> show interfaces terse
Interface               Admin Link Proto Local                 Remote
so-0/0/0                up    up
so-0/0/0.0              up    up   inet  10.1.12.1/30
                                    iso
                                    mpls
so-0/0/1                up    up
```

```
so-0/0/1.0              up    up   inet  10.1.15.1/30
                                   iso
                                   mpls
so-0/0/2                up    up
so-0/0/2.0              up    up   inet  10.1.13.1/30
                                   iso
                                   mpls
so-0/0/3                up    down

user@R2> show interfaces terse
Interface               Admin Link Proto Local                 Remote
so-0/0/0                up    up
so-0/0/0.0              up    up   inet  10.1.12.2/30
                                   iso
                                   mpls
so-0/0/1                up    up
so-0/0/1.0              up    up   inet  10.1.23.1/30
                                   iso
                                   mpls
so-0/0/2                up    up
so-0/0/2.0              up    up   inet  10.1.26.1/30
                                   iso
                                   mpls
so-0/0/3                up    up
so-0/0/3.0              up    up   inet  10.1.24.1/30
                                   iso
                                   mpls

user@R3> show interfaces terse
Interface               Admin Link Proto Local                 Remote
so-0/0/0                up    up
so-0/0/0.0              up    up   inet  10.1.34.1/30
                                   iso
                                   mpls
so-0/0/1                up    up
so-0/0/1.0              up    up   inet  10.1.23.2/30
                                   iso
                                   mpls
so-0/0/2                up    up
so-0/0/2.0              up    up   inet  10.1.13.2/30
                                   iso
                                   mpls
so-0/0/3                up    up
```

```
so-0/0/3.0              up    up   inet  10.1.36.1/30
                                   iso
                                   mpls

user@R4> show interfaces terse
Interface               Admin Link Proto Local                 Remote
so-0/0/0                up    up
so-0/0/0.0              up    up   inet  10.1.34.2/30
                                   iso
                                   mpls
so-0/0/1                up    up
so-0/0/1.0              up    up   inet  10.1.46.1/30
                                   iso
                                   mpls
so-0/0/2                up    up
so-0/0/2.0              up    up   inet  10.1.45.1/30
                                   iso
                                   mpls
so-0/0/3                up    up
so-0/0/3.0              up    up   inet  10.1.24.2/30
                                   iso
                                   mpls

user@R5> show interfaces terse
Interface               Admin Link Proto Local                 Remote
so-0/0/0                up    up
so-0/0/0.0              up    up   inet  10.1.56.1/30
                                   iso
                                   mpls
so-0/0/1                up    up
so-0/0/1.0              up    up   inet  10.1.15.2/30
                                   iso
                                   mpls
so-0/0/2                up    up
so-0/0/2.0              up    up   inet  10.1.45.2/30
                                   iso
                                   mpls
so-0/0/3                up    down

user@R6>  show interfaces terse
Interface               Admin Link Proto Local                 Remote
so-0/0/0                up    up
so-0/0/0.0              up    up   inet  10.1.56.2/30
```

```
                                    iso
                                    mpls
so-0/0/1                up    up
so-0/0/1.0              up    up    inet  10.1.46.2/30
                                    iso
                                    mpls
so-0/0/2                up    up
so-0/0/2.0              up    up    inet  10.1.26.2/30
                                    iso
                                    mpls
so-0/0/3                up    up
so-0/0/3.0              up    up    inet  10.1.36.2/30
                                    iso
                                    mpls
```

Sample Output 2

command-name

```
user@R6> show interfaces terse
Interface               Admin Link Proto Local                 Remote
so-0/0/0                up    up
so-0/0/0.0              up    up    inet  10.1.56.2/30
                                    iso
                                    mpls
so-0/0/1                up    up
so-0/0/1.0              up    up    inet  10.1.46.2/30
                                    iso
                                    mpls
so-0/0/2                up    up
so-0/0/2.0              up    up    inet  10.1.26.2/30
                                    iso  #The mpls statement is missing.
so-0/0/3                up    up
so-0/0/3.0              up    up    inet  10.1.36.2/30
                                    iso
                                    mpls
```

Meaning

Sample Output 1 shows the interface, the administrative status of the link (**Admin**), the data link layer status of the link (**Link**), the protocol families configured on the interface (**Proto**), and the local and remote addresses on the interface.

All interfaces on all routes in the network shown in MPLS Network Topology are administratively enabled and functioning at the data link layer with MPLS and IS-IS, and have an **inet** address. All are configured with an IPv4 protocol family (**inet**), and have the IS-IS (**iso**) and MPLS (**mpls**) protocol families configured at the [**edit interfaces** *type-fpc/pic/port* **unit** *number*] hierarchy level.

Sample Output 2 shows that interface **so-0/0/2.0** on **R6** does not have the `mpls` statement included at the [**edit interfaces** *type-fpc/pic/port* **unit** *number*] hierarchy level.

Verify the MPLS Configuration

IN THIS SECTION

- Purpose | **2320**
- Action | **2321**
- Meaning | **2323**

Purpose

After you have checked the transit and ingress routers, use the `traceroute` command to verify the BGP next hop, and used the `ping` command to verify the active path, you can check for problems with the MPLS configuration at the `[edit protocols mpls]` and `[edit interfaces]` hierarchy levels.

NOTE: For a labeled route to be resolved over an interface, `family mpls` must be configured at the `[edit interfaces]` hierarchy level for the route to be successfully resolved. When the interface is not configured with `family mpls`, labelled routes do not get resolved.

Action

To verify the MPLS configuration, enter the following commands from the ingress, transit, and egress routers:

```
user@host>              show configuration protocols mpls
user@host>              show configuration interfaces
```

Sample Output 1

command-name

```
user@R1> show configuration protocols mpls
label-switched-path R1-to-R6 {
    to 10.0.0.6;
}
inactive: interface so-0/0/0.0;
inactive: interface so-0/0/1.0;
interface so-0/0/2.0;
interface fxp0.0 {
    disable;
}

user@R3> show configuration protocols mpls
interface fxp0.0 {
    disable;
}
inactive: interface so-0/0/0.0;
inactive: interface so-0/0/1.0;
interface so-0/0/2.0;
interface so-0/0/3.0;

user@R6> show configuration protocols mpls
label-switched-path R6-to-R1 {
    to 10.0.0.1;
}
inactive: interface so-0/0/0.0;
inactive: interface so-0/0/1.0;
inactive: interface so-0/0/2.0;
inactive: interface so-0/0/3.0; <<< Incorrectly configured
```

Sample Output 2

command-name

```
user@R6> show configuration interfaces
so-0/0/0 {
    unit 0 {
        family inet {
            address 10.1.56.2/30;
        }
        family iso;
        family mpls;
    }
}
so-0/0/1 {
    unit 0 {
        family inet {
            address 10.1.46.2/30;
        }
        family iso;
        family mpls;
    }
}
so-0/0/2 {
    unit 0 {
        family inet {
            address 10.1.26.2/30;
        }
        family iso;
        family mpls;
    }
}
so-0/0/3 {
    unit 0 {
        family inet {
            address 10.1.36.2/30;
        }
        family iso;
        family mpls;
    }
}
fxp0 {
```

```
    unit 0 {
        family inet {
            address 192.168.70.148/21;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.0.0.6/32;
            address 127.0.0.1/32;
        }
        family iso {
            address 49.0003.1000.0000.0006.00;
        }
    }
}
```

Meaning

Sample Output 1 from the ingress, transit, and egress routers shows that the configuration of interfaces on egress router **R6** is incorrect. Interface **so-0/0/3.0** is included as inactive at the [**edit protocols mpls**] hierarchy level, when it should be active because it is the interface through which the LSP travels.

Sample Output 2 shows that interfaces are correctly configured for MPLS on egress router **R6**. The interfaces are also correctly configured on the ingress and transit routers (not shown).

Checking the MPLS Layer

IN THIS SECTION

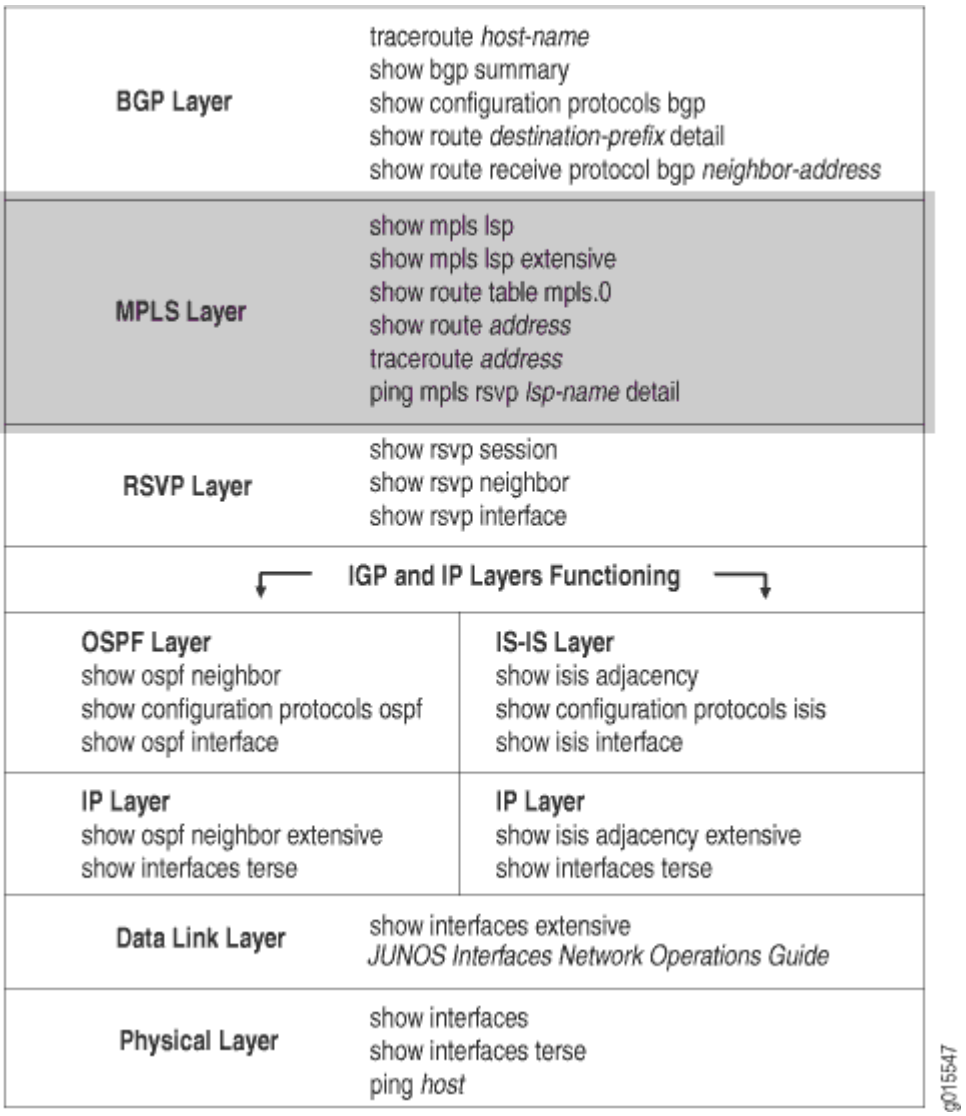
- Verify the LSP | **2326**
- Verify the LSP Route on the Transit Router | **2330**
- Verify the LSP Route on the Ingress Router | **2332**
- Verify MPLS Labels with the traceroute Command | **2334**
- Verify MPLS Labels with the ping Command | **2336**
- Take Appropriate Action | **2338**
- Verify the LSP Again | **2340**

Purpose

After you have configured the label-switched path (LSP), issued the `show mpls lsp` command, and determined that there is an error, you might find that the error is not in the physical, data link, Internet Protocol (IP), interior gateway protocol (IGP), or Resource Reservation Protocol (RSVP) layers. Continue investigating the problem at the MPLS layer of the network.

Figure 158 on page 2324 illustrates the MPLS layer of the layered MPLS model.

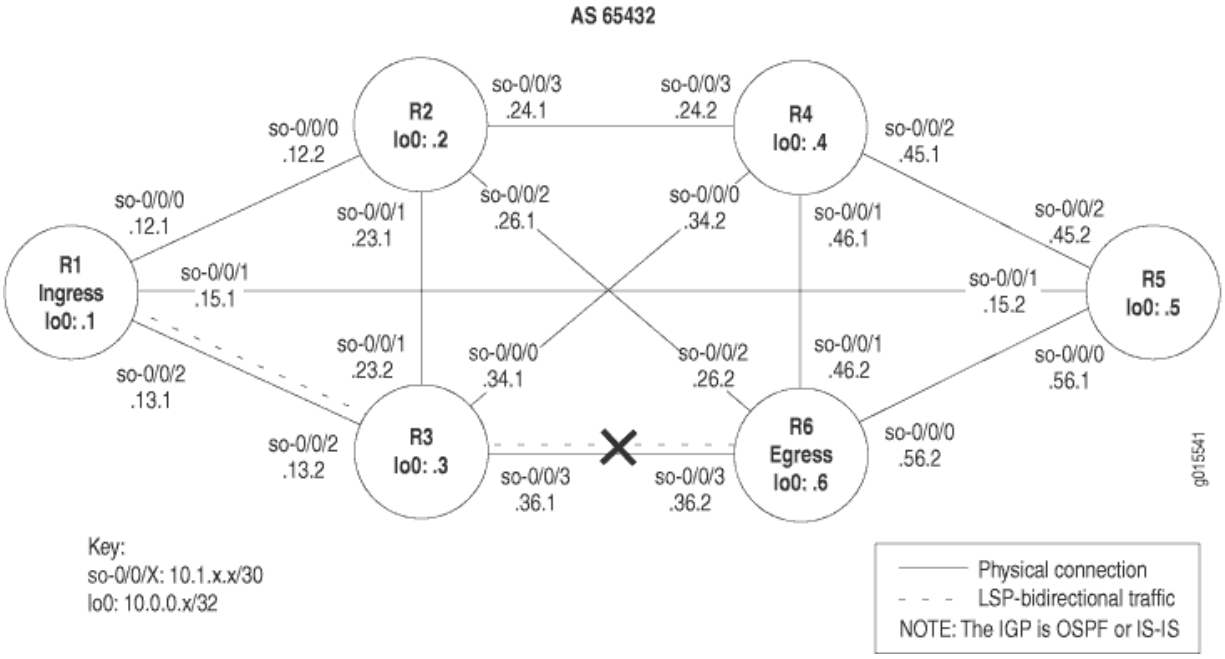
Figure 158: Checking the MPLS Layer



With the MPLS layer, you check whether the LSP is up and functioning correctly. If the network is not functioning at this layer, the LSP does not work as configured.

Figure 159 on page 2325 illustrates the MPLS network used in this topic.

Figure 159: MPLS Network Broken at the MPLS Layer



The network shown in Figure 159 on page 2325 is a fully meshed configuration where every directly connected interface can receive and send packets to every other similar interface. The LSP in this network is configured to run from ingress router **R1**, through transit router **R3**, to egress router **R6**. In addition, a reverse LSP is configured to run from **R6** through **R3** to **R1**, creating bidirectional traffic.

However, in this example, the reverse LSP is down without a path from **R6** to **R1**.

The cross shown in Figure 159 on page 2325 indicates where the LSP is broken. Some possible reasons the LSP is broken might include an incorrectly configured MPLS protocol, or interfaces that are incorrectly configured for MPLS.

In the network shown in Figure 159 on page 2325, a configuration error on egress router **R6** prevents the LSP from traversing the network as expected.

To check the MPLS layer, follow these steps:

Verify the LSP

IN THIS SECTION

- Purpose | 2326
- Action | 2326
- Meaning | 2330

Purpose

Typically, you use the `show mpls lsp extensive` command to verify the LSP. However for quick verification of the LSP state, use the `show mpls lsp` command. If the LSP is down, use the **extensive** option (`show mpls lsp extensive`) as a follow-up. If your network has numerous LSPs, you might consider specifying the name of the LSP, using the **name** option (`show mpls lsp name` ***name*** or `show mpls lsp name` ***name*** **extensive).**

Action

To verify that the LSP is up, enter some or all of the following commands from the ingress router:

```
user@host> show mpls lsp
user@host> show mpls lsp extensive
user@host> show mpls lsp name name
user@host> show mpls lsp name name extensive
```

Sample Output 1

command-name

```
user@R1> show mpls lsp
Ingress LSP: 1 sessions
To              From            State Rt ActivePath       P     LSPname
10.0.0.6        10.0.0.1        Dn     0 -                       R1-to-R6
Total 1 displayed, Up 0,  Down 1

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

```
Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

user@R3> show mpls lsp
Ingress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

user@R6> show mpls lsp
Ingress LSP: 1 sessions
To              From            State Rt ActivePath       P     LSPname
10.0.0.1        10.0.0.6        Dn     0 -                       R6-to-R1
Total 1 displayed, Up 0,  Down 1

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Sample Output 2

command-name

```
user@R1> show mpls lsp extensive
Ingress LSP: 1 sessions

10.0.0.6
  From: 10.0.0.1, State: Dn, ActiveRoute: 0, LSPname: R1-to-R6
  ActivePath: (none)
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
  Primary                   State: Dn
    Will be enqueued for recomputation in 22 second(s).
    1 Nov  2 14:43:38  CSPF failed: no route toward 10.0.0.6 [175 times]
```

```
  Created: Tue Nov  2 13:18:39 2004
Total 1 displayed, Up 0, Down 1

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

user@R3> show mpls lsp extensive
Ingress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

user@R6> show mpls lsp extensive
Ingress LSP: 1 sessions

10.0.0.1
  From: 10.0.0.6, State: Dn, ActiveRoute: 0, LSPname: R6-to-R1
  ActivePath: (none)
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
  Primary                   State: Dn
    Will be enqueued for recomputation in 13 second(s).
    1 Nov  2 14:38:12  CSPF failed: no route toward 10.0.0.1 [177 times]
  Created: Tue Nov  2 13:12:22 2004
Total 1 displayed, Up 0, Down 1

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Sample Output 3

command-name

```
user@R1> show mpls lsp name R1-to-R6
Ingress LSP: 1 sessions
To              From            State Rt ActivePath       P     LSPname
10.0.0.6        10.0.0.1        Dn     0 -                      R1-to-R6
Total 1 displayed, Up 0, Down 1

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Sample Output 4

command-name

```
user@R1> show mpls lsp name R1-to-R6 extensive
Ingress LSP: 1 sessions

10.0.0.6
  From: 10.0.0.1, State: Dn, ActiveRoute: 0, LSPname: R1-to-R6
  ActivePath: (none)
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
  Primary                    State: Dn
    Will be enqueued for recomputation in 10 second(s).
    1 Nov  2 14:51:53 CSPF failed: no route toward 10.0.0.6[192 times]
  Created: Tue Nov  2 13:18:39 2004
Total 1 displayed, Up 0, Down 1

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Meaning

Sample Output 1 shows a brief description of the state of the LSP for the ingress, transit, and egress routers. Output from ingress router **R1** and egress router **R6** shows that both LSPs are down, **R1-to-R6** and **R6-toR1**. With the configured LSPs on **R1** and **R6**, we would expect egress LSP sessions on both **R1** and **R6**. In addition, transit router **R3** has no transit sessions.

Sample Output 2 shows all information about the LSPs, including all past state history and the reason why an LSP failed. Output from **R1** and **R6** indicates that there is no route to the destination because the Constrained Shortest Path First (CSPF) algorithm failed.

Sample Outputs 3 and 4 show examples of the output for the `show mpls lsp name` command with the **extensive** option. In this instance, the output is very similar to the `show mpls lsp` command because only one LSP is configured in the example network in "MPLS Network Broken at the MPLS Layer" on page 2312. However, in a large network with many LSPs configured, the results would be quite different between the two commands.

Verify the LSP Route on the Transit Router

IN THIS SECTION

- Purpose | **2330**
- Action | **2330**
- Meaning | **2332**

Purpose

If the LSP is up, the LSP route should appear in the **mpls.0** routing table. MPLS maintains an MPLS path routing table (**mpls.0**), which contains a list of the next label-switched router in each LSP. This routing table is used on transit routers to route packets to the next router along an LSP. If routes are not present in the output for the transit router, check the MPLS protocol configuration on the ingress and egress routers.

Action

To verify the LSP route on the transit router, enter the following command from the transit router:

```
user@host> show route table mpls.0
```

Sample Output 1

command-name

```
user@R3> show route table mpls.0
mpls.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
0                  *[MPLS/0] 16w2d 21:52:40, metric 1
                      Receive
1                  *[MPLS/0] 16w2d 21:52:40, metric 1
                      Receive
2                  *[MPLS/0] 16w2d 21:52:40, metric 1
                      Receive
```

Sample Output 2

command-name

```
user@R3> show route table mpls.0
mpls.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
0                  *[MPLS/0] 16w2d 22:26:08, metric 1
                      Receive
1                  *[MPLS/0] 16w2d 22:26:08, metric 1
                      Receive
2                  *[MPLS/0] 16w2d 22:26:08, metric 1
                      Receive
100864             *[RSVP/7] 00:07:23, metric 1
                    > via so-0/0/2.0, label-switched-path R6-to-R1
100864(S=0)        *[RSVP/7] 00:07:23, metric 1
                    > via so-0/0/2.0, label-switched-path R6-to-R1
100880             *[RSVP/7] 00:07:01, metric 1
                    > via so-0/0/3.0, label-switched-path R1-to-R6
100880(S=0)        *[RSVP/7] 00:07:01, metric 1
                    > via so-0/0/3.0, label-switched-path R1-to-R6
```

Meaning

Sample Output 1 from transit router **R3** shows three route entries in the form of MPLS label entries. These MPLS labels are reserved MPLS labels defined in RFC 3032, and are always present in the **mpls.0** routing table, regardless of the state of the LSP. The incoming labels assigned by RSVP to the upstream neighbor are missing from the output, indicating that the LSP is down. For more information on MPLS label entries, see Checklist for Verifying LSP Use.

In contrast, Sample Output 2 shows the MPLS labels and routes for a correctly configured LSP. The three reserved MPLS labels are present, and the four other entries represent the incoming labels assigned by RSVP to the upstream neighbor. These four entries represent two routes. There are two entries per route because the stack values in the MPLS header may be different. For each route, the second entry **100864 (S=0)** and **100880 (S=0)** indicates that the stack depth is not 1, and additional label values are included in the packet. In contrast, the first entry, **100864** and **100880** has an inferred S=1 value which indicates a stack depth of 1 and makes each label the last label in that particular packet. The dual entries indicate that this is the penultimate router. For more information on MPLS label stacking, see RFC 3032, *MPLS Label Stack Encoding.*

Verify the LSP Route on the Ingress Router

IN THIS SECTION

- Purpose | **2332**
- Action | **2332**
- Meaning | **2334**

Purpose

Check whether the LSP route is included in the active entries in the **inet.3** routing table for the specified address.

Action

To verify the LSP route, enter the following command from the ingress router:

```
user@host>              show route destination
```

Sample Output 1

command-name

```
user@R1> show route 10.0.0.6
inet.0 : 27 destinations, 27 routes (27 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.0.0.6/32        *[IS-IS/18] 6d 01:41:37, metric 20
                      to 10.1.12.2 via so-0/0/0.0
                    > to 10.1.15.2 via so-0/0/1.0
                      to 10.1.13.2 via so-0/0/2.0

user@R6> show route 10.0.0.1

inet.0 : 28 destinations, 28 routes (28 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.0.0.1/32        *[IS-IS/18] 5d 01:01:38, metric 20
                      to 10.1.56.1 via so-0/0/0.0
                    > to 10.1.26.1 via so-0/0/2.0
                      to 10.1.36.1 via so-0/0/3.0
```

Sample Output 2

command-name

```
user@R1> show route 10.0.0.6
inet.0: 28 destinations, 28 routes (27 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.6/32        *[IS-IS/18] 6d 02:13:42, metric 20
                      to 10.1.12.2 via so-0/0/0.0
                    > to 10.1.15.2 via so-0/0/1.0
                      to 10.1.13.2 via so-0/0/2.0

inet.3 : 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.6/32        *[RSVP/7] 00:08:07, metric 20
                    > via so-0/0/2.0,  label-switched-path R1-to-R6
```

```
user@R6> show route 10.0.0.1

inet.0: 29 destinations, 29 routes (28 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.1/32        *[IS-IS/18] 5d 01:34:03, metric 20
                      to 10.1.56.1 via so-0/0/0.0
                    > to 10.1.26.1 via so-0/0/2.0
                      to 10.1.36.1 via so-0/0/3.0

inet.3 : 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.1/32        *[RSVP/7] 00:10:39, metric 20
                    >  via so-0/0/3.0, label-switched-path R6-to-R1
```

Meaning

Sample Output 1 shows entries in the **inet.0** routing table only. The **inet.3** routing table is missing from the output because the LSP is not working. The **inet.0** routing table is used by interior gateway protocols (IGPs) and Border Gateway Protocol (BGP) to store routing information. In this case, the IGP is Intermediate System-to-Intermediate System (IS-IS). For more information on the **inet.0** routing table, see the *Junos MPLS Applications Configuration Guide*.

If the LSP was working, we would expect to see entries that include the LSP in the **inet.3** routing table. The **inet.3** routing table is used on ingress routers to route BGP packets to the destination egress router. BGP uses the **inet.3** routing table on the ingress router to help resolve next-hop addresses. BGP is configured in the example network shown in "MPLS Network Broken at the MPLS Layer" on page 2312.

Sample Output 2 shows output you should receive when the LSP is up. The output shows both the **inet.0** and **inet.3** routing tables, indicating that LSPs **R1-to-R6** and **R6-to-R1** are available.

Verify MPLS Labels with the traceroute Command

IN THIS SECTION

- Purpose | 2335
- Action | 2335
- Meaning | 2336

Purpose

Display the route packets take to a BGP destination where the BGP next hop for that route is the LSP egress address. By default, BGP uses the **inet.0** and the **inet.3** routing tables to resolve the next-hop address. When the next-hop address of the BGP route is not the router ID of the egress router, traffic is mapped to IGP routes, not to the LSP. Use the `traceroute` command as a debugging tool to determine whether the LSP is being used to forward traffic.

Action

To verify MPLS labels, enter the following command from the ingress router:

```
user@host> traceroute hostname
```

Sample Output 1

command-name

```
user@R1> traceroute 100.100.6.1
traceroute to 100.100.6.1 (100.100.6.1), 30 hops max, 40 byte packets
 1  10.1.12.2 (10.1.12.2)  0.627 ms  0.561 ms  0.520 ms
 2  10.1.26.2 (10.1.26.2)  0.570 ms !N  0.558 ms !N  4.879 ms !N

user@R6> traceroute 100.100.1.1
traceroute to 100.100.1.1 (100.100.1.1), 30 hops max, 40 byte packets
 1  10.1.26.1 (10.1.26.1)  0.630 ms  0.545 ms  0.488 ms
 2  10.1.12.1 (10.1.12.1)  0.551 ms !N  0.557 ms !N  0.526 ms !N
```

Sample Output 2

command-name

```
user@R1> traceroute 100.100.6.1
 to 100.100.6.1 (100.100.6.1), 30 hops max, 40 byte packets
 1  10.1.13.2 (10.1.13.2)  0.866 ms  0.746 ms  0.724 ms
      MPLS Label=100912 CoS=0 TTL=1 S=1
 2  10.1.36.2 (10.1.36.2)  0.577 ms !N  0.597 ms !N  0.546 ms !N
```

```
user@R6> traceroute 100.100.1.1
traceroute to 100.100.1.1 (100.100.1.1), 30 hops max, 40 byte packets
 1  10.1.36.1 (10.1.36.1)  0.802 ms  0.716 ms  0.688 ms
      MPLS Label=100896 CoS=0 TTL=1 S=1
 2  10.1.13.1 (10.1.13.1)  0.570 ms !N  0.568 ms !N  0.546 ms !N
```

Meaning

Sample Output 1 shows that BGP traffic is not using the LSP, consequently MPLS labels do not appear in the output. Instead of using the LSP, BGP traffic is using the IGP (IS-IS, in the example network in "MPLS Network Broken at the MPLS Layer" on page 2312) to reach the BGP next-hop LSP egress address. The Junos OS default behavior uses LSPs for BGP traffic when the BGP next hop equals the LSP egress address.

Sample Output 2 is an example of output for a correctly configured LSP. The output shows MPLS labels, indicating that BGP traffic is using the LSP to reach the BGP next hop.

Verify MPLS Labels with the ping Command

IN THIS SECTION

- Purpose | **2336**
- Action | **2336**
- Meaning | **2338**

Purpose

When you ping a specific LSP, you check that echo requests are sent over the LSP as MPLS packets.

Action

To verify MPLS labels, enter the following command from the ingress router to ping the egress router:

```
user@host> ping mpls rsvp lsp-name detail
```

For example:

```
user@R1> ping mpls rsvp R1-to-R6 detail
```

Sample Output 1

command-name

```
user@R1> ping mpls rsvp R1-to-R6 detail
LSP R1-to-R6 - LSP has no active path, exiting.

user@R6> ping mpls rsvp R6-to-R1 detail
LSP R6-to-R1 - LSP has no active path, exiting.
```

Sample Output 2

command-name

```
user@R1> traceroute 10.0.0.6
traceroute to 10.0.0.6 (10.0.0.6), 30 hops max, 40 byte packets
 1  10.1.15.2 (10.1.15.2)  0.708 ms  0.613 ms  0.576 ms
 2  10.0.0.6 (10.0.0.6)  0.763 ms  0.708 ms  0.700 ms

user@R1> ping mpls rsvp R1-to-R6 detail
Request for seq 1, to interface 69, label 100880
Reply for seq 1, return code: Egress-ok
Request for seq 2, to interface 69, label 100880
Reply for seq 2, return code: Egress-ok
Request for seq 3, to interface 69, label 100880
Reply for seq 3, return code: Egress-ok
Request for seq 4, to interface 69, label 100880
Reply for seq 4, return code: Egress-ok
Request for seq 5, to interface 69, label 100880
Reply for seq 5, return code: Egress-ok

--- lsping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
```

```
user@R6> ping mpls rsvp R6-to-R1 detail
Request for seq 1, to interface 70, label 100864
Reply for seq 1, return code: Egress-ok
Request for seq 2, to interface 70, label 100864
Reply for seq 2, return code: Egress-ok
Request for seq 3, to interface 70, label 100864
Reply for seq 3, return code: Egress-ok
Request for seq 4, to interface 70, label 100864
Reply for seq 4, return code: Egress-ok
Request for seq 5, to interface 70, label 100864
Reply for seq 5, return code: Egress-ok

--- lsping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
```

Meaning

Sample Output 1 shows that the LSP does not have an active path to forward echo requests, indicating that the LSP is down.

Sample Output 2 is an example of output you should receive when the LSP is up and forwarding packets.

Take Appropriate Action

IN THIS SECTION

- Problem | 2338
- Solution | 2339

Problem

Description

Depending on the error you encountered in your investigation, you must take the appropriate action to correct the problem. In this example, an interface is incorrectly configured at the [**edit protocols mpls**] hierarchy level on egress router **R6.**

Solution

To correct the error in this example, follow these steps:

1. Activate the interface in the MPLS protocol configuration on egress router **R6**:

```
user@R6> edit
user@R6# edit protocols mpls
[edit protocols mpls]
user@R6# show
user@R6# activate interface so-0/0/3.0
```

2. Verify and commit the configuration:

```
[edit protocols mpls]
user@R6# show
user@R6# commit
```

Sample Output

```
user@R6> edit
Entering configuration mode

[edit]
user@R6# edit protocols mpls

[edit protocols mpls]
user@R6# show
label-switched-path R6-to-R1 {
    to 10.0.0.1;
}
inactive: interface so-0/0/0.0;
inactive: interface so-0/0/1.0;
inactive: interface so-0/0/2.0;
inactive: interface so-0/0/3.0; <<< Incorrectly configured interface

[edit protocols mpls]
user@R6#  activate interface so-0/0/3

[edit protocols mpls]
```

```
user@R6# show
label-switched-path R6-to-R1 {
    to 10.0.0.1;
}
inactive: interface so-0/0/0.0;
inactive: interface so-0/0/1.0;
inactive: interface so-0/0/2.0;
interface so-0/0/3.0; <<< Correctly configured interface

[edit protocols mpls]
user@R6# commit
commit complete
```

Meaning

The sample output shows that the incorrectly configured interface **so-0/0/3.0** on egress router **R6** is now activated at the [**edit protocols mpls**] hierarchy level. The LSP can now come up.

Verify the LSP Again

IN THIS SECTION

- Purpose | **2340**
- Action | **2340**
- Meaning | **2344**

Purpose

After taking the appropriate action to correct the error, the LSP needs to be checked again to confirm that the problem in the BGP layer has been resolved.

Action

To verify the LSP again, enter the following command from the ingress, transit, and egress routers:

```
user@host>              show mpls lsp extensive
```

Sample Output

command-name

```
user@R1> show mpls lsp extensive
Ingress LSP: 1 sessions

10.0.0.6
  From: 10.0.0.1,  State: Up ,  ActiveRoute: 1 ,  LSPname: R1-to-R6
  ActivePath:  (primary)
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                   State: Up
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 20)
 10.1.13.2 S 10.1.36.2 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt):
          10.1.13.2 10.1.36.2
    6 Nov  2 15:48:52 Selected as active path
    5 Nov  2 15:48:52 Record Route:  10.1.13.2 10.1.36.2
    4 Nov  2 15:48:52 Up
    3 Nov  2 15:48:52 Originate Call
    2 Nov  2 15:48:52 CSPF: computation result accepted
    1 Nov  2 15:48:22 CSPF failed: no route toward 10.0.0.6[308 times]
  Created: Tue Nov  2 13:18:39 2004
Total 1 displayed, Up 1, Down 0

Egress LSP: 1 sessions

10.0.0.1
  From: 10.0.0.6,  LSPstate: Up , ActiveRoute: 0
   LSPname: R6-to-R1 , LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: -
  Resv style: 1 FF, Label in: 3, Label out: -
  Time left:  159, Since: Tue Nov  2 15:48:30 2004
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 39106 protocol 0
  PATH rcvfrom: 10.1.13.2 (so-0/0/2.0) 10 pkts
  Adspec: received MTU 1500
  PATH sentto: localclient
  RESV rcvfrom: localclient
  Record route: 10.1.36.2 10.1.13.2 <self>
```

```
Total 1 displayed, Up 1, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

user@R3> show mpls lsp extensive
Ingress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 2 sessions

10.0.0.1
  From: 10.0.0.6, LSPstate: Up, ActiveRoute: 1
  LSPname: R6-to-R1, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 3
  Resv style: 1 FF, Label in: 100864, Label out: 3
  Time left:  123, Since: Tue Nov  2 15:35:41 2004
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 39106 protocol 0
  PATH rcvfrom: 10.1.36.2 (so-0/0/3.0) 10 pkts
  Adspec: received MTU 1500 sent MTU 1500
  PATH sentto: 10.1.13.1 (so-0/0/2.0) 10 pkts
  RESV rcvfrom: 10.1.13.1 (so-0/0/2.0) 10 pkts
  Explct route: 10.1.13.1
  Record route: 10.1.36.2 <self> 10.1.13.1

10.0.0.6
  From: 10.0.0.1, LSPstate: Up, ActiveRoute: 1
  LSPname: R1-to-R6, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 3
  Resv style: 1 FF, Label in: 100880, Label out: 3
  Time left:  145, Since: Tue Nov  2 15:36:03 2004
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 48015 protocol 0
  PATH rcvfrom: 10.1.13.1 (so-0/0/2.0) 10 pkts
  Adspec: received MTU 1500 sent MTU 1500
  PATH sentto: 10.1.36.2 (so-0/0/3.0) 10 pkts
  RESV rcvfrom: 10.1.36.2 (so-0/0/3.0) 10 pkts
```

```
  Explct route: 10.1.36.2
  Record route: 10.1.13.1 <self> 10.1.36.2
Total 2 displayed, Up 2, Down 0

user@R6> show mpls lsp extensive
Ingress LSP: 1 sessions

10.0.0.1
  From: 10.0.0.6,  State: Up, ActiveRoute: 1, LSPname: R6-to-R1
  ActivePath:  (primary)
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                    State: Up
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 20)
 10.1.36.1 S 10.1.13.1 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt):
          10.1.36.1 10.1.13.1
    6 Nov  2 15:41:44 Selected as active path
    5 Nov  2 15:41:44 Record Route:  10.1.36.1 10.1.13.1
    4 Nov  2 15:41:44 Up
    3 Nov  2 15:41:44 Originate Call
    2 Nov  2 15:41:44 CSPF: computation result accepted
    1 Nov  2 15:41:14 CSPF failed: no route toward 10.0.0.1[306 times]
  Created: Tue Nov  2 13:12:21 2004
Total 1 displayed, Up 1, Down 0

Egress LSP: 1 sessions

10.0.0.6
  From: 10.0.0.1,  LSPstate: Up,  ActiveRoute: 0
   LSPname: R1-to-R6 , LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: -
  Resv style: 1 FF, Label in: 3, Label out: -
  Time left:  157, Since: Tue Nov  2 15:42:06 2004
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 48015 protocol 0
  PATH rcvfrom: 10.1.36.1 (so-0/0/3.0) 11 pkts
  Adspec: received MTU 1500
  PATH sentto: localclient
  RESV rcvfrom: localclient
  Record route: 10.1.13.1 10.1.36.1 <self>
Total 1 displayed, Up 1, Down 0
```

```
Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Meaning

Sample Output 1 from ingress router **R1** shows that LSP **R1-to-R6** has an active route to **R6** and the state is up.

Sample Output 2 from transit router **R3** shows that there are two transit LSP sessions, one from **R1** to **R6** and the other from **R6** to **R1**. Both LSPs are up.

Sample Output 3 from egress router **R6** shows that the LSP is up and the active route is the primary route. The LSP is now traversing the network along the expected path, from **R1** through **R3** to **R6**, and the reverse LSP, from **R6** through **R3** to **R1**.

Verify That Node-Link Protection Is Up

IN THIS SECTION

- Purpose | **2344**
- Action | **2344**
- Meaning | **2345**

Purpose

After you configure node-link protection, you must check that bypass paths are up. You can also check the number of LSPs protected by the bypass paths. In the network shown in "Node-Link Protection" on page 361, two bypass paths should be up: one next-hop bypass path protecting the link between **R1** and **R2** (or next-hop **10.0.12.14**), and a next-next-hop bypass path avoiding **R2**.

Action

To verify node-link protection (many-to-one backup), enter the following Junos OS CLI operational mode commands on the ingress router. You can also issue the commands on transit routers and other routers used in the bypass path for slightly different information.

```
show mpls lsp
show mpls lsp extensive
```

```
show rsvp interface
show rsvp interface extensive
show rsvp session detail
```

Sample Output

command-name

```
user@R1> show mpls lsp
Ingress LSP: 1 sessions
To              From            State Rt ActivePath       P     LSPname
192.168.5.1     192.168.1.1     Up     0 via-r2           *      lsp2-r1-to-r5
Total 1 displayed,  Up 1 , Down 0

Egress LSP: 1 sessions
To              From            State   Rt Style Labelin Labelout LSPname
192.168.1.1     192.168.5.1     Up       0  1 FF       3        -  r5-to-r1
Total 1 displayed,  Up 1 , Down 0

Transit LSP: 2 sessions
To              From            State   Rt Style Labelin Labelout LSPname
192.168.0.1     192.168.6.1      Up        0  1 FF 100464   101952  lsp1-r6-to-r0
192.168.6.1     192.168.0.1      Up        0  1 FF  100448        3  r0-to-t6
Total 2 displayed, Up 2, Down 0
```

Meaning

Sample output from **R1** for the `show mpls lsp` command shows a brief description of the state of configured and active LSPs for which **R1** is the ingress, transit, and egress router. All LSPs are up. **R1** is the ingress router for **lsp2-r1-to-r5**, and the egress router for return LSP **r5-to-r1**. Two LSPs transit **R1**, **lsp1-r6-to-r0** and the return LSP **r0-to-t6**. For more detailed information about the LSP, include the **extensive** option when you issue the `show mpls lsp` command.

Sample Output

```
user@R1> show mpls lsp extensive
Ingress LSP: 1 sessions

192.168.5.1
```

```
  From: 192.168.1.1,  State: Up , ActiveRoute: 0,  LSPname: lsp2-r1-to-r5
  ActivePath: via-r2 (primary)
   Node/Link protection desired
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary   via-r2          State: Up
    SmartOptimizeTimer: 180
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 3)
 10.0.12.14 S 10.0.24.2 S 10.0.45.2 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt):
          10.0.12.14(Label=101872) 10.0.24.2(Label=101360) 10.0.45.2(Label=3)
    11 Jul 11 14:30:58 Link-protection Up
   10 Jul 11 14:28:28 Selected as active path
    [...Output truncated...]
  Created: Tue Jul 11 14:22:58 2006
Total 1 displayed, Up 1, Down 0

Egress LSP: 1 sessions

192.168.1.1
  From: 192.168.5.1,  LSPstate: Up, ActiveRoute: 0
   LSPname: r5-to-r1, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: -
  Resv style: 1 FF, Label in: 3, Label out: -
  Time left:  146, Since: Tue Jul 11 14:28:36 2006
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 29228 protocol 0
  PATH rcvfrom: 10.0.12.14 (fe-0/1/0.0) 362 pkts
  Adspec: received MTU 1500
  PATH sentto: localclient
  RESV rcvfrom: localclient
  Record route: 10.0.45.2 10.0.24.2 10.0.12.14 <self>
Total 1 displayed, Up 1, Down 0

Transit LSP: 2 sessions

192.168.0.1
  From: 192.168.6.1,  LSPstate: Up, ActiveRoute: 0
   LSPname: lsp1-r6-to-r0, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 101952
  Resv style: 1 SE, Label in: 100464, Label out: 101952
```

```
  Time left:  157, Since: Tue Jul 11 14:31:38 2006
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 11131 protocol 0
   Node/Link protection desired
   Type: Node/Link protected LSP, using Bypass->10.0.12.14->10.0.24.2
      1 Jul 11 14:31:38 Node protection up, using Bypass->10.0.12.14->10.0.24.2
  PATH rcvfrom: 10.0.16.2 (so-0/0/3.0) 509 pkts
  Adspec: received MTU 1500 sent MTU 1500
  PATH sentto: 10.0.12.14 (fe-0/1/0.0) 356 pkts
  RESV rcvfrom: 10.0.12.14 (fe-0/1/0.0) 358 pkts
  Explct route: 10.0.12.14 10.0.24.2 10.0.45.2 10.0.50.2
  Record route: 10.0.16.2 <self> 10.0.12.14 10.0.24.2 10.0.45.2 10.0.50.2

192.168.6.1
  From: 192.168.0.1,  LSPstate: Up, ActiveRoute: 0
  LSPname: r0-to-t6, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 3
  Resv style: 1 FF, Label in: 100448, Label out: 3
  Time left:  147, Since: Tue Jul 11 14:31:36 2006
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 23481 protocol 0
  PATH rcvfrom: 10.0.12.14 (fe-0/1/0.0) 358 pkts
  Adspec: received MTU 1500 sent MTU 1500
  PATH sentto: 10.0.16.2 (so-0/0/3.0) 350 pkts
  RESV rcvfrom: 10.0.16.2 (so-0/0/3.0) 323 pkts
  Explct route: 10.0.16.2
  Record route: 10.0.50.2 10.0.45.2 10.0.24.2 10.0.12.14 <self> 10.0.16.2
Total 2 displayed, Up 2, Down 0
```

Meaning

Sample output from **R1** for the `show mpls lsp extensive` command shows detailed information about all LSPs for which **R1** is the ingress, egress, or transit router, including all past state history and the reason why an LSP failed. All LSPs are up. The main two LSPs **lsp2-r1-to-r5** and **lsp1-r6-to-r0** have node-link protection as indicated by the **Node/Link protection desired** field in the ingress and transit sections of the output. In the ingress section of the output, the **Link-protection Up** field shows that **lsp2-r1-to-r5** has link protection up. In the transit section of the output, the **Type: Node/Link protected LSP** field shows that **lsp1-r6-to-r0** has node-link protection up, and in case of failure will use the bypass **LSP Bypass->10.0.12.14->10.0.24.2.**

Sample Output

```
user@R1> show rsvp interface
RSVP interface: 4 active
                  Active Subscr- Static      Available   Reserved    Highwater
Interface   State resv   iption  BW          BW          BW          mark
fe-0/1/0.0  Up         2    100%  100Mbps     100Mbps     0bps        0bps
fe-0/1/1.0  Up         1   100%  100Mbps     100Mbps     0bps        0bps
fe-0/1/2.0  Up         0   100%  100Mbps     100Mbps     0bps        0bps
so-0/0/3.0  Up         1   100%  155.52Mbps  155.52Mbps  0bps        0bps
```

Meaning

Sample output from **R1** for the `show rsvp interface` command shows four interfaces enabled with RSVP (**Up**). Interface **fe-0/1/0.0** has two active RSVP reservations (**Active resv**) that might indicate sessions for the two main LSPs, **lsp1-r6-to-r0** and **lsp2-r1-to-r5.** Interface **fe-0/1/0.0** is the connecting interface between R1 and R2, and both LSPs are configured with a strict path through **fe-0/1/0.0**. For more detailed information about what is happening on interface **fe-0/1/0.0**, issue the `show rsvp interface extensive` command.

Sample Output

```
user@R1> show rsvp interface extensive
RSVP interface: 3 active
fe-0/1/0.0 Index 67, State Ena/Up
  NoAuthentication, NoAggregate, NoReliable, LinkProtection
  HelloInterval 9(second)
  Address 10.0.12.13
  ActiveResv 2, PreemptionCnt 0, Update threshold 10%
  Subscription 100%,
  bc0 = ct0, StaticBW 100Mbps
  ct0: StaticBW 100Mbps, AvailableBW 100Mbps
    MaxAvailableBW 100Mbps = (bc0*subscription)
    ReservedBW [0] 0bps[1] 0bps[2] 0bps[3] 0bps[4] 0bps[5] 0bps[6] 0bps[7] 0bps
   Protection: On, Bypass: 2, LSP: 2, Protected LSP: 2, Unprotected LSP: 0
      2 Jul 14 14:49:40 New bypass Bypass->10.0.12.14
      1 Jul 14 14:49:34 New bypass Bypass->10.0.12.14->10.0.24.2
     Bypass: Bypass->10.0.12.14, State: Up, Type: LP, LSP: 0, Backup: 0
      3 Jul 14 14:49:42 Record Route:  10.0.17.14 10.0.27.1
      2 Jul 14 14:49:42 Up
```

```
      1 Jul 14 14:49:42 CSPF: computation result accepted
     Bypass: Bypass->10.0.12.14->10.0.24.2, State: Up, Type: NP, LSP: 2, Backup:0
      4 Jul 14 14:50:04 Record Route:  10.0.17.14 10.0.79.2 10.0.59.1 10.0.45.1
      3 Jul 14 14:50:04 Up
      2 Jul 14 14:50:04 CSPF: computation result accepted
      1 Jul 14 14:49:34 CSPF failed: no route toward 10.0.24.2
[...Output truncated...]
```

Meaning

Sample output from **R1** for the `show rsvp interface extensive` command shows more detailed information about the activity on all RSVP interfaces (**3**). However, only output for **fe-0/1/0.0** is shown. Protection is enabled (**Protection: On**), with two bypass paths (**Bypass: 2**) protecting two LSPs (**Protected LSP: 2**). All LSPs are protected, as indicated by the **Unprotected LSP: 0** field. The first bypass **Bypass->10.0.12.14**is a link protection bypass path (**Type: LP**), protecting the link between **R1** and **R2 fe-0/1/0.0.** The second bypass path **10.0.12.14->10.0.24.2** is a node-link protected LSP, avoiding **R2** in case of node failure.

Sample Output

```
user@R1> show rsvp session detail
Ingress RSVP: 2 sessions

192.168.4.1
  From: 192.168.1.1,  LSPstate: Up, ActiveRoute: 0
   LSPname: Bypass->10.0.12.14->10.0.24.2
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 102000
  Resv style: 1 SE, Label in: -, Label out: 102000
  Time left:    -, Since: Tue Jul 11 14:30:53 2006
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 60120 protocol 0
   Type: Bypass LSP
    Number of data route tunnel through: 2
    Number of RSVP session tunnel through: 0
  PATH rcvfrom: localclient
  Adspec: sent MTU 1500
  Path MTU: received 1500
  PATH sentto: 10.0.17.14 (fe-0/1/1.0) 336 pkts
  RESV rcvfrom: 10.0.17.14 (fe-0/1/1.0) 310 pkts
   Explct route: 10.0.17.14 10.0.79.2 10.0.59.1 10.0.45.1
  Record route: <self> 10.0.17.14 10.0.79.2 10.0.59.1 10.0.45.1
```

```
192.168.5.1
  From: 192.168.1.1,  LSPstate: Up, ActiveRoute: 0
   LSPname: lsp2-r1-to-r5, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 101872
  Resv style: 1 SE, Label in: -, Label out: 101872
  Time left:    -, Since: Tue Jul 11 14:28:28 2006
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 2 receiver 60118 protocol 0
   Node/Link protection desired
  Type: Node/Link protected LSP
  PATH rcvfrom: localclient
  Adspec: sent MTU 1500
  Path MTU: received 1500
  PATH sentto: 10.0.12.14 (fe-0/1/0.0) 344 pkts
  RESV rcvfrom: 10.0.12.14 (fe-0/1/0.0) 349 pkts
   Explct route: 10.0.12.14 10.0.24.2 10.0.45.2
  Record route: <self> 10.0.12.14 10.0.24.2 10.0.45.2
Total 2 displayed, Up 2, Down 0

Egress RSVP: 1 sessions

192.168.1.1
  From: 192.168.5.1,  LSPstate: Up, ActiveRoute: 0
  LSPname: r5-to-r1, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: -
  Resv style: 1 FF, Label in: 3, Label out: -
  Time left:  147, Since: Tue Jul 11 14:28:36 2006
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 29228 protocol 0
  PATH rcvfrom: 10.0.12.14 (fe-0/1/0.0) 348 pkts
  Adspec: received MTU 1500
  PATH sentto: localclient
  RESV rcvfrom: localclient
  Record route: 10.0.45.2 10.0.24.2 10.0.12.14 <self>
Total 1 displayed, Up 1, Down 0

Transit RSVP: 2 sessions

192.168.0.1
  From: 192.168.6.1,  LSPstate: Up, ActiveRoute: 0
```

```
    LSPname: lsp1-r6-to-r0, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 101952
  Resv style: 1 SE, Label in: 100464, Label out: 101952
  Time left:  134, Since: Tue Jul 11 14:31:38 2006
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 11131 protocol 0
    Node/Link protection desired
  Type: Node/Link protected LSP
  PATH rcvfrom: 10.0.16.2 (so-0/0/3.0) 488 pkts
  Adspec: received MTU 1500 sent MTU 1500
  PATH sentto: 10.0.12.14 (fe-0/1/0.0) 339 pkts
  RESV rcvfrom: 10.0.12.14 (fe-0/1/0.0) 343 pkts
    Explct route: 10.0.12.14 10.0.24.2 10.0.45.2 10.0.50.2
  Record route: 10.0.16.2 <self> 10.0.12.14 10.0.24.2 10.0.45.2 10.0.50.2

192.168.6.1
  From: 192.168.0.1,  LSPstate: Up, ActiveRoute: 0
  LSPname: r0-to-t6, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 3
  Resv style: 1 FF, Label in: 100448, Label out: 3
  Time left:  158, Since: Tue Jul 11 14:31:36 2006
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 23481 protocol 0
  PATH rcvfrom: 10.0.12.14 (fe-0/1/0.0) 344 pkts
  Adspec: received MTU 1500 sent MTU 1500
  PATH sentto: 10.0.16.2 (so-0/0/3.0) 337 pkts
  RESV rcvfrom: 10.0.16.2 (so-0/0/3.0) 310 pkts
  Explct route: 10.0.16.2
  Record route: 10.0.50.2 10.0.45.2 10.0.24.2 10.0.12.14 <self> 10.0.16.2
Total 2 displayed, Up 2, Down 0
```

Meaning

Sample output from **R1** shows detailed information about the RSVP sessions active on **R1**. All sessions are up, with two ingress sessions, one egress session, and two transit sessions.

Within the ingress section, the first session is a bypass path, as indicated by the **Type: Bypass LSP** field; and the second session is a protected LSP (**lsp2-r1-to-r5)** originating on **R1**, as indicated by the **Type: Node/Link protected LSP** field.

Conclusion

Multiprotocol Label Switching (MPLS) label-switched path (LSP) link protection and node-link protection are facility-based methods used to reduce the amount of time needed to reroute LSP traffic. These protection methods are often compared to fast reroute—the other Junos OS LSP protection method.

While fast reroute protects LSPs on a one-to-one basis, link protection and node-link protection protect multiple LSPs by using a single, logical bypass LSP. Link protection provides robust backup support for a link, node-link protection bypasses a node or a link, and both types of protection are designed to interoperate with other vendor equipment. Such functionality makes link protection and node-link protection excellent choices for scalability, redundancy, and performance in MPLS-enabled networks.

Related Information

For additional information about MPLS fast reroute and MPLS protection methods, see the following:

- *Junos User Guide*
- *Junos MPLS Applications Configuration Guide*
- Semeria, Chuck. *RSVP Signaling Extensions for MPLS Traffic Engineering*. White paper. 2002
- Semeria, Chuck. *IP Dependability: Network Link and Node Protection*. White paper. 2002
- RFC 4090, *Fast Reroute Extensions to RSVP-TE for LSP Tunnels*

Verify That Link Protection Is Up

IN THIS SECTION

- Purpose | 2352
- Action | 2353
- Meaning | 2354

Purpose

When you verify link protection, you must check that the bypass LSP is up. You can also check the number of LSPs protected by the bypass. In the network shown in Many-to-One or Link Protection, a bypass path should be up to protect the link between **R1** and **R2**, or next-hop **10.0.12.14**, and the two LSPs traversing the link, **lsp2-r1-to-r5** and **lsp1-r6-to-r0**.

Action

To verify link protection (many-to-one backup), enter the following Junos OS CLI operational mode commands on the ingress router:

```
user@host> show mpls lsp extensive
user@host> show rsvp session detail
user@host> show rsvp interface
```

Sample Output

command-name

```
user@R1> show mpls lsp extensive | no-more
Ingress LSP: 1 sessions

192.168.5.1
  From: 192.168.1.1, State: Up, ActiveRoute: 0, LSPname: lsp2-r1-to-r5
  ActivePath:  via-r2 (primary)
   Link protection desired
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary   via-r2           State: Up
    SmartOptimizeTimer: 180
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 3)
 10.0.12.14 S 10.0.24.2 S 10.0.45.2 S
     Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node  10=SoftPreempt):
          10.0.12.14(Label=101264) 10.0.24.2(Label=100736) 10.0.45.2(Label=3)
    6 Jun 16 14:06:33  Link-protection Up
    5 Jun 16 14:05:39 Selected as active path
    4 Jun 16 14:05:39 Record Route:  10.0.12.14(Label=101264) 10.0.24.2(Label=100736)
10.0.45.2(Label=3)
    3 Jun 16 14:05:39 Up
    2 Jun 16 14:05:39 Originate Call
    1 Jun 16 14:05:39 CSPF: computation result accepted
  Created: Fri Jun 16 14:05:38 2006
Total 1 displayed, Up 1, Down 0

[...Output truncated...]
```

```
Transit LSP: 2 sessions

192.168.0.1
  From: 192.168.6.1, LSPstate: Up, ActiveRoute: 0
   LSPname: lsp1-r6-to-r0, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 101296
  Resv style: 1 SE, Label in: 100192, Label out: 101296
  Time left:  116, Since: Mon Jun 19 10:26:32 2006
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 58739 protocol 0
  Link protection desired
  Type: Link protected LSP, using Bypass->10.0.12.14
      1 Jun 19 10:26:32 Link protection up, using Bypass->10.0.12.14
  PATH rcvfrom: 10.0.16.2 (so-0/0/3.0) 579 pkts
  Adspec: received MTU 1500 sent MTU 1500
  PATH sentto: 10.0.12.14 (fe-0/1/0.0) 474 pkts
  RESV rcvfrom: 10.0.12.14 (fe-0/1/0.0) 501 pkts
  Explct route: 10.0.12.14 10.0.24.2 10.0.45.2 10.0.50.2
  Record route: 10.0.16.2 <self> 10.0.12.14 10.0.24.2 10.0.45.2 10.0.50.2
[...Output truncated...]
```

Meaning

The sample output from ingress router **R1** shows that **lsp2-r1-to-r5** and **lsp1-r6-to-r0** have link protection up, and both LSPs are using the bypass path, **10.0.12.14**. However, the `show mpls lsp` command does not list the bypass path. For information about the bypass path, use the `show rsvp session` command.

Sample Output

```
user@R1> show rsvp session detail
Ingress RSVP: 2 sessions
192.168.2.1
  From: 192.168.1.1, LSPstate: Up, ActiveRoute: 0
   LSPname: Bypass->10.0.12.14
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 101456
  Resv style: 1 SE, Label in: -, Label out: 101456
  Time left:    -, Since: Fri May 26 18:38:09 2006
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 18709 protocol 0
```

```
    Type: Bypass LSP
      Number of data route tunnel through: 2
     Number of RSVP session tunnel through: 0
  PATH rcvfrom: localclient
  Adspec: sent MTU 1500
  Path MTU: received 1500
  PATH sentto: 10.0.17.14 (fe-0/1/1.0) 51939 pkts
  RESV rcvfrom: 10.0.17.14 (fe-0/1/1.0) 55095 pkts
   Explct route: 10.0.17.14 10.0.27.1
  Record route: <self> 10.0.17.14 10.0.27.1

192.168.5.1
  From: 192.168.1.1, LSPstate: Up, ActiveRoute: 0
  LSPname: lsp2-r1-to-r5, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 101264
  Resv style: 1 SE, Label in: -, Label out: 101264
  Time left:    -, Since: Fri Jun 16 14:05:39 2006
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 18724 protocol 0
  Link protection desired
  Type: Link protected LSP
  PATH rcvfrom: localclient
  Adspec: sent MTU 1500
  Path MTU: received 1500
  PATH sentto: 10.0.12.14 (fe-0/1/0.0) 8477 pkts
  RESV rcvfrom: 10.0.12.14 (fe-0/1/0.0) 8992 pkts
  Explct route: 10.0.12.14 10.0.24.2 10.0.45.2
  Record route: <self> 10.0.12.14 10.0.24.2 10.0.45.2
Total 2 displayed, Up 2, Down 0

Egress RSVP: 1 sessions

192.168.1.1
  From: 192.168.5.1, LSPstate: Up, ActiveRoute: 0
  LSPname: r5-to-r1, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: -
  Resv style: 1 FF, Label in: 3, Label out: -
  Time left:  159, Since: Mon May 22 22:08:16 2006
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 64449 protocol 0
  PATH rcvfrom: 10.0.17.14 (fe-0/1/1.0) 63145 pkts
```

```
  Adspec: received MTU 1500
  PATH sentto: localclient
  RESV rcvfrom: localclient
  Record route: 10.0.59.1 10.0.79.2 10.0.17.14 <self>
Total 1 displayed, Up 1, Down 0
Transit RSVP: 2 sessions
192.168.0.1
  From: 192.168.6.1, LSPstate: Up, ActiveRoute: 0
  LSPname: lsp1-r6-to-r0, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 101296
  Resv style: 1 SE, Label in: 100192, Label out: 101296
  Time left:  129, Since: Mon Jun 19 10:26:32 2006
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 58739 protocol 0
  Link protection desired
  Type: Link protected LSP
  PATH rcvfrom: 10.0.16.2 (so-0/0/3.0) 3128 pkts
  Adspec: received MTU 1500 sent MTU 1500
  PATH sentto: 10.0.12.14 (fe-0/1/0.0) 2533 pkts
  RESV rcvfrom: 10.0.12.14 (fe-0/1/0.0) 2685 pkts
  Explct route: 10.0.12.14 10.0.24.2 10.0.45.2 10.0.50.2
  Record route: 10.0.16.2 <self> 10.0.12.14 10.0.24.2 10.0.45.2 10.0.50.2

192.168.6.1
  From: 192.168.0.1, LSPstate: Up, ActiveRoute: 0
  LSPname: r0-to-r6, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 3
  Resv style: 1 FF, Label in: 100128, Label out: 3
  Time left:  143, Since: Thu May 25 12:30:26 2006
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 4111 protocol 0
  PATH rcvfrom: 10.0.17.14 (fe-0/1/1.0) 57716 pkts
  Adspec: received MTU 1500 sent MTU 1500
  PATH sentto: 10.0.16.2 (so-0/0/3.0) 54524 pkts
  RESV rcvfrom: 10.0.16.2 (so-0/0/3.0) 50534 pkts
  Explct route: 10.0.16.2
  Record route: 10.0.50.2 10.0.59.1 10.0.79.2 10.0.17.14 <self> 10.0.16.2
Total 2 displayed, Up 2, Down 0
```

Meaning

The sample output from ingress router **R1** shows the ingress, egress, and transit LSPs for **R1**. Some information is similar to that found in the `show mpls lsp` command. However, because link protection is an RSVP feature, information about bypass paths is provided. The bypass path appears as a separate RSVP ingress session for the protected interface, as indicated by the **Type** field.

The bypass path name is automatically generated. By default, the name appears as **Bypass > *interface-address***, where the interface address is the next downstream router's interface (**10.0.12.14**). The explicit route **10.0.17.14 10.0.27.1** for the session shows **R7** as the transit node and **R2** as the egress node.

Within the ingress RSVP section of the output, the LSP originating at **R1** (**lsp2-r1-to-r5)** is shown requesting link protection. Since a bypass path is in place to protect the downstream link, **lsp2-r1-to-r5** is associated with the bypass, as indicated by the **Link protected LSP** field.

The egress section of the output shows the return LSP **r5-to-r1**, which is not protected.

The transit section of the output shows link protection requested by **lsp1-r6-to-r0.** Since a bypass path is in place to protect the downstream link, **lsp1-r6-to-r0** is associated with the bypass, as indicated by the **Link protected LSP** field. Also included in the transit section of the output is the return LSP **r0-to-r6**, which is not protected.

Sample Output

```
user@R1> show rsvp interface
RSVP interface: 4 active
                  Active Subscr- Static      Available   Reserved    Highwater
Interface   State resv   iption  BW          BW          BW          mark
fe-0/1/0.0  Up          2    100%  100Mbps     100Mbps     0bps        35Mbps
fe-0/1/1.0  Up          1    100%  100Mbps     100Mbps     0bps        0bps
fe-0/1/2.0  Up          0    100%  100Mbps     100Mbps     0bps        0bps
so-0/0/3.0  Up          1    100%  155.52Mbps  155.52Mbps  0bps        0bps
```

Meaning

The sample output from ingress router **R1** shows the number of LSPs going through the interfaces configured on **R1**. The **Active resv** field shows the number of LSPs for each interface. For example, interface **fe-0/1/0.0** between **R1** and **R2** has two active reservations, **lsp1-r6-to-r0** and **lsp2-r1-to-r5**; interface **fe-0/1/1.0** between **R1** and **R7** has one, the bypass (**10.0.12.14**); interface **fe-0/1/2.0** between **R6** and **R1** has no LSP reservations; and interface **so-0/0/3.0** between **R6** and **R1** has one LSP reservation, **lsp1-r6-to-r0**.

Verify One-to-One Backup

IN THIS SECTION

- Purpose | 2358
- Action | 2358
- Meaning | 2359

Purpose

You can verify that one-to-one backup is established by examining the ingress router and the other routers in the network.

Action

To verify one-to-one backup, enter the following Junos OS CLI operational mode commands:

```
user@host> show mpls lsp ingress extensive
user@host> show rsvp session
```

Sample Output

command-name

The following sample output is from the ingress router **R1** :

```
user@R1> show mpls lsp ingress extensive
Ingress LSP: 1 sessions

192.168.5.1
  From: 192.168.1.1, State: Up, ActiveRoute: 0, LSPname: r1-to-r5
  ActivePath: via-r2 (primary)
   FastReroute desired
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary   via-r2          State: Up
    SmartOptimizeTimer: 180
```

```
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 3)
 10.0.12.14 S 10.0.24.2 S 10.0.45.2 S
     Received RRO  (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt):
           10.0.12.14(flag=9) 10.0.24.2(flag=1) 10.0.45.2
     8 May 11 14:51:46 Fast-reroute Detour Up
     7 May 11 14:50:55 Record Route:  10.0.12.14(flag=9) 10.0.24.2(flag=1)  10.0.45.2
    6 May 11 14:50:55 Record Route:  10.0.12.14(flag=9) 10.0.24.2 10.0.45.2
    5 May 11 14:50:52 Selected as active path
    4 May 11 14:50:52 Record Route:  10.0.12.14 10.0.24.2 10.0.45.2
    3 May 11 14:50:52 Up
    2 May 11 14:50:52 Originate Call
    1 May 11 14:50:52 CSPF: computation result accepted
  Created: Thu May 11 14:50:52 2006
Total 1 displayed, Up 1, Down 0
```

Meaning

The sample output from **R1** shows that the **FastReroute desired** object was included in the Path messages for the LSP, allowing **R1** to select the active path for the LSP and establish a detour path to avoid **R2**.

In line 8, **Fast-reroute Detour Up** shows that the detour is operational. Lines 6 and 7 indicate that transit routers **R2** and **R4** have established their detour paths.

R2, **10.0.12.14**, includes **(flag=9)**, indicating that node protection is available for the downstream node and link. **R4**, **10.0.24.2**, includes **(flag=1)**, indicating that link protection is available for the next downstream link. In this case, **R4** can protect only the downstream link because the node is the egress router **R5**, which cannot be protected. For more information about flags, see the *Junos User Guide*.

The output for the `show mpls lsp extensive` command does not show the actual path of the detour. To see the actual links used by the detour paths, you must use the `show rsvp session ingress detail` command.

Sample Output

The following sample output is from the ingress router **R1** in the network shown in One-to-One Backup Detours.

```
user@R1> show rsvp session ingress detail
Ingress RSVP: 1 sessions

192.168.5.1
  From: 192.168.1.1,  LSPstate: Up,  ActiveRoute: 0
```

```
  LSPname: r1-to-r5, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 100848
  Resv style: 1 FF, Label in: -, Label out: 100848
  Time left:    -, Since: Thu May 11 14:17:15 2006
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 9228 protocol 0
   FastReroute desired
  PATH rcvfrom: localclient
  Adspec: sent MTU 1500
  Path MTU: received 1500
  PATH sentto: 10.0.12.14 (fe-0/1/0.0) 35 pkts
  RESV rcvfrom: 10.0.12.14 (fe-0/1/0.0) 25 pkts
  Explct route: 10.0.12.14 10.0.24.2 10.0.45.2
  Record route: <self> 10.0.12.14 10.0.24.2 10.0.45.2
     Detour is Up
    Detour Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
    Detour adspec: sent MTU 1500
    Path MTU: received 1500
    Detour PATH sentto: 10.0.17.14 (fe-0/1/1.0) 23 pkts
    Detour RESV rcvfrom: 10.0.17.14 (fe-0/1/1.0) 20 pkts
     Detour Explct route: 10.0.17.14 10.0.79.2 10.0.59.1
    Detour Record route: <self> 10.0.17.14 10.0.79.2 10.0.59.1
    Detour Label out: 100848
Total 1 displayed, Up 1, Down 0
```

Meaning

The sample output from **R1** shows the RSVP session of the main LSP. The detour path is established, **Detour is Up**. The physical path of the detour is displayed in **Detour Explct route**. The detour path uses **R7** and **R9** as transit routers to reach **R5**, the egress router.

Sample Output

The following sample output is from the first transit router R2 in the network shown in One-to-One Backup Detours:

```
user@R2> show rsvp session transit detail
Transit RSVP: 1 sessions

192.168.5.1
```

```
  From: 192.168.1.1, LSPstate: Up, ActiveRoute: 1
  LSPname: r1-to-r5, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 100448
  Resv style: 1 FF, Label in: 100720, Label out: 100448
  Time left:  126, Since: Wed May 10 16:12:21 2006
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 5 receiver 9216 protocol 0
   FastReroute desired
  PATH rcvfrom: 10.0.12.13 (fe-0/1/0.0) 173 pkts
  Adspec: received MTU 1500 sent MTU 1500
  PATH sentto: 10.0.24.2 (so-0/0/1.0) 171 pkts
  RESV rcvfrom: 10.0.24.2 (so-0/0/1.0) 169 pkts
  Explct route: 10.0.24.2 10.0.45.2
  Record route: 10.0.12.13 <self> 10.0.24.2 10.0.45.2
     Detour is Up
    Detour Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
    Detour adspec: received MTU 1500 sent MTU 1500
    Path MTU: received 1500
    Detour PATH sentto: 10.0.27.2 (so-0/0/3.0) 169 pkts
    Detour RESV rcvfrom: 10.0.27.2 (so-0/0/3.0) 167 pkts
     Detour Explct route: 10.0.27.2 10.0.79.2 10.0.59.1
    Detour Record route: 10.0.12.13 <self> 10.0.27.2 10.0.79.2 10.0.59.1
    Detour Label out: 100736
Total 1 displayed, Up 1, Down 0
```

Meaning

The sample output from **R2** shows the detour is established (**Detour is Up**) and avoids **R4**, and the link connecting **R4** and **R5** (**10.0.45.2**). The detour path is through **R7** (**10.0.27.2**) and **R9** (**10.0.79.2**) to **R5** (**10.0.59.1**), which is different from the explicit route for the detour from **R1**. **R1** has the detour passing through the **10.0.17.14** link on **R7**, while **R1** is using the **10.0.27.2** link. Both detours merge at **R9** through the **10.0.79.2** link to **R5** (**10.0.59.1**).

Sample Output

The following sample output is from the second transit router R4 in the network shown in One-to-One Backup Detours:

```
user@R4> show rsvp session transit detail
Transit RSVP: 1 sessions
```

```
192.168.5.1
  From: 192.168.1.1,  LSPstate: Up, ActiveRoute: 1
   LSPname: r1-to-r5, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 3
  Resv style: 1 FF, Label in: 100448, Label out: 3
  Time left:  155, Since: Wed May 10 16:15:38 2006
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 5 receiver 9216 protocol 0
   FastReroute desired
  PATH rcvfrom: 10.0.24.1 (so-0/0/1.0) 178 pkts
  Adspec: received MTU 1500 sent MTU 1500
  PATH sentto: 10.0.45.2 (so-0/0/2.0) 178 pkts
  RESV rcvfrom: 10.0.45.2 (so-0/0/2.0) 175 pkts
  Explct route: 10.0.45.2
  Record route: 10.0.12.13 10.0.24.1 <self> 10.0.45.2
     Detour is Up
    Detour Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
    Detour adspec: received MTU 1500 sent MTU 1500
    Path MTU: received 1500
    Detour PATH sentto: 10.0.49.2 (so-0/0/3.0) 176 pkts
    Detour RESV rcvfrom: 10.0.49.2 (so-0/0/3.0) 175 pkts
     Detour Explct route: 10.0.49.2 10.0.59.1
     Detour Record route: 10.0.12.13 10.0.24.1 <self> 10.0.49.2 10.0.59.1
    Detour Label out: 100352
Total 1 displayed, Up 1, Down 0
```

Meaning

The sample output from **R4** shows the detour is established (**Detour is Up**) and avoids the link connecting **R4** and **R5** (**10.0.45.2**). The detour path is through R9 (**10.0.49.2**) to **R5** (**10.0.59.1**). Some of the information is similar to that found in the output for **R1** and **R2**. However, the explicit route for the detour is different, going through the link connecting **R4** and **R9** (**so-0/0/3** or **10.0.49.2.**

Sample Output

The following sample output is from **R7**, which is used in the detour path in the network shown in One-to-One Backup Detours:

```
user@R7> show rsvp session transit detail
Transit RSVP: 1 sessions, 1 detours

192.168.5.1
  From: 192.168.1.1,  LSPstate: Up, ActiveRoute: 1
   LSPname: r1-to-r5, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 100368
  Resv style: 1 FF, Label in: 100736, Label out: 100368
  Time left:  135, Since: Wed May 10 16:14:42 2006
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 5 receiver 9216 protocol 0
   Detour branch from 10.0.27.1, to skip 192.168.4.1, Up
    Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
    Adspec: received MTU 1500
    Path MTU: received 0
    PATH rcvfrom: 10.0.27.1 (so-0/0/3.0) 179 pkts
    Adspec: received MTU 1500 sent MTU 1500
    PATH sentto: 10.0.79.2 (so-0/0/1.0) 177 pkts
    RESV rcvfrom: 10.0.79.2 (so-0/0/1.0) 179 pkts
    Explct route: 10.0.79.2 10.0.59.1
     Record route: 10.0.12.13 10.0.27.1 <self> 10.0.79.2 10.0.59.1
     Label in: 100736, Label out: 100368
   Detour branch from 10.0.17.13, to skip 192.168.2.1, Up
    Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
    Adspec: received MTU 1500
    Path MTU: received 0
    PATH rcvfrom: 10.0.17.13 (fe-0/1/1.0) 179 pkts
    Adspec: received MTU 1500
    PATH sentto: 10.0.79.2 (so-0/0/1.0) 0 pkts
    RESV rcvfrom: 10.0.79.2 (so-0/0/1.0) 0 pkts
    Explct route: 10.0.79.2 10.0.59.1
     Record route: 10.0.17.13 <self> 10.0.79.2 10.0.59.1
     Label in: 100752, Label out: 100368
Total 1 displayed, Up 1, Down 0
```

Meaning

The sample output from **R7** shows the same information as for a regular transit router used in the primary path of the LSP: the ingress address (**192.168.1.1**), the egress address (**192.168.5.1**), and the name of the LSP (**r1-to-r5**). Two detour paths are displayed; the first to avoid **R4** (**192.168.4.1**) and the second to avoid **R2** (**192.168.2.1**). Because **R7** is used as a transit router by **R2** and **R4**, **R7** can merge the detour paths together as indicated by the identical **Label out** value (**100368**) for both detour paths. Whether **R7** receives traffic from **R4** with a label value of **100736** or from **R2** with a label value of **100752**, **R7** forwards the packet to **R5** with a label value of **100368**.

Sample Output

The following sample output is from R9, which is a router used in the detour path in the network shown in One-to-One Backup Detours:

```
user@R9> show rsvp session transit detail
Transit RSVP: 1 sessions, 1 detours

192.168.5.1
  From: 192.168.1.1, LSPstate: Up, ActiveRoute: 1
  LSPname: r1-to-r5, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 3
  Resv style: 1 FF, Label in: 100352, Label out: 3
  Time left:  141, Since: Wed May 10 16:16:40 2006
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 5 receiver 9216 protocol 0
   Detour branch from 10.0.49.1, to skip 192.168.5.1, Up
    Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
    Adspec: received MTU 1500
    Path MTU: received 0
    PATH rcvfrom: 10.0.49.1 (so-0/0/3.0) 183 pkts
    Adspec: received MTU 1500 sent MTU 1500
    PATH sentto: 10.0.59.1 (so-0/0/0.0) 182 pkts
    RESV rcvfrom: 10.0.59.1 (so-0/0/0.0) 183 pkts
    Explct route: 10.0.59.1
     Record route: 10.0.12.13 10.0.24.1 10.0.49.1 <self> 10.0.59.1
    Label in: 100352, Label out: 3
   Detour branch from 10.0.27.1, to skip 192.168.4.1, Up
    Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
    Adspec: received MTU 1500
    Path MTU: received 0
```

```
    Detour branch from 10.0.17.13, to skip 192.168.2.1, Up
     Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
     Adspec: received MTU 1500
     Path MTU: received 0
     PATH rcvfrom: 10.0.79.1 (so-0/0/1.0) 181 pkts
     Adspec: received MTU 1500
     PATH sentto: 10.0.59.1 (so-0/0/0.0) 0 pkts
     RESV rcvfrom: 10.0.59.1 (so-0/0/0.0) 0 pkts
      Explct route: 10.0.59.1
     Record route: 10.0.12.13 10.0.27.1 10.0.79.1 <self> 10.0.59.1
      Label in: 100368,  Label out: 3
Total 1 displayed, Up 1, Down 0
```

Meaning

The sample output from **R9** shows that **R9** is the penultimate router for the detour path, the explicit route includes only the egress link address (**10.0.59.1**), and the **Label out** value (**3**) indicates that **R9** has performed penultimate-hop label popping. Also, the detour branch from **10.0.27.1** does not include path information because **R7** has merged the detour paths from **R2** and **R4**. Notice that the **Label out** value in the detour branch from **10.0.17.13** is **100368**, the same value as the **Label out** value on **R7**.

Sample Output

The following sample output is from the egress router R5 in the network shown in One-to-One Backup Detours:

```
user@R5> show rsvp session egress detail
Egress RSVP: 1 sessions, 1 detours

192.168.5.1
  From: 192.168.1.1, LSPstate: Up, ActiveRoute: 0
  LSPname: r1-to-r5, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: -
  Resv style: 1 FF, Label in: 3, Label out: -
  Time left:  119, Since: Thu May 11 14:44:31 2006
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 9230 protocol 0
   FastReroute desired
  PATH rcvfrom: 10.0.45.1 (so-0/0/2.0) 258 pkts
  Adspec: received MTU 1500
```

```
  PATH sentto: localclient
  RESV rcvfrom: localclient
   Record route: 10.0.12.13 10.0.24.1 10.0.45.1 <self>
   Detour branch from 10.0.49.1, to skip 192.168.5.1, Up
    Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
    Adspec: received MTU 1500
    Path MTU: received 0
   Detour branch from 10.0.27.1, to skip 192.168.4.1, Up
    Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
    Adspec: received MTU 1500
    Path MTU: received 0
   Detour branch from 10.0.17.13, to skip 192.168.2.1, Up
    Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
    Adspec: received MTU 1500
    Path MTU: received 0
    PATH rcvfrom: 10.0.59.2 (so-0/0/0.0) 254 pkts
    Adspec: received MTU 1500
    PATH sentto: localclient
    RESV rcvfrom: localclient
    Record route: 10.0.12.13 10.0.24.1 10.0.49.1 10.0.59.2 <self>
     Label in: 3, Label out: -
Total 1 displayed, Up 1, Down 0
```

Meaning

The sample output from **R5** shows the main LSP in the **Record route** field and the detours through the network.

Verify That the Primary Path Is Operational

IN THIS SECTION

- Purpose | **2367**
- Action | **2367**
- Meaning | **2368**

Purpose

Primary paths must always be used in the network if they are available, therefore an LSP always moves back to the primary path after a failure, unless the configuration is adjusted. For more information on adjusting the configuration to prevent a failed primary path from reestablishing, see "Preventing Use of a Path That Previously Failed" on page 367.

Action

To verify that the primary path is operational, enter the following Junos OS command-line interface (CLI) operational mode commands:

```
user@host> show mpls lsp extensive ingress
user@host> show rsvp interface
```

Sample Output 1

command-name

```
user@R1> show mpls lsp extensive ingress
Ingress LSP: 1 sessions

192.168.5.1
  From: 192.168.1.1, State: Up, ActiveRoute: 0, LSPname: r1-to-r5
   ActivePath: via-r2 (primary)
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary   via-r2          State: Up
     Priorities: 6 6
     Bandwidth: 35Mbps
    SmartOptimizeTimer: 180
     Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 11)
 10.0.12.14 S 10.0.24.2 S
     Received RRO  (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt):
           10.0.12.14 10.0.24.2
    5 Apr 29 14:40:43 Selected as active path
    4 Apr 29 14:40:43 Record Route:  10.0.12.14 10.0.24.2
    3 Apr 29 14:40:43 Up
    2 Apr 29 14:40:43 Originate Call
    1 Apr 29 14:40:43 CSPF: computation result accepted
```

```
  Standby   via-r7           State: Dn
    SmartOptimizeTimer: 180
        No computed ERO.
  Created: Sat Apr 29 14:40:43 2006
Total 1 displayed, Up 1, Down 0
```

Sample Output 2

command-name

```
user@R1> show rsvp interface
RSVP interface: 3 active
                 Active Subscr- Static      Available   Reserved    Highwater
Interface   State resv   iption  BW          BW          BW          mark
fe-0/1/0.0  Up         2   100%   100Mbps      100Mbps     0bps        0bps
fe-0/1/1.0  Up         1   100%   100Mbps      100Mbps     0bps        0bps
so-0/0/3.0  Up         1   100%   155.52Mbps   155.52Mbps  0bps        0bps
```

Meaning

Sample output 1 shows that the LSP is operational and is using the primary path (**via-r2**) with **R2** (**10.0.12.14**) and **R4** (**10.0.24.2**) as transit routers. The priority values are the same for setup and hold, **6 6**. Priority 0 is the highest (best) priority and 7 is the lowest (worst) priority. The Junos OS default for setup and hold priority is 7:0. Unless some LSPs are more important than others, preserving the default is a good practice. Configuring a setup priority that is better than the hold priority is not allowed, resulting in a failed commit in order to avoid preemption loops.

Verify That the Secondary Path Is Established

IN THIS SECTION

- Purpose | **2369**
- Action | **2369**
- Meaning | **2371**

Purpose

When the secondary path is configured with the `standby` statement, the secondary path should be *up* but *not active*; it will become active if the primary path fails. A secondary path configured without the `standby` statement will not come up unless the primary path fails. To test that the secondary path is correctly configured and would come up if the primary path were to fail, you must deactivate a link or node critical to the primary path, then issue the `show mpls lsp` *`lsp-path-name`* `extensive` command.

Action

To verify that the secondary path is established, enter the following Junos OS CLI operational mode command:

Sample Output

```
user@R1> show mpls lsp extensive
```

Sample Output

command-name

The following sample output shows a correctly configured secondary path before and after it comes up. In the example, interface **fe-0/1/0** on **R2** is deactivated, which brings down the primary path **via-r2**. The ingress router **R1** switches traffic to the secondary path **via-r7**.

```
user@R1> show mpls lsp extensive
Ingress LSP: 1 sessions

192.168.5.1
  From: 192.168.1.1, State: Up, ActiveRoute: 0, LSPname: r1-to-r5
  ActivePath: via-r2 (primary)
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
  *Primary   via-r2           State: Up
    Priorities: 6 6
    Bandwidth: 35Mbps
    SmartOptimizeTimer: 180
     Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 3)
 10.0.12.14 S 10.0.24.2 S 10.0.45.2 S
```

```
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt):
          10.0.12.14 10.0.24.2 10.0.45.2
    5 Apr 29 14:40:43 Selected as active path
    4 Apr 29 14:40:43 Record Route:  10.0.12.14 10.0.24.2
    3 Apr 29 14:40:43 Up
    2 Apr 29 14:40:43 Originate Call
    1 Apr 29 14:40:43 CSPF: computation result accepted
   Secondary via-r7           State: Dn
    SmartOptimizeTimer: 180
        No computed ERO.
  Created: Sat Apr 29 14:40:43 2006
Total 1 displayed, Up 1, Down 0

[edit interfaces]
user@R2# deactivate fe-0/1/0

[edit interfaces]
user@R2# show
inactive: fe-0/1/0 {
    unit 0 {
        family inet {
            address 10.0.12.14/30;
        }
        family iso;
        family mpls;
    }
}

user@R1> show mpls lsp name r1-to-r4 extensive
Ingress LSP: 1 sessions

192.168.4.1
  From: 192.168.1.1, State: Up, ActiveRoute: 0, LSPname: r1-to-r4
   ActivePath: via-r7 (secondary)
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
   Primary   via-r2           State: Dn
    Priorities: 6 6
    Bandwidth: 35Mbps
    SmartOptimizeTimer: 180
    Will be enqueued for recomputation in 14 second(s).
   10 Apr 29 14:52:33 CSPF failed: no route toward 10.0.12.1 4[21 times]
    9 Apr 29 14:42:48 Clear Call
```

```
    8 Apr 29 14:42:48 Deselected as active
    7 Apr 29 14:42:48 Session preempted
    6 Apr 29 14:42:48 Down
    5 Apr 29 14:40:43 Selected as active path
    4 Apr 29 14:40:43 Record Route:  10.0.12.14 10.0.24.2
    3 Apr 29 14:40:43 Up
    2 Apr 29 14:40:43 Originate Call
    1 Apr 29 14:40:43 CSPF: computation result accepted
  *Standby   via-r7           State: Up
    SmartOptimizeTimer: 180
     Computed ERO  (S [L] denotes strict [loose] hops): (CSPF metric: 11)
 10.0.17.14 S 10.0.47.1 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt):
          10.0.17.14 10.0.47.1
    5 Apr 29 14:42:48 Selected as active path
    4 Apr 29 14:41:12 Record Route:  10.0.17.14 10.0.47.1
    3 Apr 29 14:41:12 Up
    2 Apr 29 14:41:12 Originate Call
    1 Apr 29 14:41:12 CSPF: computation result accepted
  Created: Sat Apr 29 14:40:43 2006
Total 1 displayed, Up 1, Down 0
```

Meaning

The sample output from egress router **R1** shows a correctly configured standby secondary path in a down state because the primary path is still up. Upon deactivation of an interface (**interface fe-0/1/0** on **R2**) critical to the primary path, the primary path **via-r2** goes down and the standby secondary path **via-r7** comes up, allowing **R1** to switch traffic to the standby secondary path.

Verifying the Physical Layer

IN THIS SECTION

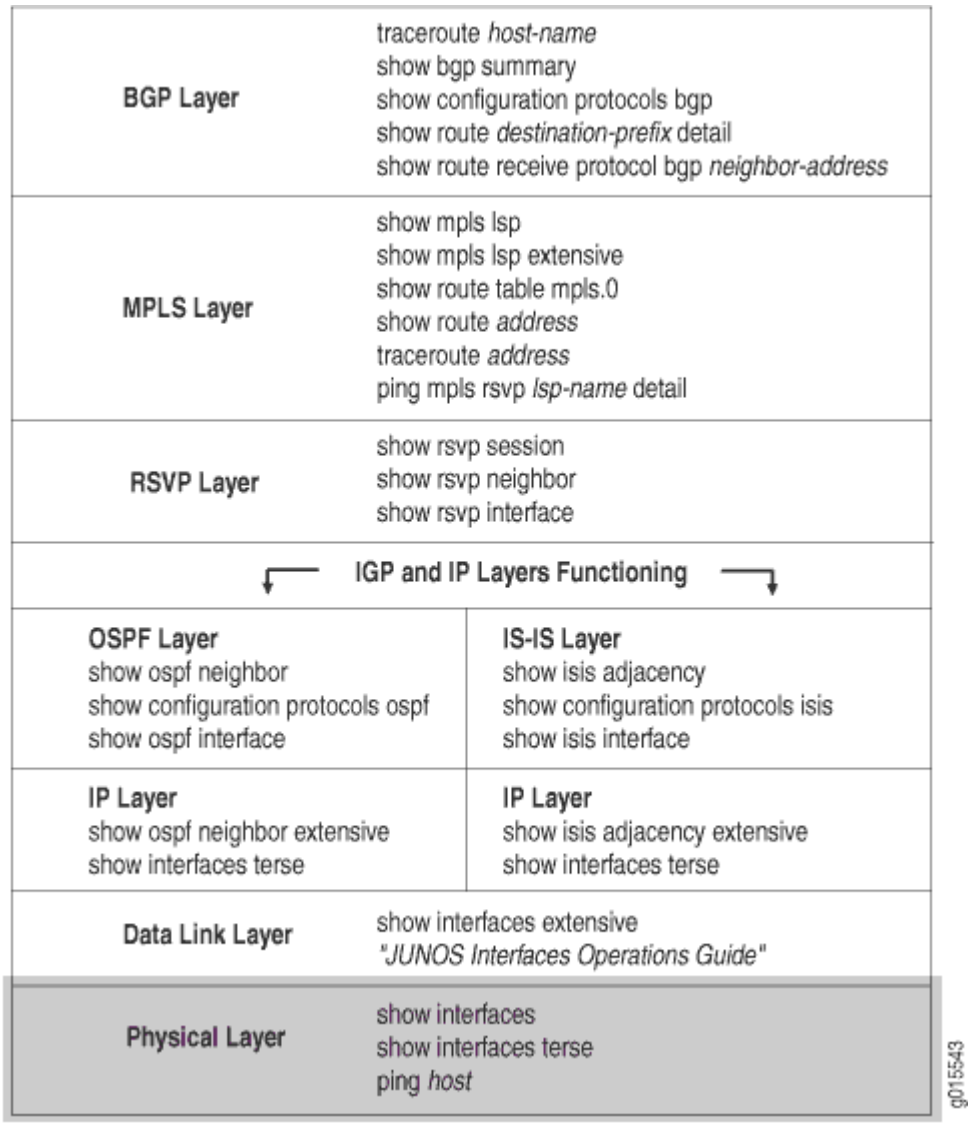
- Verify the LSP | **2374**
- Verify Router Connection | **2376**
- Verify Interfaces | **2377**
- Take Appropriate Action | **2379**
- Verify the LSP Again | **2380**

Purpose

After you have configured the LSP, issued the `show mpls lsp extensive` command, and determined that there is an error, you can start investigating the problem at the physical layer of the network.

Figure 160 on page 2372 illustrates the physical layer of the layered MPLS model.

Figure 160: Verifying the Physical Layer

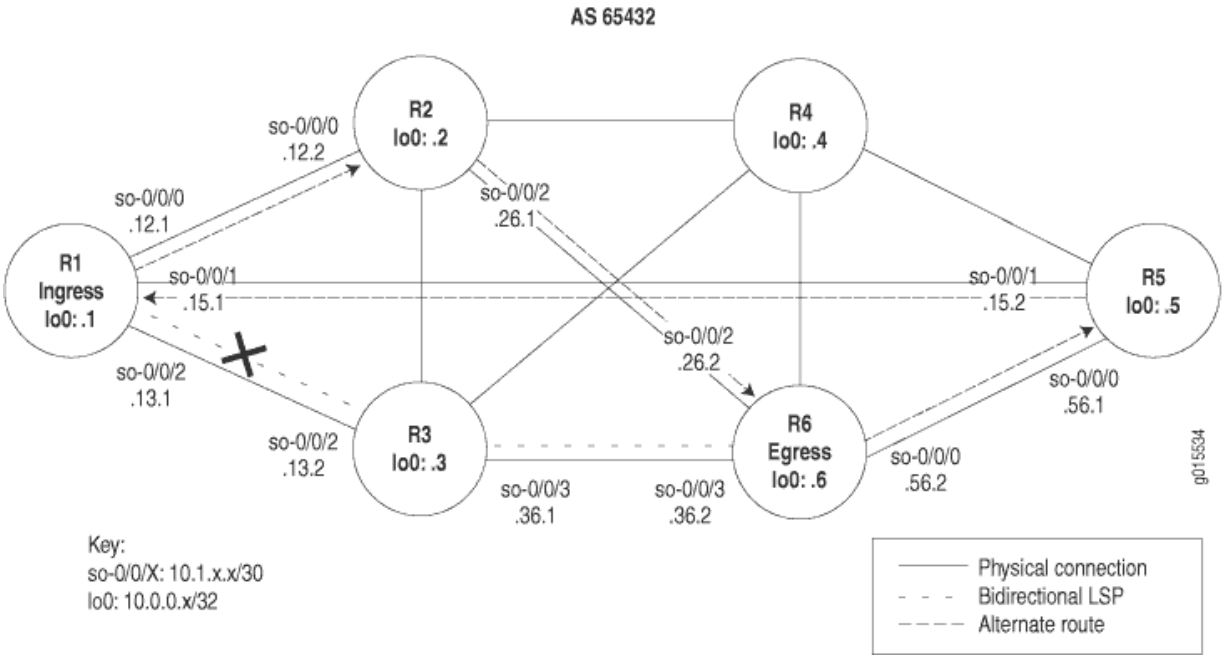


With this layer, you must ensure that the routers are connected, and that the interfaces are up and configured correctly on the ingress, egress, and transit routers.

If the network is not functioning at this layer, the label-switched path (LSP) does not work as configured.

Figure 161 on page 2373 illustrates the MPLS network and the problem described in this topic.

Figure 161: MPLS Network Broken at the Physical Layer



The network shown in Figure 161 on page 2373 is a fully meshed configuration where every directly connected interface can receive and send packets to every other similar interface. The LSP in this network is configured to run from ingress router **R1**, through transit router **R3**, to egress router **R6**. In addition, a reverse LSP is configured to run from **R6** through **R3** to **R1**, creating bidirectional traffic.

However, in this example, traffic does not use the configured LSP. Instead traffic uses the alternate route from **R1** through **R2** to **R6**, and in the reverse direction, from **R6** through **R5 to R1**.

When you become aware of a situation where an alternate route is used rather than the configured LSP, verify that the physical layer is functioning correctly. You might find that routers are not connected, or that interfaces are not up and configured correctly on the ingress, egress, or transit routers.

The cross shown in Figure 161 on page 2373 indicates where the LSP is broken because of a configuration error on ingress router **R1**.

To check the physical layer, follow these steps:

Verify the LSP

IN THIS SECTION

- Purpose | 2374
- Action | 2374
- Meaning | 2376

Purpose

Typically, you use the `show mpls lsp extensive` command to verify the LSP. However, for quick verification of the LSP state, use the `show mpls lsp` command. If the LSP is down, use the **extensive** option (`show mpls lsp extensive`) as a follow-up. If your network has numerous LSPs, you might consider specifying the name of the LSP, using the **name** option (`show mpls lsp name` ***name*** or `show mpls lsp name` ***name*** **extensive).**

Action

To determine whether the LSP is up, enter the following command from the ingress router:

```
user@ingress-router> show mpls lsp extensive
```

Sample Output

command-name

```
user@R1> show mpls lsp extensive
Ingress LSP: 1 sessions

10.0.0.6
  From: 10.0.0.1, State: Up, ActiveRoute: 1, LSPname: R1-to-R6
  ActivePath:  (primary)
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                    State: Up
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 20)
 10.1.12.2 S 10.1.26.2  S
```

```
     Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt):
           10.1.12.2 10.1.26.2
    99 Sep 18 14:19:04 CSPF: computation result accepted
     98 Sep 18 14:19:04 CSPF: link down/deleted
    10.1.13.1(R1.00/10.0.0.1)->10.1.13.2(R3.00/10.0.0.3)
    97 Sep 18 14:19:01 Record Route:  10.1.12.2 10.1.26.2
    96 Sep 18 14:19:01 Up
    95 Sep 18 14:19:01 Clear Call
    94 Sep 18 14:19:01 CSPF: computation result accepted
    93 Sep 18 14:19:01 MPLS label allocation failure
    92 Sep 18 14:19:01 Down
    91 Aug 17 12:22:52 Selected as active path
    90 Aug 17 12:22:52 Record Route:  10.1.13.2 10.1.36.2
    89 Aug 17 12:22:52 Up
    [...Output truncated...]
  Created: Sat Jul 10 18:18:44 2004
Total 1 displayed, Up 1, Down 0

Egress LSP: 1 sessions

10.0.0.1
  From: 10.0.0.6, LSPstate: Up, ActiveRoute: 0
  LSPname: R6-to-R1, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: -
  Resv style: 1 FF, Label in: 3, Label out: -
  Time left:  144, Since: Tue Aug 17 12:23:14 2004
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 39024 protocol 0
  PATH rcvfrom: 10.1.15.2 (so-0/0/1.0) 67333 pkts
  Adspec: received MTU 1500
  PATH sentto: localclient
  RESV rcvfrom: localclient
  Record route:  10.1.56.2 10.1.15.2 <self>
Total 1 displayed, Up 1, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Meaning

The sample output from ingress router **R1** shows that the LSP is using an alternate path rather than the configured path. The configured path for the LSP is **R1** through **R3** to **R6,** and for the reverse LSP, **R6** through **R3** to **R1**. The alternate path used by the LSP is **R1** through **R2** to **R6,** and for the reverse LSP, **R6 t**hrough **R5** to **R1**.

Verify Router Connection

IN THIS SECTION

- Purpose | **2376**
- Action | **2376**
- Meaning | **2377**

Purpose

Confirm that the appropriate ingress, transit, and egress routers are functioning by examining whether the packets have been received and transmitted with 0% packet loss.

Action

To determine that the routers are connected, enter the following command from the ingress and transit routers:

```
user@host>              ping host
```

Sample Output

command-name

```
user@R1> ping 10.0.0.3 count 3
PING 10.0.0.3 (10.0.0.3): 56 data bytes
64 bytes from 10.0.0.3: icmp_seq=0 ttl=254 time=0.859 ms
64 bytes from 10.0.0.3: icmp_seq=1 ttl=254 time=0.746 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=254 time=0.776 ms
```

```
--- 10.0.0.3 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.746/0.794/0.859/0.048 ms

user@R3>  ping 10.0.0.6 count 3
PING 10.0.0.6 (10.0.0.6): 56 data bytes
64 bytes from 10.0.0.6: icmp_seq=0 ttl=255 time=0.968 ms
64 bytes from 10.0.0.6: icmp_seq=1 ttl=255 time=3.221 ms
64 bytes from 10.0.0.6: icmp_seq=2 ttl=255 time=0.749 ms

--- 10.0.0.6 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.749/1.646/3.221/1.117 ms
```

Meaning

The sample output shows that ingress router **R1** is receiving packets from transit router **R3**, and that the transit router is receiving packets from the egress router. Therefore, the routers in the LSP are connected.

Verify Interfaces

IN THIS SECTION

- Purpose | **2377**
- Action | **2378**
- Meaning | **2378**

Purpose

Confirm that the interfaces are configured correctly with the `family mpls` statement.

Action

To determine that the relevant interfaces are up and configured correctly, enter the following commands from the ingress, transit, and egress routers:

```
user@host> show interfaces terse
user@host> show configuration interfaces type-fpc/pic/port
```

Sample Output

command-name

```
user@R1> show interfaces so* terse
Interface               Admin Link Proto Local                 Remote
so-0/0/0                up    up
so-0/0/0.0              up    up   inet  10.1.12.1/30
                                  iso
                                  mpls
so-0/0/1                up    up
so-0/0/1.0              up    up   inet  10.1.15.1/30
                                  iso
                                  mpls
so-0/0/2                up    up
so-0/0/2.0              up    up   inet  10.1.13.1/30
                                  iso   <<< family mpls is missing
so-0/0/3                up    down

user@R1>  show configuration interfaces so-0/0/2
unit 0 {
    family inet {
        address 10.1.13.1/30;
    }
    family iso; <<<  family mpls is missing
}
```

Meaning

The sample output shows that interface **so-0/0/2.0** on the ingress router does not have the `family mpls` statement configured at the [**edit interfaces *type-fpc/pic/port***] hierarchy level, indicating that the

interface is incorrectly configured to support the LSP. The LSP is configured correctly at the [**edit protocols mpls**] hierarchy level.

The output from the transit and egress routers (not shown) shows that the interfaces on those routers are configured correctly.

Take Appropriate Action

IN THIS SECTION

- Problem | 2379
- Solution | 2379

Problem

Description

Depending on the error you encountered in your investigation, you must take the appropriate action to correct the problem. In the example below, the `family mpls` statement, which was missing, is included in the configuration of ingress router **R1**.

Solution

To correct the error in this example, enter the following commands:

```
[edit interfaces                type-fpc/pic/port]
user@R1# set family mpls
user@R1# show
user@R1# commit
```

Sample Output

```
[edit interfaces so-0/0/2 unit 0]
user@R1# set family mpls

[edit interfaces so-0/0/2 unit 0]
user@R1# show
```

```
family inet {
    address 10.1.13.1/30;
}
family iso;
family mpls;

[edit interfaces so-0/0/2 unit 0]
user@R1# commit
commit complete
```

Meaning

The sample output from ingress router **R1** shows that the `family mpls` statement is configured correctly for interface **so-0/0/2.0**, and that the LSP is now functioning as originally configured.

Verify the LSP Again

IN THIS SECTION

- Purpose | 2380
- Action | 2380
- Meaning | 2382

Purpose

After taking the appropriate action to correct the error, the LSP needs to be checked again to confirm that the problem in the physical layer has been resolved.

Action

To verify that the LSP is up and traversing the network as expected, enter the following command:

```
user@host> show mpls lsp extensive
```

Sample Output 1

command-name

```
user@R1> show mpls lsp extensive
Ingress LSP: 1 sessions

10.0.0.6
  From: 10.0.0.1, State: Up, ActiveRoute: 1, LSPname: R1-to-R6
  ActivePath:  (primary)
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                   State: Up
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 20)
  10.1.13.2 S 10.1.36.2 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt):
          10.1.13.2 10.1.36.2
   112 Sep 21 16:27:33 Record Route:  10.1.13.2 10.1.36.2
   111 Sep 21 16:27:33 Up
   110 Sep 21 16:27:33 CSPF: computation result accepted
   109 Sep 21 16:27:33 CSPF: link down/deleted 10.1.12.1(R1.00/10.0.0.1)-
>10.1.12.2(R2.00/10.0.0.2)
   108 Sep 21 16:27:33 CSPF: link down/deleted 10.1.15.1(R1.00/10.0.0.1)-
>10.1.15.2(R5.00/10.0.0.5)
   [Output truncated...]
  Created: Sat Jul 10 18:18:44 2004
Total 1 displayed, Up 1, Down 0

Egress LSP: 1 sessions

10.0.0.1
  From: 10.0.0.6, LSPstate: Up, ActiveRoute: 0
  LSPname: R6-to-R1, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: -
  Resv style: 1 FF, Label in: 3, Label out: -
  Time left:  149, Since: Tue Sep 21 16:29:43 2004
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 2 receiver 39024 protocol 0
  PATH rcvfrom: 10.1.13.2 (so-0/0/2.0) 7 pkts
  Adspec: received MTU 1500
  PATH sentto: localclient
```

```
  RESV rcvfrom: localclient
   Record route: 10.1.36.2 10.1.13.2 <self>
Total 1 displayed, Up 1, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Sample Output 2

command-name

```
[edit protocols mpls]
user@R1# show
label-switched-path R1-to-R6 {
    to 10.0.0.6;
}
interface fxp0.0 {
    disable;
}
inactive: interface so-0/0/0.0;
inactive: interface so-0/0/1.0;
interface so-0/0/2.0;
```

Meaning

Sample Output 1 from ingress router **R1** shows that the LSP is now traversing the network along the expected path, from **R1** through **R3** to **R6**, and the reverse LSP, from **R6** through **R3** to **R1**.

Sample Output 2 from ingress router **R1** shows that the LSP is forced to take the intended path because MPLS is deactivated on **R1** interfaces **so-0/0/0.0** and **so-0/0/1.0**. If these interfaces were not deactivated, even though the configuration is now correct, the LSP would still traverse the network through the alternate path.

Checking the Data Link Layer

IN THIS SECTION

- Verify the LSP | **2385**
- Verify Interfaces | **2387**

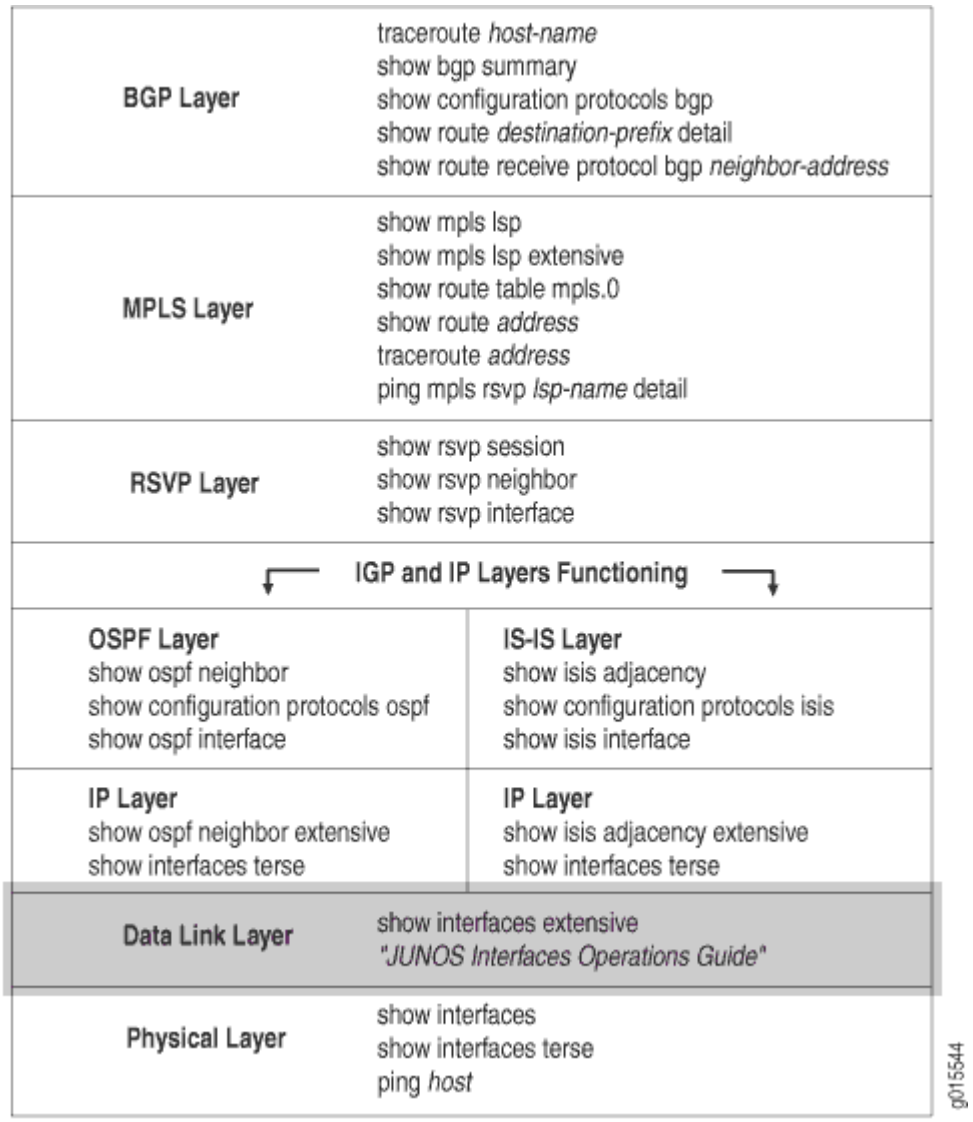
Take Appropriate Action | **2392**
Verify the LSP Again | **2394**

Purpose

After you have configured the label-switched path (LSP), issued the `show mpls lsp extensive` command, and determined that there is an error, you might find that the error is not in the physical layer. Continue investigating the problem at the data link layer of the network.

Figure 162 on page 2384 illustrates the data link layer of the layered MPLS model.

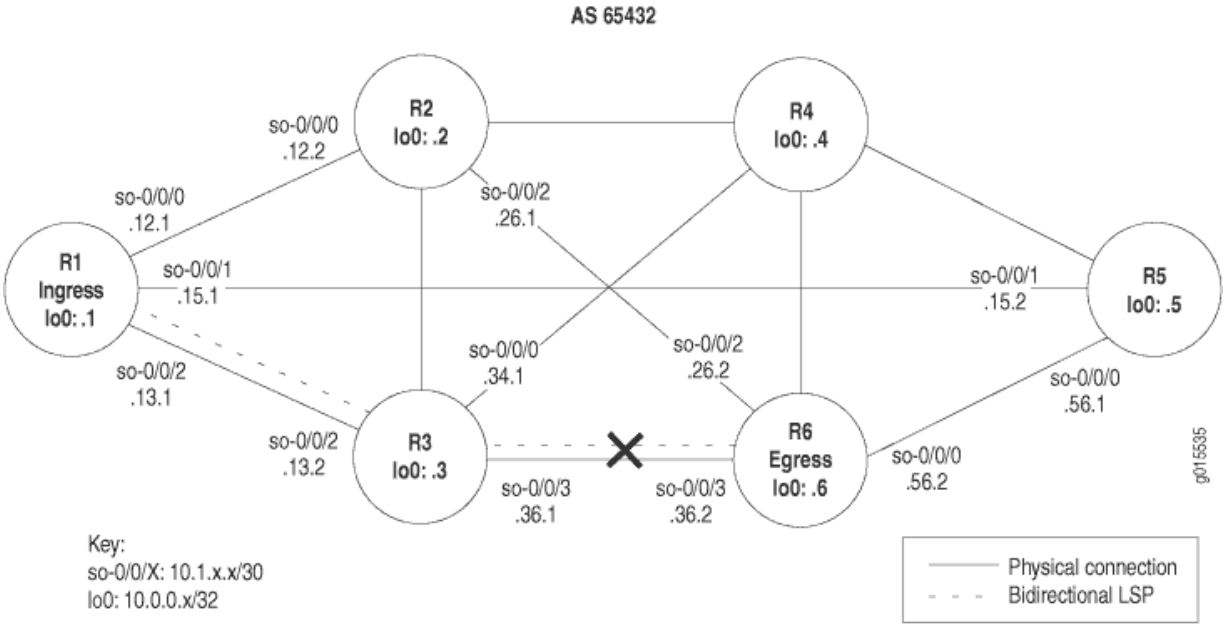
Figure 162: Checking the Data Link Layer



With this layer, you must check the encapsulation mode, for example, Point-to-Point Protocol (PPP) or Cisco High-Level Data Link Control (HDLC); PPP options, for example, header encapsulation; frame check sequence (FCS) size; and whether keepalive frames are enabled or disabled. Also, check the ingress, egress, and transit routers.

Figure 163 on page 2385 illustrates the MPLS network used in this topic.

Figure 163: MPLS Network Broken at the Data Link Layer



The network shown in Figure 163 on page 2385 is a fully meshed configuration where every directly connected interface can receive and send packets to every other similar interface. The LSP in this network is configured to run from ingress router **R1**, through transit router **R3**, to egress router **R6**. In addition, a reverse LSP is configured to run from **R6** through **R3** to **R1**, creating bidirectional traffic.

However, in this example, the LSP is down without a path in either direction, from **R1** to **R6** or from **R6** to **R1**.

When you verify that the data link layer is not functioning correctly, you might find a mismatch with PPP or Cisco HDLC encapsulation, PPP options, or keepalive frames.

The cross shown in Figure 163 on page 2385 indicates where the LSP is broken because of a configuration error on ingress router **R1** that prevents the LSP from traversing the network as expected.

To check the data link layer, follow these steps:

Verify the LSP

IN THIS SECTION

- Purpose | **2386**
- Action | **2386**
- Meaning | **2387**

Purpose

Typically, you use the `show mpls lsp extensive` command to verify the LSP. However for quick verification of the LSP state, use the `show mpls lsp` command. If the LSP is down, use the **extensive** option (`show mpls lsp extensive`) as a follow-up. If your network has numerous LSPs, you might consider specifying the name of the LSP, using the **name** option (`show mpls lsp name` ***name*** or `show mpls lsp name` ***name*** **extensive).**

Action

To determine whether the LSP is up, enter the following command from the ingress router:

```
user@host> show mpls lsp extensive
```

Sample Output 1

command-name

```
user@R1> show mpls lsp extensive
Ingress LSP: 1 sessions

10.0.0.6
  From: 10.0.0.1 , State: Dn, ActiveRoute: 0,  LSPname: R1-to-R6
   ActivePath: (none)
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
  Primary                   State: Dn
    Will be enqueued for recomputation in 15 second(s).
    140 Sep 30 12:01:12 CSPF failed: no route toward 10.0.0.6[26 times]
   139 Sep 30 11:48:57 Deselected as active
   138 Sep 30 11:48:56 CSPF failed: no route toward 10.0.0.6
   137 Sep 30 11:48:56 Clear Call
   136 Sep 30 11:48:56 CSPF: link down/deleted
10.1.36.1(R3.00/10.0.0.3)->10.1.36.2(R6.00/10.0.0.6)
135 Sep 30 11:48:56 ResvTear received
   134 Sep 30 11:48:56 Down
   133 Sep 30 11:48:56 CSPF failed: no route toward 10.0.0.6
   132 Sep 30 11:48:56 10.1.13.2: No Route toward dest
   [...Output truncated...]
  Created: Sat Jul 10 18:18:44 2004
Total 1 displayed, Up 0, Down 1
```

```
Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Meaning

The sample output from ingress router **R1** shows the LSPs within which it participates. The ingress LSP is down, without a path from **R1** to **R6.** Because a reverse LSP is configured in the network shown in "MPLS Network Broken at the Data Link Layer" on page 2312, we would expect an egress LSP session to be up. However, **R1** does not have any egress LSPs, indicating that the LSP from **R6** to **R1** is not functioning.

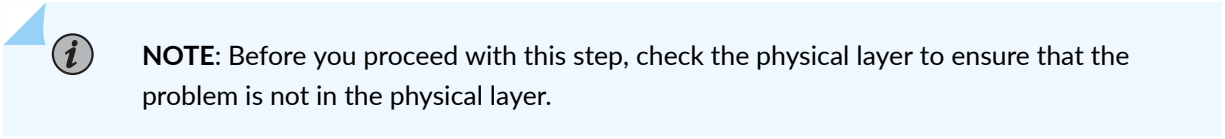
Verify Interfaces

IN THIS SECTION

- Purpose | **2387**
- Action | **2388**
- Meaning | **2392**

Purpose

From your network topology, determine the adjacent interfaces through which the LSP is meant to traverse, and examine the output for the encapsulation type, PPP options, FCS size, and whether keepalive frames are enabled or disabled



NOTE: Before you proceed with this step, check the physical layer to ensure that the problem is not in the physical layer.

Action

To verify the functioning of adjacent interfaces, enter the following commands from the relevant routers:

```
user@host> show interfaces type-fpc/pic/port extensive
user@host> show interfaces type-fpc/pic/port
```

Sample Output 1

command-name

```
user@R6> show interfaces so-0/0/3 extensive
Physical interface: so-0/0/3, Enabled, Physical link is Up
  Interface index: 131, SNMP ifIndex: 27, Generation: 14
   Link-level type: Cisco-HDLC , MTU: 4474, Clocking: Internal, SONET mode, Speed: OC3,
Loopback: None,
   FCS: 16 , Payload scrambler: Enabled
  Device flags   : Present Running
   Interface flags: Link-Layer-Down  Point-To-Point SNMP-Traps 16384
   Link flags     : Keepalives
  Hold-times     : Up 0 ms, Down 0 ms
  Keepalive settings: Interval 10 seconds, Up-count 1, Down-count 3
  Keepalive statistics:
    Input : 0 (last seen: never)
    Output: 357 (last sent 00:00:04 ago)
  CoS queues     : 4 supported
  Last flapped   : 2004-07-21 16:03:49 PDT (10w0d 07:01 ago)
  Statistics last cleared: Never
  Traffic statistics:
   Input  bytes  :            203368873                    0 bps
   Output bytes  :            186714992                   88 bps
   Input  packets:              3641808                    0 pps
   Output packets:              3297569                    0 pps
  Input errors:
    Errors: 0, Drops: 0, Framing errors: 0, Runts: 0, Giants: 0, Bucket drops: 0,
    Policed discards: 1770, L3 incompletes: 0, L2 channel errors: 0, L2 mismatch timeouts: 0,
    HS link CRC errors: 0, HS link FIFO overflows: 0
  Output errors:
    Carrier transitions: 1, Errors: 0, Drops: 0, Aged packets: 0, HS link FIFO underflows: 0,
```

```
  MTU errors: 0
Queue counters:       Queued packets  Transmitted packets      Dropped packets
  0 best-effort               197012               197012                    0
  1 expedited-fo                   0                    0                    0
  2 assured-forw                   0                    0                    0
  3 network-cont             3100557              3100557                    0
 SONET alarms   : None
SONET defects  : None
SONET PHY:            Seconds        Count   State
  PLL Lock                  0            0   OK
  PHY Light                 0            0   OK
SONET section:
  BIP-B1                    0            0
  SEF                       1            3   OK
  LOS                       1            1   OK
  LOF                       1            1   OK
  ES-S                      1
  SES-S                     1
  SEFS-S                    1
SONET line:
  BIP-B2                    0            0
  REI-L                     0            0
  RDI-L                     0            0   OK
  AIS-L                     0            0   OK
  BERR-SF                   0            0   OK
  BERR-SD                   0            0   OK
  ES-L                      1
  SES-L                     1
  UAS-L                     0
  ES-LFE                    0
  SES-LFE                   0
  UAS-LFE                   0
SONET path:
  BIP-B3                    0            0
  REI-P                     0            0
  LOP-P                     0            0   OK
  AIS-P                     0            0   OK
  RDI-P                     0            0   OK
  UNEQ-P                    0            0   OK
  PLM-P                     0            0   OK
  ES-P                      1
  SES-P                     1
  UAS-P                     0
```

```
  ES-PFE                    0
  SES-PFE                   0
  UAS-PFE                   0
Received SONET overhead:
  F1      : 0x00, J0      : 0x00, K1      : 0x00, K2      : 0x00
  S1      : 0x00, C2      : 0xcf, C2(cmp) : 0xcf, F2      : 0x00
  Z3      : 0x00, Z4      : 0x00, S1(cmp) : 0x00
Transmitted SONET overhead:
  F1      : 0x00, J0      : 0x01, K1      : 0x00, K2      : 0x00
  S1      : 0x00, C2      : 0xcf, F2      : 0x00, Z3      : 0x00
  Z4      : 0x00
 Received path trace: R3 so-0/0/3
  52 33 20 73 6f 2d 30 2f 30 2f 33 00 00 00 00 00    R3 so-0/0/3.. ...
  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
  00 00 00 00 00 00 00 00 00 00 00 00 00 0d 0a      ...............
 Transmitted path trace: R6 so-0/0/3
  52 36 20 73 6f 2d 30 2f 30 2f 33 00 00 00 00 00    R6 so-0/0/3 .....
  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
HDLC configuration:
  Policing bucket: Disabled
  Shaping bucket : Disabled
  Giant threshold: 4484, Runt threshold: 3
Packet Forwarding Engine configuration:
  Destination slot: 0, PLP byte: 1 (0x00)
  CoS transmit queue             Bandwidth           Buffer Priority   Limit
                              %          bps   %        bytes
  0 best-effort              95    147744000  95            0      low    none
  3 network-control           5      7776000   5            0      low    none

Logical interface so-0/0/3.0 (Index 71) (SNMP ifIndex 28) (Generation 16)
   Flags: Device-Down  Point-To-Point SNMP-Traps  Encapsulation: Cisco-HDLC
  Traffic statistics:
   Input  bytes  :            406737746
   Output bytes  :            186714992
   Input  packets:              7283616
   Output packets:              3297569
  Local statistics:
   Input  bytes  :            203368873
   Output bytes  :            186714992
   Input  packets:              3641808
```

```
     Output packets:              3297569
    Transit statistics:
     Input  bytes  :            203368873                   0 bps
     Output bytes  :                    0                   0 bps
     Input  packets:              3641808                   0 pps
     Output packets:                    0                   0 pps
    Protocol inet, MTU: 4470, Generation: 46, Route table: 0
      Flags: None
       Addresses, Flags: Dest-route-down Is-Preferred Is-Primary
        Destination: 10.1.36.0/30, Local: 10.1.36.2, Broadcast: 10.1.36.3, Generation: 38
    Protocol iso, MTU: 4469, Generation: 47, Route table: 0
      Flags: None
    Protocol mpls, MTU: 4458, Generation: 48, Route table: 0
      Flags: None
```

Sample Output 2

command-name

```
user@R3> show interfaces so-0/0/3
Physical interface: so-0/0/3, Enabled, Physical link is Up
  Interface index: 131, SNMP ifIndex: 24
   Link-level type: PPP , MTU: 4474, Clocking: Internal, SONET mode, Speed: OC3, Loopback:
None,  FCS: 16 ,
  Payload scrambler: Enabled
  Device flags   : Present Running
  Interface flags: Point-To-Point SNMP-Traps
   Link flags     : Keepalives
  Keepalive settings: Interval 10 seconds, Up-count 1, Down-count 3
  Keepalive: Input: 736827 (00:00:03 ago), Output: 736972 (00:00:05 ago)
  LCP state: Opened
  NCP state: inet: Opened, inet6: Not-configured, iso: Opened, mpls: Opened
  CHAP state: Not-configured
  CoS queues     : 4 supported
  Last flapped   : 2004-07-21 16:08:01 PDT (10w5d 19:57 ago)
  Input rate     : 40 bps (0 pps)
  Output rate    : 48 bps (0 pps)
  SONET alarms   : None
  SONET defects  : None

   Logical interface so-0/0/3.0 (Index 70) (SNMP ifIndex 51)
```

```
    Flags: Point-To-Point SNMP-Traps  Encapsulation: PPP
    Protocol inet, MTU: 4470
      Flags: None
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 10.1.36.0/30, Local: 10.1.36.1, Broadcast: 10.1.36.3
    Protocol iso, MTU: 4470
      Flags: None
    Protocol mpls, MTU: 4458
      Flags: None
```

Meaning

Sample Output 1 from egress router **R6** shows that there are no SONET alarms or defects (**none**), the states are all **OK**, and the path trace shows the distant end (**R3 so-0.0.0**), indicating that the physical link is up. However, the logical link is down, and the link-level type is Cisco HDLC.

Sample Output 2 from transit router **R3** shows that the link-level type is PPP, indicating that the encapsulation types are mismatched, resulting in the LSP going down.

Take Appropriate Action

IN THIS SECTION

- Problem | 2392
- Solution | 2393

Problem

Description

Depending on the error you encountered in your investigation, you must take the appropriate action to correct the problem. In the example below, the encapsulation types are mismatched.

Solution

To correct the error in this example, enter the following commands:

```
[edit interfaces so-0/0/3]
user@R1# show
user@R1# delete encapsulation
user@R1# show
user@R1# commit
```

Sampel Output

```
[edit interfaces so-0/0/3]
user@R6# show
encapsulation cisco-hdlc;
unit 0 {
    family inet {
        address 10.1.36.2/30;
    }
    family iso;
    family mpls;
}

[edit interfaces so-0/0/3]
user@R6# delete encapsulation

[edit interfaces so-0/0/3]
user@R6# show
unit 0 {
    family inet {
        address 10.1.36.2/30;
    }
    family iso;
    family mpls;
}

[edit interfaces so-0/0/3]
user@R6# commit
commit complete
```

Meaning

The sample output from egress router **R6** shows that the Cisco HDLC was incorrectly configured on interface **so-0/0/3** which prevented the LSP from using the intended path. The problem was corrected when the `encapsulation` statement was deleted and the configuration committed.

Verify the LSP Again

IN THIS SECTION

- Purpose | 2394
- Action | 2394
- Meaning | 2399

Purpose

After taking the appropriate action to correct the error, the LSP needs to be checked again to confirm that the problem in the data link layer has been resolved.

Action

From the ingress, egress, and transit routers, verify that the LSP is up and traversing the network as expected:

```
user@host>  show mpls lsp extensive
```

Sample Output 1

command-name

```
user@R1> show mpls lsp extensive
Ingress LSP: 1 sessions

10.0.0.6
  From: 10.0.0.1 , State: Up,  ActiveRoute: 1 ,  LSPname: R1-to-R6
  ActivePath:  (primary)
```

```
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                    State: Up
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 20)
  10.1.13.2 S 10.1.36.2 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt):
          10.1.13.2 10.1.36.2
   145 Sep 30 12:25:01 Selected as active path
   144 Sep 30 12:25:01 Record Route:  10.1.13.2 10.1.36.2
   143 Sep 30 12:25:01 Up
   142 Sep 30 12:25:01 Originate Call
   141 Sep 30 12:25:01 CSPF: computation result accepted
   140 Sep 30 12:24:32 CSPF failed: no route toward 10.0.0.6[74 times]
   139 Sep 30 11:48:57 Deselected as active
   138 Sep 30 11:48:56 CSPF failed: no route toward 10.0.0.6
   137 Sep 30 11:48:56 Clear Call
   136 Sep 30 11:48:56 CSPF: link down/deleted 10.1.36.1(R3.00/10.0.0.3)-
>10.1.36.2(R6.00/10.0.0.6)
   [...Output truncated...]
  Created: Sat Jul 10 18:18:43 2004
Total 1 displayed, Up 1 , Down 0

Egress LSP: 1 sessions

10.0.0.1
  From: 10.0.0.6 , LSPstate: Up, ActiveRoute: 0
  LSPname: R6-to-R1, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: -
  Resv style: 1 FF, Label in: 3, Label out: -
  Time left:  134, Since: Thu Sep 30 12:24:56 2004
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 6 receiver 39024 protocol 0
  PATH rcvfrom: 10.1.13.2 (so-0/0/2.0) 7 pkts
  Adspec: received MTU 1500
  PATH sentto: localclient
  RESV rcvfrom: localclient
   Record route: 10.1.36.2 10.1.13.2 <self>
Total 1 displayed, Up 1, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Sample Output 2

command-name

```
user@R6> show mpls lsp extensive
Ingress LSP: 1 sessions

10.0.0.1
  From: 10.0.0.6, State: Up, ActiveRoute: 1, LSPname: R6-to-R1
  ActivePath:  (primary)
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                   State: Up
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 20)
 10.1.36.1 S 10.1.13.1 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt):
          10.1.36.1 10.1.13.1
   50 Sep 30 12:24:12 Selected as active path
   49 Sep 30 12:24:12 Record Route:  10.1.36.1 10.1.13.1
   48 Sep 30 12:24:12 Up
   47 Sep 30 12:24:12 Originate Call
   46 Sep 30 12:24:12 CSPF: computation result accepted
   45 Sep 30 12:23:43 CSPF failed: no route toward 10.0.0.1[73 times]
   44 Sep 30 11:48:12 Deselected as active
   43 Sep 30 11:48:12 CSPF failed: no route toward 10.0.0.1
   42 Sep 30 11:48:12 CSPF: link down/deleted 10.1.36.2(R6.00/10.0.0.6)-
>10.1.36.1(R3.00/10.0.0.3)
   [...Output truncated...]
  Created: Tue Aug 17 12:18:34 2004
Total 1 displayed, Up 1, Down 0

Egress LSP: 1 sessions

10.0.0.6
  From: 10.0.0.1 , LSPstate: Up, ActiveRoute: 0
  LSPname: R1-to-R6, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: -
  Resv style: 1 FF, Label in: 3, Label out: -
  Time left:  159, Since: Thu Sep 30 12:24:16 2004
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 19 receiver 44251 protocol 0
```

```
  PATH rcvfrom: 10.1.36.1 (so-0/0/3.0) 4 pkts
  Adspec: received MTU 1500
  PATH sentto: localclient
  RESV rcvfrom: localclient
   Record route: 10.1.13.1 10.1.36.1 <self>
Total 1 displayed, Up 1, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Sample Output 3

command-name

```
user@R3> show mpls lsp extensive
Ingress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 2 sessions

10.0.0.1
  From: 10.0.0.6 , LSPstate: Up, ActiveRoute: 1
  LSPname: R6-to-R1, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 3
  Resv style: 1 FF, Label in: 100176, Label out: 3
  Time left:  143, Since: Thu Sep 30 12:21:25 2004
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 6 receiver 39024 protocol 0
  PATH rcvfrom: 10.1.36.2 (so-0/0/3.0) 10 pkts
  Adspec: received MTU 1500 sent MTU 1500
  PATH sentto: 10.1.13.1 (so-0/0/2.0) 9 pkts
  RESV rcvfrom: 10.1.13.1 (so-0/0/2.0) 9 pkts
  Explct route: 10.1.13.1
   Record route: 10.1.36.2 <self> 10.1.13.1

10.0.0.6
  From: 10.0.0.1 , LSPstate: Up, ActiveRoute: 1
```

```
  LSPname: R1-to-R6, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 3
  Resv style: 1 FF, Label in: 100192, Label out: 3
  Time left:  148, Since: Thu Sep 30 12:21:30 2004
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 19 receiver 44251 protocol 0
  PATH rcvfrom: 10.1.13.1 (so-0/0/2.0) 9 pkts
  Adspec: received MTU 1500 sent MTU 1500
  PATH sentto: 10.1.36.2 (so-0/0/3.0) 9 pkts
  RESV rcvfrom: 10.1.36.2 (so-0/0/3.0) 9 pkts
  Explct route: 10.1.36.2
   Record route: 10.1.13.1 <self> 10.1.36.2
Total 2 displayed, Up 2, Down 0
```

Sample Output 4

command-name

```
user@R1> show configuration protocols mpls
label-switched-path R1-to-R6 {
    to 10.0.0.6;
}
inactive: interface so-0/0/0.0;
inactive: interface so-0/0/1.0;
interface so-0/0/2.0;

user@R6> show configuration protocols mpls
label-switched-path R6-to-R1 {
    to 10.0.0.1;
}
inactive: interface so-0/0/0.0;
inactive: interface so-0/0/1.0;
inactive: interface so-0/0/2.0;
interface so-0/0/3.0;

user@R3> show configuration protocols mpls
interface fxp0.0 {
    disable;
}
inactive: interface so-0/0/0.0;
```

```
inactive: interface so-0/0/1.0;
interface so-0/0/2.0;
interface so-0/0/3.0;
```

Meaning

Sample Outputs 1 and 2 from ingress router **R1** and egress router **R6,** respectively, show that the LSP is now traversing the network along the expected path, from **R1** through **R3** to **R6**, and the reverse LSP, from **R6** through **R3** to **R1**.

Sample Output 3 from transit router **R3** shows that there are two transit LSP sessions, one from **R1** to **R6** and the other from **R6** to **R1**.

Sample Output 4 shows the interfaces that were deactivated on the ingress, egress, and transit routers, forcing the LSP to take the intended path. If these interfaces were not deactivated, even though the configuration is now correct, the LSP would still traverse the network through the alternate path.

Verifying the IP and IGP Layers

IN THIS SECTION

- Problem | **2399**
- Solution | **2400**

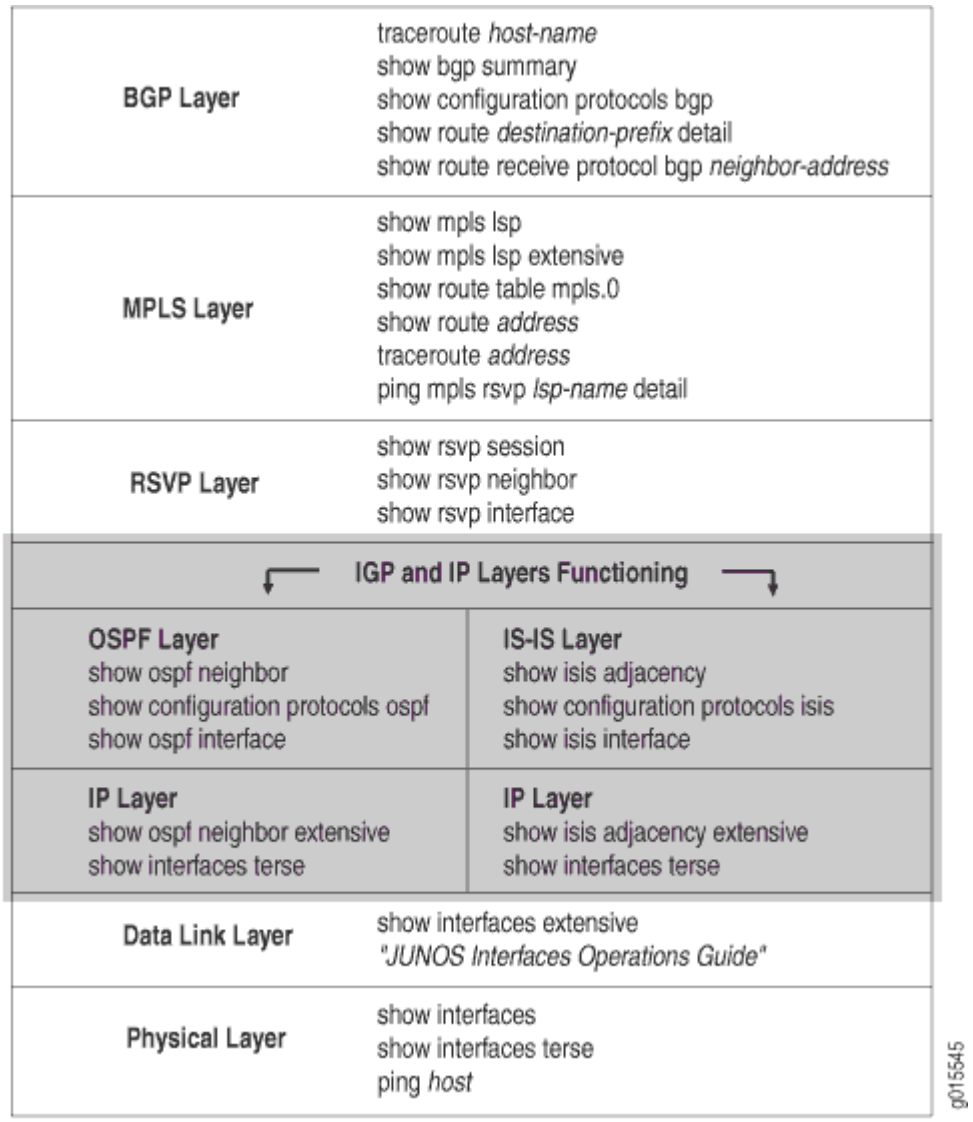
Problem

Description

After you have configured the label-switched path (LSP), issued the `show mpls lsp extensive` command, and determined that there is an error, you might find that the error is not in the physical or data link layers. Continue investigating the problem at the IP and IGP layers of the network.

Figure 164 on page 2400 illustrates the IP and IGP layers of the layered MPLS model.

Figure 164: IP and IGP Layers



Solution

At the IP and IGP layers, you must check the following:

- Interfaces have correct IP addressing, and the IGP neighbors or adjacencies are established.
- Open Shortest Path First (OSPF) or Intermediate System-to-Intermediate System (IS-IS) protocols are configured and running correctly.
    - If the OSPF protocol is configured, check the IP layer first, then the OSPF configuration, making sure that the protocol, interfaces, and traffic engineering are configured correctly.

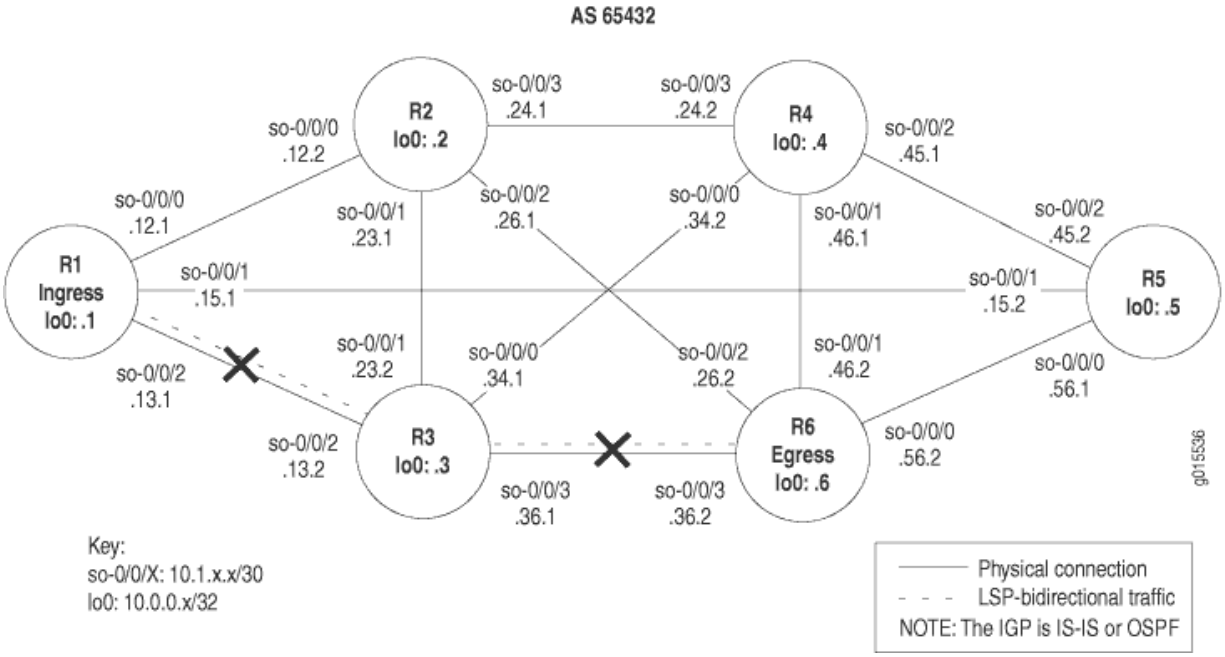
- If the IS-IS protocol is configured, it doesn't matter whether you check IS-IS or IP first because both protocols are independent of each other. Verify that IS-IS adjacencies are up, and that the interfaces and IS-IS protocol are configured correctly.

> **NOTE**: The IS-IS protocol has traffic engineering enabled by default.

If the network is not functioning at the IP or IGP layers, the LSP does not work as configured.

Figure 165 on page 2401 illustrates the MPLS network used in this topic.

Figure 165: MPLS Network Broken at the IP and IGP Layers



The network shown in Figure 165 on page 2401 is a fully meshed configuration where every directly connected interface can receive and send packets to every other similar interface. The LSP in this network is configured to run from ingress router **R1**, through transit router **R3**, to egress router **R6**. In addition, a reverse LSP is configured to run from **R6**, through **R3**, to **R1**, creating bidirectional traffic. The crosses in Figure 165 on page 2401 indicate where the LSP is not working because of the following problems at the IP and IGP layer:

- An IP address is configured incorrectly on the ingress router (**R1**).
- The OSPF protocol is configured with a router ID (RID) but without the loopback (**lo0**) interface, and traffic engineering is missing from the transit router (**R3**).
- Levels in the IS-IS network are mismatched.

Verifying the IP Layer

IN THIS SECTION

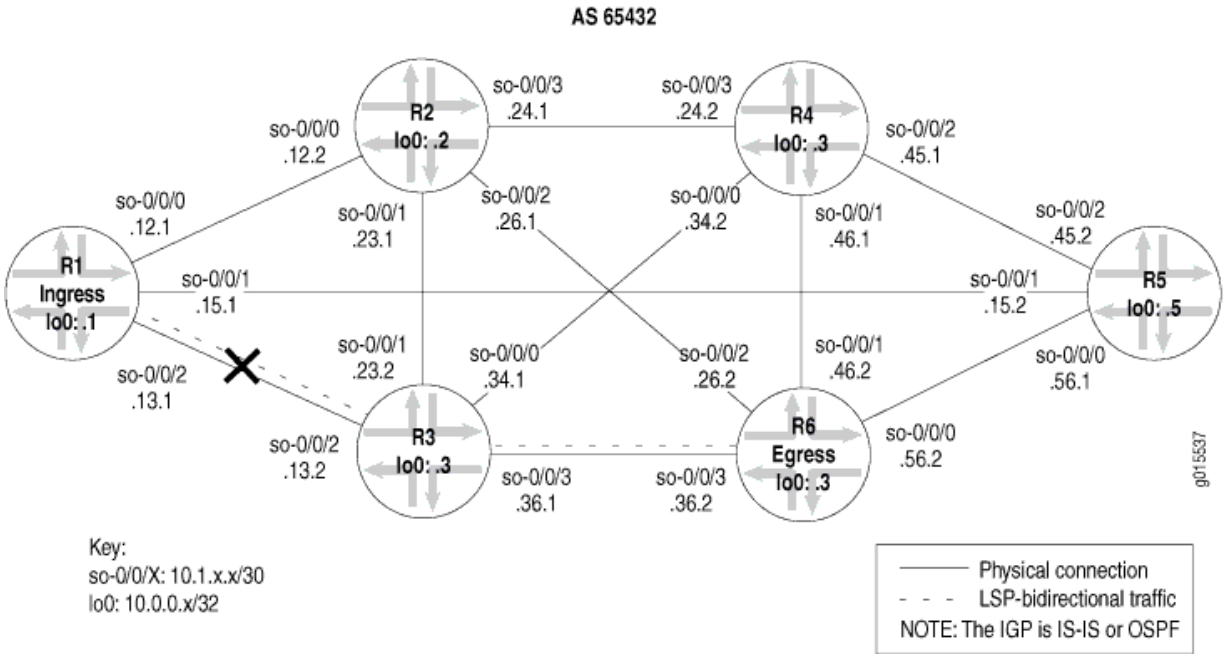
- Verify the LSP | **2403**
- Verify IP Addressing | **2404**
- Verify Neighbors or Adjacencies at the IP Layer | **2407**
- Take Appropriate Action | **2411**
- Verify the LSP Again | **2413**

Purpose

You can check the IP layer before or after you check the interior gateway protocol (IGP) layer, depending on whether you have OSPF or IS-IS configured as the IGP. If your MPLS network is configured with OSPF as the IGP, you must first verify the IP layer, checking that the interfaces have correct IP addressing and that the OSPF neighbors are established before you check the OSPF layer.

If you have IS-IS configured as the IGP in your MPLS network, you can verify either the IP layer or the IS-IS protocol layer first. The order in which you check the IP or IS-IS layer does not affect the results.

Figure 166: MPLS Network Broken at the IP Layer



The cross in Figure 166 on page 2402 indicates where the LSP is broken because of the incorrect configuration of an IP address on ingress router **R1**.

Verify the LSP

IN THIS SECTION

- Purpose | **2403**
- Action | **2403**
- Meaning | **2404**

Purpose

After configuring the LSP, you must verify that the LSP is up. LSPs can be ingress, transit, or egress. Use the `show mpls lsp` command for quick verification of the LSP state, with the **extensive** option (`show mpls lsp extensive`) as a follow-up if the LSP is down. If your network has numerous LSPs, you might consider specifying the name of the LSP, using the **name** option (`show mpls lsp name` ***name*** or `show mpls lsp name` ***name*** **extensive)**.

Action

To verify that the LSP is up, enter the following command from the ingress router:

```
user@host> show mpls lsp extensive
```

Sample Output 1

command-name

```
user@R1> show mpls lsp extensive
Ingress LSP: 1 sessions

10.0.0.6
  From: 10.0.0.1, State: Dn, ActiveRoute: 0, LSPname: R1-to-R6
   ActivePath: (none)
  LoadBalance: Random
```

```
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
  Primary                   State: Dn
    Will be enqueued for recomputation in 25 second(s).
   44 Oct 15 16:56:11 CSPF failed:  no route toward 10.0.0.6 [2685 times]
   43 Oct 14 19:07:09 Clear Call
   42 Oct 14 19:06:56 Deselected as active
   41 Oct 14 19:06:56 10.1.12.1: MPLS label allocation failure
   40 Oct 14 19:06:56 Down
   39 Oct 14 18:43:43 Selected as active path
   38 Oct 14 18:43:43 Record Route:  10.1.13.2 10.1.36.2
   37 Oct 14 18:43:43 Up
   [...Output truncated...]
  Created: Thu Oct 14 16:04:33 2004
Total 1 displayed, Up 0,  Down 1

Egress LSP: 0 sessions
Total 0 displayed , Up 0, Down 0

Transit LSP: 0 sessions
Total 0 displayed , Up 0, Down 0
```

Meaning

The sample output from ingress router **R1** shows that an MPLS label allocation failure occurred and the Constrained Shortest Path First (CSPF) algorithm failed, resulting in no route to destination **10.0.0.6** on **R6**.

Verify IP Addressing

IN THIS SECTION

- Purpose | **2405**
- Action | **2405**
- Meaning | **2406**

Purpose

When you investigate the IP layer, you verify that interfaces have correct IP addressing, and that OSPF neighbors or IS-IS adjacencies are established. In this example, an IP address is configured incorrectly on the ingress router (**R1**).

Action

To verify IP addressing, enter the following command from the ingress, transit, and egress routers:

```
user@host> show interfaces terse
```

Sample Output

command-name

```
user@R1> show interfaces terse
Interface               Admin Link Proto Local                 Remote
so-0/0/0                up    up
so-0/0/0.0              up    up   inet  10.1.12.1/30
                                  iso
                                  mpls
so-0/0/1                up    up
so-0/0/1.0              up    up   inet  10.1.15.1/30
                                  iso
                                  mpls
so-0/0/2                up    up
so-0/0/2.0              up    up   inet  10.1.13.2 <<< Incorrect IP address
                                  iso
                                  mpls
lo0                     up    up
lo0.0                   up    up   inet  10.0.0.1
                                  iso   49.0004.1000.0000.0001.00

user@R3> show interfaces terse
Interface               Admin Link Proto Local                 Remote
so-0/0/0                up    up
so-0/0/0.0              up    up   inet  10.1.34.1/30
                                  iso
                                  mpls
```

```
so-0/0/1                up    up
so-0/0/1.0              up    up   inet  10.1.23.2/30
                                   iso
                                   mpls
so-0/0/2                up    up
so-0/0/2.0              up    up   inet  10.1.13.2/30 <<< Identical to R1
                                   iso
                                   mpls
so-0/0/3                up    up
so-0/0/3.0              up    up   inet  10.1.36.1/30
                                   iso
                                   mpls
lo0                     up    up
lo0.0                   up    up   inet  10.0.0.3
                                   iso   49.0004.1000.0000.0003.00

user@R6> show interfaces terse
Interface               Admin Link Proto Local                 Remote
so-0/0/0                up    up
so-0/0/0.0              up    up   inet  10.1.56.2/30
                                   iso
                                   mpls
so-0/0/1                up    up
so-0/0/1.0              up    up   inet  10.1.46.2/30
                                   iso
                                   mpls
so-0/0/2                up    up
so-0/0/2.0              up    up   inet  10.1.26.2/30
                                   iso
                                   mpls
so-0/0/3                up    up
so-0/0/3.0              up    up   inet  10.1.36.2/30
                                   iso
                                   mpls
lo0.0                   up    up   inet  10.0.0.6
                                   iso   49.0004.1000.0000.0006.00
```

Meaning

The sample output shows that the IP addresses for interface **so-0/0/2.0** on **R1** and interface **so-0/0/2.0** on **R3** are identical. Interface IP addresses within a network must be unique for the interface to be identified correctly.

Verify Neighbors or Adjacencies at the IP Layer

IN THIS SECTION

- Purpose | 2407
- Action | 2407
- Meaning | 2411

Purpose

If the IP addressing is configured incorrectly then the OSPF neighbors or IS-IS adjacencies both need to be checked to determine if one or both of them are established.

Action

To verify neighbors (OSPF) or adjacencies (IS-IS), enter the following commands from the ingress, transit, and egress routers:

```
user@host> show ospf neighbor extensive
user@host> show isis adjacency extensive
```

Sample Output 1

command-name

```
user@R1> show ospf neighbor extensive
Address         Interface             State      ID              Pri  Dead
10.1.12.2        so-0/0/0.0             Full      10.0.0.2          128   34
  area 0.0.0.0, opt 0x42, DR 0.0.0.0, BDR 0.0.0.0
  Up 1d 04:45:20, adjacent 1d 04:45:20
10.1.15.2        so-0/0/1.0             Full      10.0.0.5          128   35
  area 0.0.0.0, opt 0x42, DR 0.0.0.0, BDR 0.0.0.0
  Up 1d 04:45:20, adjacent 1d 04:45:10  <<< no adjacency with R3 so-0/0/2

user@R3> show ospf neighbor extensive
Address         Interface             State      ID              Pri  Dead
```

```
10.1.23.1        so-0/0/1.0             Full      10.0.0.2         128   35
  area 0.0.0.0, opt 0x42, DR 0.0.0.0, BDR 0.0.0.0
  Up 1w2d 04:54:30, adjacent 1w2d 04:54:21
10.1.36.2        so-0/0/3.0             Full      10.0.0.6         128   39
  area 0.0.0.0, opt 0x42, DR 0.0.0.0, BDR 0.0.0.0
  Up 1w2d 04:54:30, adjacent 1w2d 04:54:30  <<< no adjacency with R1 so-0/0/2

user@R6> show ospf neighbor extensive
Address          Interface              State      ID              Pri  Dead
10.1.56.1        so-0/0/0.0             Full      10.0.0.5         128   39
  area 0.0.0.0, opt 0x42, DR 0.0.0.0, BDR 0.0.0.0
  Up 1d 02:59:35, adjacent 1d 02:59:35
10.1.26.1        so-0/0/2.0             Full      10.0.0.2         128   36
  area 0.0.0.0, opt 0x42, DR 0.0.0.0, BDR 0.0.0.0
  Up 1w2d 04:57:30, adjacent 1w2d 04:57:30
10.1.36.1        so-0/0/3.0             Full      10.0.0.3         128   36
  area 0.0.0.0, opt 0x42, DR 0.0.0.0, BDR 0.0.0.0
  Up 1w2d 04:56:11, adjacent 1w2d 04:56:11
```

Sample Output 2

command-name

```
user@R1> show isis adjacency extensive
R2
  Interface: so-0/0/0.0,  Level: 2, State: Up , Expires in 23 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 05:57:16 ago
  Circuit type: 2,  Speaks: IP , IPv6
  Topologies: Unicast
  Restart capable: Yes
   IP addresses: 10.1.12.2
  Transition log:
  When                  State        Reason
  Fri Oct 15 14:58:35   Up           Seenself

R5
  Interface: so-0/0/1.0,  Level: 2, State: Up, Expires in 26 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 05:56:52 ago
  Circuit type: 2,  Speaks: IP , IPv6
  Topologies: Unicast
  Restart capable: Yes
```

```
   IP addresses: 10.1.15.2
  Transition log:
  When                  State        Reason
  Fri Oct 15 14:59:00   Up           Seenself

R3
  Interface: so-0/0/2.0, Level: 2, State: Up, Expires in 26 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 05:56:51 ago
  Circuit type: 2,  Speaks: IP , IPv6
  Topologies: Unicast
  Restart capable: Yes
  IP addresses: 10.1.13.2
  Transition log:
  When                  State        Reason
  Fri Oct 15 14:59:01   Up           Seenself

user@R3> show isis adjacency extensive
R4
  Interface: so-0/0/0.0,  Level: 2, State: Up , Expires in 25 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 1w1d 00:22:51 ago
  Circuit type: 2,  Speaks: IP , IPv6
  Topologies: Unicast
  Restart capable: Yes
   IP addresses: 10.1.34.2
  Transition log:
  When                  State        Reason
  Thu Oct 28 15:13:12   Up           Seenself

R2
  Interface: so-0/0/1.0,  Level: 2, State: Up , Expires in 25 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 2w2d 18:02:48 ago
  Circuit type: 2,  Speaks: IP , IPv6
  Topologies: Unicast
  Restart capable: Yes
   IP addresses: 10.1.23.1
  Transition log:
  When                  State        Reason
  Tue Oct 19 21:33:15   Up           Seenself

R1
  Interface: so-0/0/2.0,  Level: 2, State: Up , Expires in 22 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 2w2d 17:24:06 ago
  Circuit type: 2,  Speaks: IP , IPv6
```

```
  Topologies: Unicast
  Restart capable: Yes
   IP addresses: 10.1.13.1
  Transition log:
  When                  State        Reason
  Tue Oct 19 22:11:57   Up           Seenself

R6
  Interface: so-0/0/3.0,  Level: 2, State: Up , Expires in 21 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 2w1d 00:07:00 ago
  Circuit type: 2,  Speaks: IP , IPv6
  Topologies: Unicast
  Restart capable: Yes
   IP addresses: 10.1.36.2
  Transition log:
  When                  State        Reason
  Thu Oct 21 15:29:03   Up           Seenself

user@R6>  show isis adjacency extensive
R5
  Interface: so-0/0/0.0,  Level: 2, State: Up , Expires in 23 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 1w2d 01:10:03 ago
  Circuit type: 2,  Speaks: IP , IPv6
  Topologies: Unicast
  Restart capable: Yes
   IP addresses: 10.1.56.1
  Transition log:
  When                  State        Reason
  Wed Oct 27 14:35:32   Up           Seenself

R4
  Interface: so-0/0/1.0,  Level: 2, State: Up , Expires in 25 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 1w1d 00:26:50 ago
  Circuit type: 2,  Speaks: IP , IPv6
  Topologies: Unicast
  Restart capable: Yes
   IP addresses: 10.1.46.1
  Transition log:
  When                  State        Reason
  Thu Oct 28 15:18:45   Up           Seenself

R2
  Interface: so-0/0/2.0,  Level: 2, State: Up , Expires in 24 secs
```

```
  Priority: 0, Up/Down transitions: 1, Last transition: 2w1d 00:11:40 ago
  Circuit type: 2,  Speaks: IP , IPv6
  Topologies: Unicast
  Restart capable: Yes
   IP addresses: 10.1.26.1
  Transition log:
  When                  State        Reason
  Thu Oct 21 15:33:55   Up           Seenself

R3
  Interface: so-0/0/3.0,  Level: 2, State: Up , Expires in 19 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 2w1d 00:11:40 ago
  Circuit type: 2,  Speaks: IP , IPv6
  Topologies: Unicast
  Restart capable: Yes
   IP addresses: 10.1.36.1
  Transition log:
  When                  State        Reason
  Thu Oct 21 15:33:55   Up           Seenself
```

Meaning

Sample Output 1 from the ingress, transit, and egress routers shows that **R1** and **R3** are not established OSPF neighbors. Considering that the two interfaces **so-0/0/2.0** (**R1** and **R3**) are configured with identical IP addresses, you would expect this. The OSPF protocol routes IP packets based solely on the destination IP address contained in the IP packet header. Therefore, identical IP addresses in the autonomous system (AS) result in neighbors not establishing.

Sample Output 2 from the ingress, transit, and egress routers shows that **R1** and **R3** have established an IS-IS adjacency despite the identical IP addresses configured on interfaces **so-0/0/2.0** on **R1** and **R3**. The IS-IS protocol behaves differently from the OSPF protocol because it does not rely on IP to establish an adjacency. However, if the LSP is not up, it is still useful to check the IP subnet addressing in case there is a mistake in that layer. Correcting the addressing error might bring the LSP back up.

Take Appropriate Action

IN THIS SECTION

- Problem | 2412
- Solution | 2412

Problem

Description

Depending on the error you encountered in your investigation, you must take the appropriate action to correct the problem. In this example, the IP address of an interface on transit router **R2** is incorrectly configured.

Solution

To correct the error in this example, enter the following commands:

```
[edit interfaces                          so-0/0/2]
user@R1# show
user@R1#             rename unit 0 family inet address 10.1.13.2/30 to address
10.1.13.1/30
user@R1# show
user@R1# commit
```

Sample Output

```
[edit interfaces so-0/0/2]
user@R1# show
unit 0 {
    family inet {
        address 10.1.13.2/30;  <<< Incorrect IP address
    }
    family iso;
    family mpls;
}

[edit interfaces so-0/0/2]
user@R1# rename unit 0 family inet address 10.1.13.2/30 to address 10.1.13.1/30

[edit interfaces so-0/0/2]
user@R1# show
unit 0 {
    family inet {
        address 10.1.13.1/30;  <<< Correct IP address.
    }
```

```
    family iso;
    family mpls;
}

[edit interfaces so-0/0/2]
user@R1# commit
commit complete
```

Meaning

The sample output shows that interface **so-0/0/2** on ingress router **R1** is now configured with the correct IP address. This correction results in unique subnet IP addresses for all interfaces in the MPLS network in "MPLS Network Broken at the IP and IGP Layers" on page 2312, and the possibility that the LSP might come up.

Verify the LSP Again

IN THIS SECTION

- Purpose | **2413**
- Action | **2413**
- Meaning | **2417**

Purpose

After taking the appropriate action to correct the error, the LSP needs to be checked again to confirm that the problem in the OSPF protocol has been resolved.

Action

To verify the LSP again, enter the following command on the ingress, transit, and egress routers:

```
user@host> show mpls lsp extensive
```

Sample Output 1

command-name

```
user@R1> show mpls lsp extensive
Ingress LSP: 1 sessions

10.0.0.6
  From: 10.0.0.1, State: Up,  ActiveRoute: 1 , LSPname: R1-to-R6
  ActivePath:  (primary)
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                    State: Up
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 20)
 10.1.13.2 S 10.1.36.2 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt):
          10.1.13.2 10.1.36.2
   54 Oct 15 21:28:16 Selected as active path
   53 Oct 15 21:28:16 Record Route:  10.1.13.2 10.1.36.2
   52 Oct 15 21:28:16 Up
   51 Oct 15 21:28:16 10.1.15.1: MPLS label allocation failure[2 times]
   50 Oct 15 21:28:11 CSPF: computation result accepted
   49 Oct 15 21:27:42 10.1.15.1: MPLS label allocation failure
   48 Oct 15 21:27:42 CSPF: computation result accepted
   47 Oct 15 21:27:31 10.1.15.1: MPLS label allocation failure[4 times]
   46 Oct 15 21:27:13 Originate Call
   45 Oct 15 21:27:13 CSPF: computation result accepted
   [...Output truncated...]
  Created: Thu Oct 14 16:04:34 2004
Total 1 displayed,  Up 1 , Down 0

Egress LSP: 1 sessions

10.0.0.1
  From: 10.0.0.6,  LSPstate: Up , ActiveRoute: 0
   LSPname: R6-to-R1 , LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
   Recovery label received: -, Recovery label sent: -
  Resv style: 1 FF, Label in: 3, Label out: -
  Time left:  149, Since: Fri Oct 15 21:28:13 2004
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 13 receiver 39024 protocol 0
```

```
  PATH rcvfrom: 10.1.13.2 (so-0/0/2.0) 10 pkts
  Adspec: received MTU 1500
  PATH sentto: localclient
  RESV rcvfrom: localclient
   Record route: 10.1.36.2 10.1.13.2 <self>
Total 1 displayed,  Up 1 , Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Sample Output 2

command-name

```
user@R3> show mpls lsp extensive
Ingress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 2 sessions

10.0.0.1
  From: 10.0.0.6,  LSPstate: Up , ActiveRoute: 1
   LSPname: R6-to-R1 , LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 3
  Resv style: 1 FF, Label in: 100336, Label out: 3
  Time left:  156, Since: Fri Oct 15 21:15:47 2004
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 13 receiver 39024 protocol 0
  PATH rcvfrom: 10.1.36.2 (so-0/0/3.0) 11 pkts
  Adspec: received MTU 1500 sent MTU 1500
  PATH sentto: 10.1.13.1 (so-0/0/2.0) 11 pkts
  RESV rcvfrom: 10.1.13.1 (so-0/0/2.0) 11 pkts
  Explct route: 10.1.13.1
   Record route: 10.1.36.2 <self> 10.1.13.1

10.0.0.6
  From: 10.0.0.1,  LSPstate: Up , ActiveRoute: 1
```

```
   LSPname: R1-to-R6 , LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 3
  Resv style: 1 FF, Label in: 100352, Label out: 3
  Time left:  159, Since: Fri Oct 15 21:15:50 2004
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 5 receiver 47901 protocol 0
  PATH rcvfrom: 10.1.13.1 (so-0/0/2.0) 11 pkts
  Adspec: received MTU 1500 sent MTU 1500
  PATH sentto: 10.1.36.2 (so-0/0/3.0) 11 pkts
  RESV rcvfrom: 10.1.36.2 (so-0/0/3.0) 11 pkts
  Explct route: 10.1.36.2
   Record route: 10.1.13.1 <self> 10.1.36.2
Total 2 displayed,  Up 2 , Down 0
```

Sample Output 3

command-name

```
user@R6> show mpls lsp extensive
Ingress LSP: 1 sessions

10.0.0.1
  From: 10.0.0.6,  State: Up , ActiveRoute: 1,  LSPname: R6-to-R1
  ActivePath:  (primary)
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                   State: Up
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 20)
 10.1.36.1 S 10.1.13.1 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt):
          10.1.36.1 10.1.13.1
   187 Oct 15 21:20:05 Selected as active path
   186 Oct 15 21:20:05 Record Route:  10.1.36.1 10.1.13.1
   185 Oct 15 21:20:05 Up
   184 Oct 15 21:20:05 Clear Call
   183 Oct 15 21:20:05 CSPF: computation result accepted
   182 Oct 15 21:20:05 CSPF: link down/deleted 10.1.13.2(R3.00/10.0.0.3)-
>10.1.13.2(R1.00/10.0.0.1)
   [...Output truncated...]
  Created: Tue Aug 17 12:18:33 2004
```

```
Total 1 displayed,  Up 1 , Down 0

Egress LSP: 1 sessions

10.0.0.6
  From: 10.0.0.1, LSPstate: Up, ActiveRoute: 0
   LSPname: R1-to-R6 , LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: -
  Resv style: 1 FF, Label in: 3, Label out: -
  Time left:  144, Since: Fri Oct 15 21:20:08 2004
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 5 receiver 47901 protocol 0
  PATH rcvfrom: 10.1.36.1 (so-0/0/3.0) 11 pkts
  Adspec: received MTU 1500
  PATH sentto: localclient
  RESV rcvfrom: localclient
   Record route: 10.1.13.1 10.1.36.1 <self>
Total 1 displayed,  Up 1 , Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Meaning

Sample Output 1 from ingress router **R1** shows that LSP **R1-to-R6** has an active route to **R6** and the state is up. The output shows that the egress LSP session **R6-to-R1** received and sent a recovery label.

Sample Output 2 from transit router **R3** shows that there are two transit LSP sessions, one from **R1** to **R6** and the other from **R6** to **R1.** Both LSPs are up.

Sample Output 3 from egress router **R6** shows that the LSP is up and the active route is the primary route. The LSP is now traversing the network along the expected path, from **R1** through **R3** to **R6**, and the reverse LSP, from **R6** through **R3** to **R1**.

Verify the LSP Again

IN THIS SECTION

- Purpose | **2418**
- Action | **2418**

- Meaning | **2421**

Purpose

After taking the appropriate action to correct the error, the LSP needs to be checked again to confirm that the problem in the IS-IS protocol has been resolved.

Action

To verify that the LSP is up and traversing the network as expected, enter the following command from the ingress, egress, and transit routers:

```
user@host>              show mpls lsp extensive
```

Sample Output

command-name

```
user@R1> show mpls lsp extensive
Ingress LSP: 1 sessions

10.0.0.6
  From: 10.0.0.1,  State: Up , ActiveRoute: 1,  LSPname: R1-to-R6
   ActivePath:  (primary)
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                    State: Up
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt):
          10.1.13.2 10.1.36.2
    4 Oct 19 21:22:54 Selected as active path
    3 Oct 19 21:22:53 Record Route:  10.1.13.2 10.1.36.2
    2 Oct 19 21:22:53 Up
    1 Oct 19 21:22:53 Originate Call
  Created: Tue Oct 19 21:22:53 2004
Total 1 displayed,  Up 1 , Down 0

Egress LSP: 1 sessions
```

```
10.0.0.1
  From: 10.0.0.6,  LSPstate: Up , ActiveRoute: 0
   LSPname: R6-to-R1 , LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: -
  Resv style: 1 FF, Label in: 3, Label out: -
  Time left:  117, Since: Tue Oct 19 21:17:42 2004
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 2 receiver 39064 protocol 0
  PATH rcvfrom: 10.1.13.2 (so-0/0/2.0) 10 pkts
  Adspec: received MTU 1500
  PATH sentto: localclient
  RESV rcvfrom: localclient
  Record route: 10.1.36.2 10.1.13.2 <self>
Total 1 displayed,  Up 1 , Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

user@R3> show mpls lsp extensive
Ingress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP:  2 sessions

10.0.0.1
  From: 10.0.0.6, LSPstate: Up, ActiveRoute: 1
   LSPname: R6-to-R1 , LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 3
  Resv style: 1 FF, Label in: 100416, Label out: 3
  Time left:  139, Since: Tue Oct 19 21:05:11 2004
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 2 receiver 39064 protocol 0
  PATH rcvfrom: 10.1.36.2 (so-0/0/3.0) 11 pkts
  Adspec: received MTU 1500 sent MTU 1500
  PATH sentto: 10.1.13.1 (so-0/0/2.0) 11 pkts
  RESV rcvfrom: 10.1.13.1 (so-0/0/2.0) 11 pkts
  Explct route: 10.1.13.1
```

```
  Record route: 10.1.36.2 <self> 10.1.13.1

10.0.0.6
  From: 10.0.0.1, LSPstate: Up, ActiveRoute: 1
   LSPname: R1-to-R6 , LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 3
  Resv style: 1 FF, Label in: 100448, Label out: 3
  Time left:  135, Since: Tue Oct 19 21:10:22 2004
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 47951 protocol 0
  PATH rcvfrom: 10.1.13.1 (so-0/0/2.0) 4 pkts
  Adspec: received MTU 1500 sent MTU 1500
  PATH sentto: 10.1.36.2 (so-0/0/3.0) 4 pkts
  RESV rcvfrom: 10.1.36.2 (so-0/0/3.0) 4 pkts
  Record route: 10.1.13.1 <self> 10.1.36.2
Total 2 displayed,  Up 2 , Down 0

user@R6>  run show mpls lsp extensive
Ingress LSP: 1 sessions

10.0.0.1
  From: 10.0.0.6, State: Up, ActiveRoute: 1, LSPname: R6-to-R1
  ActivePath:  (primary)
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                   State: Up
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 2)
 10.1.36.1 S 10.1.13.1 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt):
          10.1.36.1 10.1.13.1
   19 Oct 19 21:09:52 Selected as active path
   18 Oct 19 21:09:52 Record Route:  10.1.36.1 10.1.13.1
   17 Oct 19 21:09:52 Up
   16 Oct 19 21:09:52 Originate Call
   15 Oct 19 21:09:52 CSPF: computation result accepted
  Created: Tue Oct 19 18:30:09 2004
Total 1 displayed,  Up 1 , Down 0

Egress LSP: 1 sessions

10.0.0.6
  From: 10.0.0.1, LSPstate: Up, ActiveRoute: 0
```

```
    LSPname: R1-to-R6 , LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: -
  Resv style: 1 FF, Label in: 3, Label out: -
  Time left:  120, Since: Tue Oct 19 21:15:03 2004
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 47951 protocol 0
  PATH rcvfrom: 10.1.36.1 (so-0/0/3.0) 4 pkts
  Adspec: received MTU 1500
  PATH sentto: localclient
  RESV rcvfrom: localclient
  Record route: 10.1.13.1 10.1.36.1 <self>
Total 1 displayed,  Up 1 , Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Meaning

The sample output from ingress router **R1** and egress router **R6** shows that the LSP is now traversing the network along the expected path, from **R1** through **R3** to **R6**, and the reverse LSP, from **R6** through **R3** to **R1**. In addition, the sample output from transit router **R3** shows that there are two transit LSP sessions, one from **R1** to **R6**, and the other from **R6** to **R1**.

Checking the RSVP Layer

IN THIS SECTION

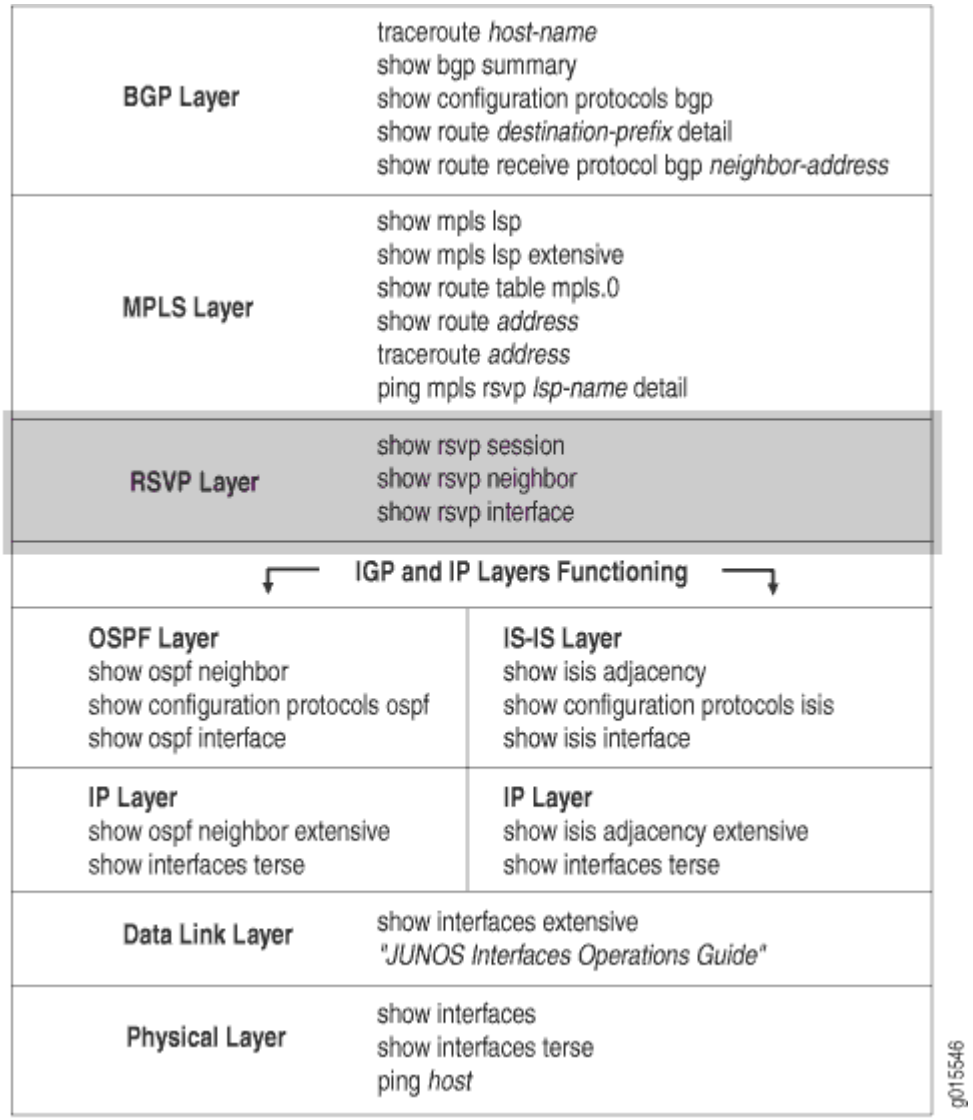
- Verify the LSP | **2424**
- Verify RSVP Sessions | **2426**
- Verify RSVP Neighbors | **2429**
- Verify RSVP Interfaces | **2431**
- Verify the RSVP Protocol Configuration | **2433**
- Take Appropriate Action | **2434**
- Verify the LSP Again | **2436**

Purpose

After you have configured the label-switched path (LSP), issued the `show mpls lsp extensive` command, and determined that there is an error, you might find that the error is not in the physical, data link, or Internet Protocol (IP) and interior gateway protocol (IGP) layers. Continue investigating the problem at the RSVP layer of the network.

Figure 167 on page 2422 illustrates the RSVP layer of the layered MPLS model.

Figure 167: Checking the RSVP Layer

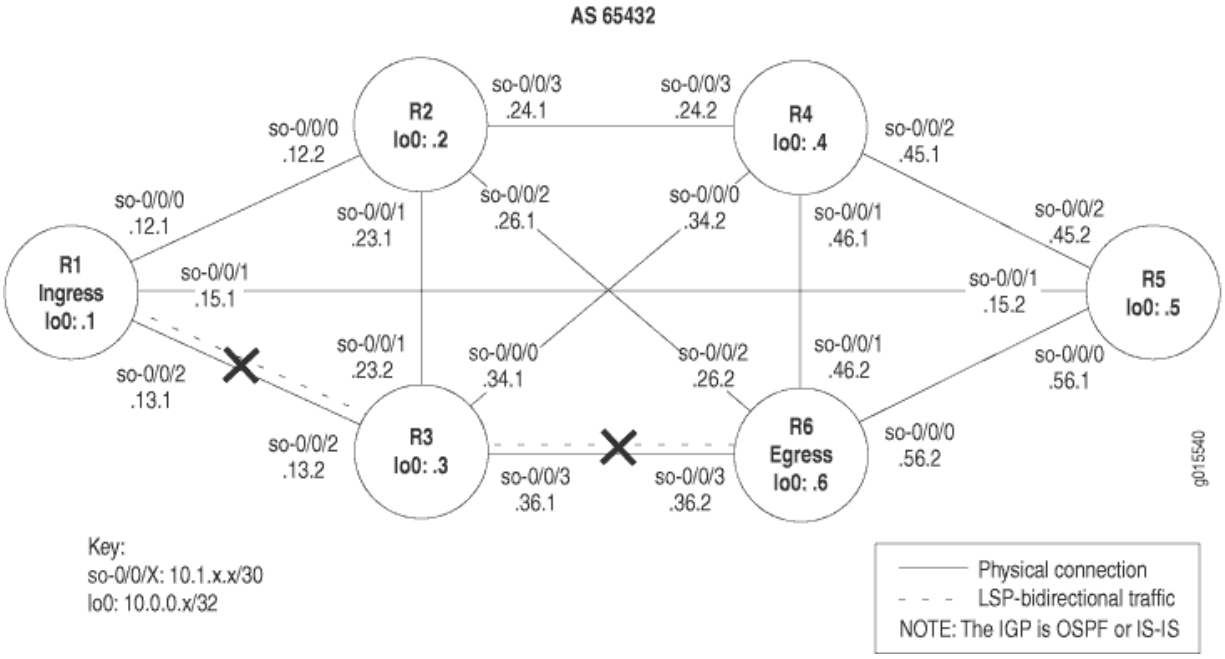


With this layer, you check that dynamic RSVP signaling is occurring as expected, neighbors are connected, and interfaces are configured correctly for RSVP. Check the ingress, egress, and transit routers.

If the network is not functioning at this layer, the LSP does not work as configured.

Figure 168 on page 2423 illustrates the MPLS network used in this topic.

Figure 168: MPLS Network Broken at the RSVP Layer



The network shown in Figure 168 on page 2423 is a fully meshed configuration where every directly connected interface can receive and send packets to every other similar interface. The LSP in this network is configured to run from ingress router **R1**, through transit router **R3**, to egress router **R6**. In addition, a reverse LSP is configured to run from **R6** through **R3** to **R1**, creating bidirectional traffic.

However, in this example, the LSP is down without a path in either direction, from **R1** to **R6** or from **R6** to **R1**.

The crosses shown in Figure 168 on page 2423 indicate where the LSP is broken. Some possible reasons the LSP is broken might include that dynamic RSVP signaling is not occurring as expected, neighbors are not connected, or interfaces are incorrectly configured for RSVP.

In the network in Figure 168 on page 2423, a configuration error on transit router **R3** prevents the LSP from traversing the network as expected.

To check the RSVP layer, follow these steps:

Verify the LSP

IN THIS SECTION

- Purpose | 2424
- Action | 2424
- Meaning | 2426

Purpose

Typically, you use the `show mpls lsp extensive` command to verify the LSP. However for quick verification of the LSP state, use the `show mpls lsp` command. If the LSP is down, use the **extensive** option (`show mpls lsp extensive`) as a follow-up. If your network has numerous LSPs, you might consider specifying the name of the LSP, using the **name** option (`show mpls lsp name` ***name*** or `show mpls lsp name` ***name*** **extensive).**

Action

To determine whether the LSP is up, enter the following command from the ingress router:

```
user@host>              show mpls lsp extensive
```

Sample Output 1

command-name

```
user@R1> show mpls lsp extensive
Ingress LSP: 1 sessions

10.0.0.6
  From: 10.0.0.1, State: Dn, ActiveRoute: 0,  LSPname: R1-to-R6
  ActivePath: (none)
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
  Primary                   State: Dn
    2 Oct 27 15:06:05 10.1.13.2:  No Route toward dest [4 times]
    1 Oct 27 15:05:56 Originate Call
```

```
  Created: Wed Oct 27 15:05:55 2004
Total 1 displayed, Up 0,  Down 1

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

user@R3> show mpls lsp extensive
Ingress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

user@R6> show mpls lsp extensive
Ingress LSP: 1 sessions

10.0.0.1
  From: 10.0.0.6, State: Dn, ActiveRoute: 0,  LSPname: R6-to-R1
  ActivePath: (none)
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
  Primary                   State: Dn
    Will be enqueued for recomputation in 22 second(s).
    1 Oct 27 14:59:12  CSPF failed: no route toward 10.0.0.1 [4 times]
  Created: Wed Oct 27 14:57:44 2004
Total 1 displayed, Up 0,  Down 1

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Meaning

The sample output shows that the LSP is down in both directions, from **R1** to **R6**, and from **R6** to **R1**. The output from **R1** shows that **R1** is using a no-cspf LSP since it tried to originate the call without being able to reach the destination. The output from **R6** shows that the Constrained Shortest Path First (CSPF) algorithm failed, resulting in no route to destination **10.0.0.1**.

Verify RSVP Sessions

IN THIS SECTION

- Purpose | **2426**
- Action | **2426**
- Meaning | **2428**

Purpose

When an RSVP session is successfully created, the LSP is set up along the paths created by the RSVP session. If the RSVP session is unsuccessful, the LSP does not work as configured.

Action

To verify currently active RSVP sessions, enter the following command from the ingress, transit, and egress routers:

```
user@host>              show rsvp session
```

Sample Output 1

command-name

```
user@R1> show rsvp session
Ingress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

Egress RSVP: 0 sessions
```

```
Total 0 displayed, Up 0, Down 0

Transit RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

user@R3> show rsvp session
Ingress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

Egress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

user@R6> show rsvp session
Ingress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

Egress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Sample Output 2

command-name

```
user@R1> show rsvp session
Ingress RSVP: 1 sessions
To              From            State Rt Style Labelin Labelout LSPname
10.0.0.6        10.0.0.1        Up     1 1 FF       -   100768  R1-to-R6
Total 1 displayed,  Up 1 , Down 0

Egress RSVP: 1 sessions
To              From            State Rt Style Labelin Labelout LSPname
10.0.0.1        10.0.0.6        Up     0 1 FF       3        -  R6-to-R1
Total 1 displayed,  Up 1 , Down 0

Transit RSVP: 0 sessions
```

```
Total 0 displayed, Up 0, Down 0

user@R3> show rsvp session
Ingress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

Egress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit RSVP:  2 sessions
To              From            State Rt Style Labelin Labelout LSPname
10.0.0.1        10.0.0.6        Up     1 1 FF  100784        3  R6-to-R1
10.0.0.6        10.0.0.1        Up     1 1 FF  100768        3  R1-to-R6
Total 2 displayed,  Up 2 , Down 0

user@R6> show rsvp session
Ingress RSVP: 1 sessions
To              From            State Rt Style Labelin Labelout LSPname
10.0.0.1        10.0.0.6        Up     1 1 FF       -   100784  R6-to-R1
Total 1 displayed,  Up 1 , Down 0

Egress RSVP: 1 sessions
To              From            State Rt Style Labelin Labelout LSPname
10.0.0.6        10.0.0.1        Up     0 1 FF       3        -  R1-to-R6
Total 1 displayed,  Up 1 , Down 0

Transit RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Meaning

Sample Output 1 from all routers shows that no RSVP sessions were successfully created, even though the LSP **R6-to-R1** is configured.

In contrast to Sample Output 1and to illustrate correct output, Sample Output 2 shows the output from the ingress, transit, and egress routers when the RSVP configuration is correct, and the LSP is traversing the network as configured. **R1** and **R6** both show an ingress and egress RSVP session, with the LSP **R1-to-R6**, and the reverse LSP **R6-to-R1**. Transit router **R3** shows two transit RSVP sessions.

Verify RSVP Neighbors

IN THIS SECTION

- Purpose | 2429
- Action | 2429
- Meaning | 2430

Purpose

Display a list of RSVP neighbors that were learned dynamically when exchanging RSVP packets. Once a neighbor is learned, it is never removed from the list of RSVP neighbors unless the RSVP configuration is removed from the router.

Action

To verify RSVP neighbors, enter the following command from the ingress, transit, and egress routers:

```
user@host>              show rsvp neighbor
```

Sample Output 1

command-name

```
user@R1> show rsvp neighbor
RSVP neighbor: 1 learned
Address            Idle Up/Dn LastChange HelloInt HelloTx/Rx MsgRcvd
10.1.13.2            10   1/0         9:22        9    64/64   32

user@R3> show rsvp neighbor
RSVP neighbor: 2 learned
Address            Idle Up/Dn LastChange HelloInt HelloTx/Rx MsgRcvd
10.1.13.1             0  1/0       28:20        9   190/190  41
10.1.36.2         16:50   1/1        15:37        9   105/78   38

user@R6> show rsvp neighbor
```

```
RSVP neighbor: 1 learned
Address            Idle Up/Dn LastChange HelloInt HelloTx/Rx MsgRcvd
10.1.36.1         17:30   1/1        16:15        9   104/78   39
```

Sample Output 2

command-name

```
user@R3> show rsvp neighbor
RSVP neighbor: 2 learned
Address            Idle Up/Dn LastChange HelloInt HelloTx/Rx MsgRcvd
10.1.13.1             5   1/0         9:14        9    63/63   33
10.1.36.2             5   1/0         9:05        9    62/62   32

user@R6> show rsvp neighbor
RSVP neighbor: 1 learned
Address            Idle Up/Dn LastChange HelloInt HelloTx/Rx MsgRcvd
10.1.36.1             5   1/0        8:54        9    61/61   32
```

Meaning

Sample Output 1 shows that **R1** and **R6** have one RSVP neighbor each, **R3**. However, the values in the **Up/Dn** field are different. **R1** has a value of **1/0** and **R6** has a value of **1/1**, indicating that **R1** is an active neighbor with **R3**, but **R6** is not. When the up count is one more than the down count, the neighbor is active; if the values are equal, the neighbor is down. The values for **R6** are equal, **1/1**, indicating that the neighbor **R3** is down.

Transit router **R3** knows about two neighbors, **R1** and **R6**. The **Up/Dn** field indicates that **R1** is an active neighbor and **R6** is down. At this point it is not possible to determine if the problem resides with **R3** or **R6**, because both neighbors are not active.

In contrast to Sample Output 1 and to illustrate correct output, Sample Output 2 shows the correct neighbor relationship between transit router **R3** and egress router **R6**. The **Up/Dn** field shows the up count to be one more than the down count, **1/0**, indicating that the neighbors are active.

Verify RSVP Interfaces

IN THIS SECTION

- Purpose | 2431
- Action | 2431
- Meaning | 2433

Purpose

Display the status of each interface on which RSVP is enabled to determine where the configuration error occurred.

Action

To verify the status of RSVP interfaces, enter the following command from the ingress, transit, and egress routers:

```
user@host>              show rsvp interface
```

Sample Output 1

command-name

```
user@R1> show rsvp interface
RSVP interface: 3 active
                   Active  Subscr- Static      Available   Reserved    Highwater
Interface   State  resv    iption  BW          BW          BW          mark
so-0/0/0.0  Up         0   100%  155.52Mbps  155.52Mbps  0bps        0bps
so-0/0/1.0  Up         0   100%  155.52Mbps  155.52Mbps  0bps        0bps
so-0/0/2.0  Up          0    100%  155.52Mbps  155.52Mbps  0bps        0bps

user@R3> show rsvp interface
RSVP interface: 3 active
                   Active  Subscr- Static      Available   Reserved    Highwater
Interface   State  resv    iption  BW          BW          BW          mark
```

```
so-0/0/0.0  Up         0   100%  155.52Mbps  155.52Mbps  0bps        0bps
so-0/0/1.0  Up         0   100%  155.52Mbps  155.52Mbps  0bps        0bps
so-0/0/2.0  Up          0    100%  155.52Mbps  155.52Mbps  0bps        0bps
<<< Missing interface so-0/0/3.0

user@R6> show rsvp interface
RSVP interface: 4 active
                  Active  Subscr- Static      Available   Reserved    Highwater
Interface   State  resv    iption  BW          BW          BW          mark
so-0/0/0.0  Up         0   100%  155.52Mbps  155.52Mbps  0bps        0bps
so-0/0/1.0  Up         0   100%  155.52Mbps  155.52Mbps  0bps        0bps
so-0/0/2.0  Up         0   100%  155.52Mbps  155.52Mbps  0bps        0bps
so-0/0/3.0  Up          0    100%  155.52Mbps  155.52Mbps  0bps        0bps
```

Sample Output 2

command-name

```
user@R1> show rsvp interface
RSVP interface: 3 active
                  Active Subscr- Static      Available   Reserved    Highwater
Interface   State resv   iption  BW          BW          BW          mark
so-0/0/0.0  Up         0   100%  155.52Mbps  155.52Mbps  0bps        0bps
so-0/0/1.0  Up         0   100%  155.52Mbps  155.52Mbps  0bps        0bps
so-0/0/2.0  Up          1    100%  155.52Mbps  155.52Mbps  0bps        0bps

user@R3> show rsvp interface
RSVP interface: 4 active
                  Active Subscr- Static      Available   Reserved    Highwater
Interface   State resv   iption  BW          BW          BW          mark
so-0/0/0.0  Up         0   100%  155.52Mbps  155.52Mbps  0bps        0bps
so-0/0/1.0  Up         0   100%  155.52Mbps  155.52Mbps  0bps        0bps
so-0/0/2.0  Up          1    100%  155.52Mbps  155.52Mbps  0bps        0bps
so-0/0/3.0  Up          1    100%  155.52Mbps  155.52Mbps  0bps        0bps

user@R6> show rsvp interface
RSVP interface: 4 active
                  Active Subscr- Static      Available   Reserved    Highwater
Interface   State resv   iption  BW          BW          BW          mark
so-0/0/0.0  Up         0   100%  155.52Mbps  155.52Mbps  0bps        0bps
so-0/0/1.0  Up         0   100%  155.52Mbps  155.52Mbps  0bps        0bps
```

```
so-0/0/2.0  Up         0   100%  155.52Mbps  155.52Mbps  0bps        0bps
so-0/0/3.0  Up          1    100%  155.52Mbps  155.52Mbps  0bps        0bps
```

Meaning

Sample Output 1 shows that even though each router has interfaces that are up and have RSVP active, there are no reservations (**Active resv**) on any of the routers. In this example, we would expect at least one reservation on the ingress and egress routers, and two reservations on the transit router.

In addition, interface **so-0/0/3** on transit router **R3** is not included in the configuration. The inclusion of this interface is critical to the success of the LSP.

In contrast to Sample Output 1 and to illustrate correct output, Sample Output 2 shows the relevant interfaces with active reservations.

Verify the RSVP Protocol Configuration

IN THIS SECTION

- Purpose | **2433**
- Action | **2433**
- Meaning | **2434**

Purpose

After you have checked RSVP sessions, interfaces, neighbors, and determined that there might be a configuration error, verify the RSVP protocol configuration.

Action

To verify the RSVP configuration, enter the following command from the ingress, transit, and egress routers:

```
user@host>              show configuration protocols rsvp
```

Sample Output

command-name

```
user@R1> show configuration protocols rsvp
interface so-0/0/0.0;
interface so-0/0/1.0;
interface so-0/0/2.0;
interface fxp0.0 {
    disable;
}

user@R3> show configuration protocols rsvp
interface so-0/0/0.0;
interface so-0/0/1.0;
interface so-0/0/2.0; <<< Missing interface so-0/0/3.0
interface fxp0.0 {
    disable;
}

user@R6> show configuration protocols rsvp
interface so-0/0/0.0;
interface so-0/0/1.0;
interface so-0/0/2.0;
interface so-0/0/3.0;
interface fxp0.0 {
    disable;
}
```

Meaning

The sample output shows that **R3** has interface **so-0/0/3.0** missing from the RSVP protocol configuration. This interface is critical for the correct functioning of the LSP.

Take Appropriate Action

IN THIS SECTION

Problem | 2435

- Solution | 2435

Problem

Description

Depending on the error you encountered in your investigation, you must take the appropriate action to correct the problem. In this example, an interface is missing from the configuration of router R3.

Solution

To correct the error in this example, follow these steps:

1. Include the missing interface in the configuration of transit router R3:

```
user@R3> edit
user@R3#                    edit protocols rsvp
[edit protocols rsvp]
user@R3# show
user@R3#                    set interface so-0/0/3.0
```

2. Verify and commit the configuration:

```
[edit protocols rsvp]
user@R3# show
user@R3# commit
```

Sample Output

```
user@R3> edit
Entering configuration mode

[edit]
user@R3# edit protocols rsvp

[edit protocols rsvp]
```

```
user@R3# show
interface so-0/0/0.0;
interface so-0/0/1.0;
interface so-0/0/2.0; <<< Missing interface so-0/0/3.0
interface fxp0.0 {
    disable;
}
[edit protocols rsvp]
user@R3# set interface so-0/0/3.0

[edit protocols rsvp]
user@R3# show
interface so-0/0/0.0;
interface so-0/0/1.0;
interface so-0/0/2.0;
interface fxp0.0 {
    disable;
}
interface so-0/0/3.0; <<< Interface now included in the configuration

[edit protocols rsvp]
user@R3# commit
commit complete
```

Meaning

The sample output shows that the missing interface **so-0/0/3.0** on transit router **R3** is now correctly included at the [**edit protocols rsvp**] hierarchy level. This results in the possibility that the LSP might come up.

Verify the LSP Again

IN THIS SECTION

- Purpose | 2437
- Action | 2437
- Meaning | 2440

Purpose

After taking the appropriate action to correct the error, the LSP needs to be checked again to confirm that the problem in the MPLS layer has been resolved.

Action

To verify the LSP again, enter the following command on the ingress, transit, and egress routers:

```
user@host>              show mpls lsp extensive
```

Sample Output 1

command-name

```
user@R1> show mpls lsp extensive
Ingress LSP: 1 sessions

10.0.0.6
  From: 10.0.0.1, State: Up,  ActiveRoute: 1 ,  LSPname: R1-to-R6
  ActivePath:  (primary)
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                   State: Up
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt):
          10.1.13.2 10.1.36.2
    5 Oct 27 15:28:57 Selected as active path
    4 Oct 27 15:28:57  Record Route:  10.1.13.2 10.1.36.2
    3 Oct 27 15:28:57 Up
    2 Oct 27 15:28:44 10.1.13.2: No Route toward dest[35 times]
    1 Oct 27 15:05:56 Originate Call
  Created: Wed Oct 27 15:05:56 2004
Total 1 displayed, Up 1, Down 0

Egress LSP: 1 sessions

10.0.0.1
  From: 10.0.0.6, LSPstate: Up, ActiveRoute: 0
  LSPname: R6-to-R1, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
```

```
  Recovery label received: -, Recovery label sent: -
  Resv style: 1 FF, Label in: 3, Label out: -
  Time left:  136, Since: Wed Oct 27 15:29:20 2004
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 39092 protocol 0
  PATH rcvfrom: 10.1.13.2 (so-0/0/2.0) 6 pkts
  Adspec: received MTU 1500
  PATH sentto: localclient
  RESV rcvfrom: localclient
  Record route: 10.1.36.2 10.1.13.2 <self>
Total 1 displayed, Up 1, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Sample Output 2

command-name

```
user@R3> show mpls lsp extensive
Ingress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 2 sessions

10.0.0.1
  From: 10.0.0.6, LSPstate: Up, ActiveRoute: 1
  LSPname: R6-to-R1, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 3
  Resv style: 1 FF, Label in: 100672, Label out: 3
  Time left:  152, Since: Wed Oct 27 15:16:39 2004
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 39092 protocol 0
  PATH rcvfrom: 10.1.36.2 (so-0/0/3.0) 7 pkts
```

```
  Adspec: received MTU 1500 sent MTU 1500
  PATH sentto: 10.1.13.1 (so-0/0/2.0) 7 pkts
  RESV rcvfrom: 10.1.13.1 (so-0/0/2.0) 7 pkts
  Explct route: 10.1.13.1
  Record route: 10.1.36.2 <self> 10.1.13.1

10.0.0.6
  From: 10.0.0.1, LSPstate: Up, ActiveRoute: 1
  LSPname: R1-to-R6, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 3
  Resv style: 1 FF, Label in: 100656, Label out: 3
  Time left:  129, Since: Wed Oct 27 14:53:14 2004
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 47977 protocol 0
  PATH rcvfrom: 10.1.13.1 (so-0/0/2.0) 40 pkts
  Adspec: received MTU 1500 sent MTU 1500
  PATH sentto: 10.1.36.2 (so-0/0/3.0) 7 pkts
  RESV rcvfrom: 10.1.36.2 (so-0/0/3.0) 7 pkts
  Record route: 10.1.13.1 <self> 10.1.36.2
Total 2 displayed, Up 2, Down 0
```

Sample Output 3

command-name

```
user@R6> show mpls lsp extensive
Ingress LSP: 1 sessions

10.0.0.1
  From: 10.0.0.6, State: Up,  ActiveRoute: 1 ,  LSPname: R6-to-R1
  ActivePath:  (primary)
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                    State: Up
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 20)
 10.1.36.1 S 10.1.13.1 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt):
          10.1.36.1 10.1.13.1
    6 Oct 27 15:22:06 Selected as active path
    5 Oct 27 15:22:06 Record Route:  10.1.36.1 10.1.13.1
```

```
    4 Oct 27 15:22:06 Up
    3 Oct 27 15:22:06 Originate Call
    2 Oct 27 15:22:06 CSPF: computation result accepted
    1 Oct 27 15:21:36 CSPF failed: no route toward 10.0.0.1[50 times]
  Created: Wed Oct 27 14:57:45 2004
Total 1 displayed, Up 1, Down 0

Egress LSP: 1 sessions

10.0.0.6
  From: 10.0.0.1, LSPstate: Up, ActiveRoute: 0
  LSPname: R1-to-R6, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: -
  Resv style: 1 FF, Label in: 3, Label out: -
  Time left:  119, Since: Wed Oct 27 15:21:43 2004
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 47977 protocol 0
  PATH rcvfrom: 10.1.36.1 (so-0/0/3.0) 7 pkts
  Adspec: received MTU 1500
  PATH sentto: localclient
  RESV rcvfrom: localclient
  Record route: 10.1.13.1 10.1.36.1 <self>
Total 1 displayed, Up 1, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Meaning

Sample Output 1 from ingress router **R1** shows that LSP **R1-to-R6** has an active route to **R6** and the state is up.

Sample Output 2 from transit router **R3** shows that there are two transit LSP sessions, one from **R1** to **R6** and the other from **R6** to **R1**. Both LSPs are up.

Sample Output 3 from egress router **R6** shows that the LSP is up and the active route is the primary route. The LSP is now traversing the network along the expected path, from **R1** through **R3** to **R6**, and the reverse LSP, from **R6** through **R3** to **R1**.

Determining LSP Statistics

IN THIS SECTION

- Purpose | 2441
- Action | 2441
- Meaning | 2442

Purpose

Display detailed information about RSVP objects to assist the diagnosis of an LSP problem.

Action

To verify RSVP objects, enter the following Junos OS CLI operational mode command:

```
user@host>                      show rsvp session detail
```

Sample Output

command-name

```
user@R1> show rsvp session detail
Ingress RSVP: 1 sessions

10.0.0.6
  From: 10.0.0.1, LSPstate: Up, ActiveRoute: 1
  LSPname:  R1-to-R6 , LSPpath:  Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent:  100064
  Resv style: 1 FF, Label in: -, Label out: 100064
  Time left:    -, Since: Tue Aug 17 12:22:52 2004
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 12 receiver 44251 protocol 0
  PATH rcvfrom: localclient
  Adspec: sent MTU 1500
  PATH sentto: 10.1.13.2 (so-0/0/2.0) 182 pkts
```

```
  RESV rcvfrom: 10.1.13.2 (so-0/0/2.0) 159 pkts
  Explct route: 10.1.13.2 10.1.36.2
  Record route: <self> 10.1.13.2 10.1.36.2
Total 1 displayed, Up 1, Down 0

Egress RSVP: 1 sessions

10.0.0.1
  From:  10.0.0.6 , LSPstate: Up, ActiveRoute: 0
  LSPname: R6-to-R1, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: -
  Resv style: 1 FF, Label in: 3, Label out: -
  Time left:  135, Since: Tue Aug 17 12:23:14 2004
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 39024 protocol 0
  PATH rcvfrom: 10.1.15.2 (so-0/0/1.0) 158 pkts
  Adspec: received MTU 1500
  PATH sentto: localclient
  RESV rcvfrom: localclient
  Record route: 10.1.56.2 10.1.15.2 <self>
Total 1 displayed, Up 1, Down 0

Transit RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Meaning

The sample output shows that there is one ingress and one egress RSVP session. The ingress session has a source address of **10.0.0.1** (**R1**), and the session is up, with one active route. The LSP name is **R1-to-R6** and it is the primary path for the LSP.

The recovery label (**100064**) is sent by a graceful restart router to its neighbor to recover a forwarding state. It is probably the old label that the router advertised before it went down.

This session is using the fixed filter (**FF**) reservation style (**Resv style**). Since this is an ingress router, there is no inbound label. The outbound label (provided by the next downstream router) is **100064**.

The **Time Left** field provides the number of seconds remaining in the RSVP session, and the **Tspec** object provides information about the controlled load rate (**rate**) and maximum burst size (**peak**), an infinite value (**Infbps**) for the guaranteed delivery option, and the indication that packets smaller than 20 bytes are treated as 20 bytes, while packets larger than 1500 bytes are treated as 1500 bytes.

The port number is the IPv4 tunnel ID, while the sender/receiver port number is the LSP ID. The IPv4 tunnel ID is unique for the life of the LSP, while the sender/receiver LSP ID can change, for example, with an SE style reservation.

The **PATH rcvfrom** field includes the source of the path message. Since this is the ingress router, the local client originated the path message.

The **PATH sentto** field includes the path message destination (**10.1.13.2**) and outgoing interface (**so-0/0/2.0**). The **RESV rcvfrom** field includes both the source of the Resv message received (**10.1.13.2**) and the incoming interface (**so-0/0/2.0**).

The RSVP explicit route and the route record values are identical: **10.1.13.2** and **10.1.36.2**. In most cases, the explicit route and the record route values are identical. Differences indicate that some path rerouting has occurred, typically during Fast-Reroute.

The **Total** fields indicate the total number of ingress, egress, and transit RSVP sessions, with the total being equal to the sum of the up and down sessions. In this example, there is one ingress session, one egress session, and no transit RSVP sessions.

Verifying LSP Use in Your Network

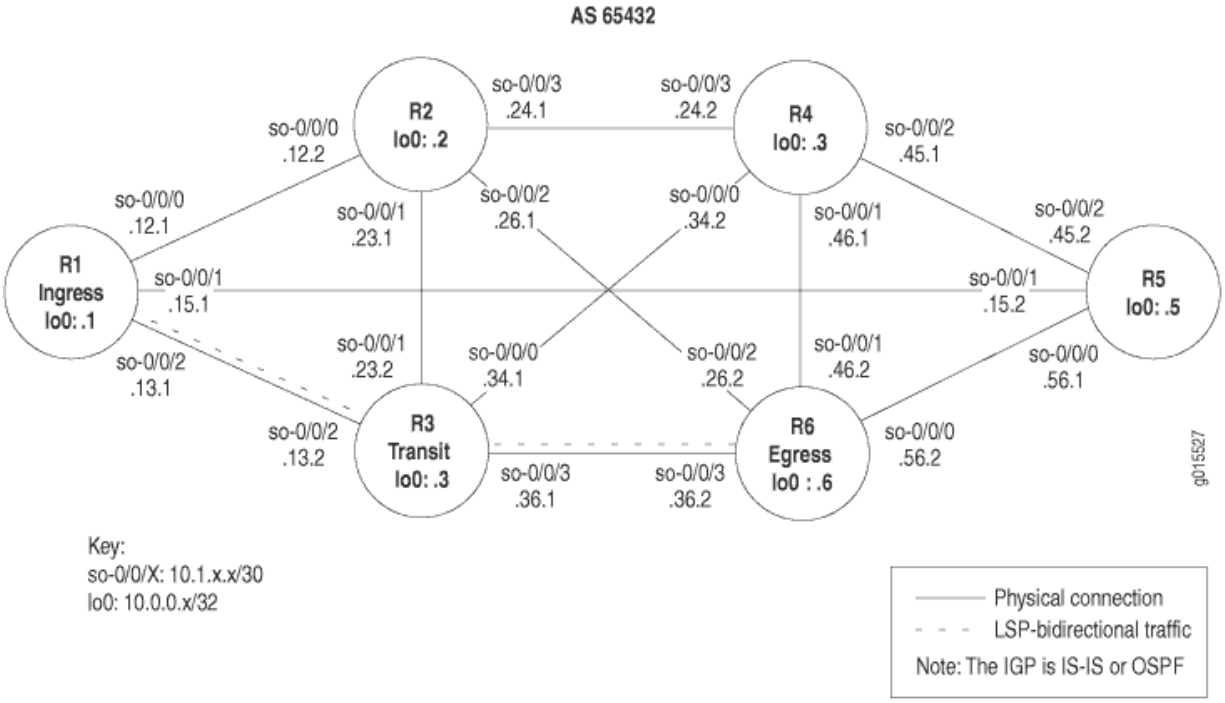
IN THIS SECTION

- Verifying an LSP on the Ingress Router | **2444**
- Verifying an LSP on a Transit Router | **2446**

Purpose

When you verify the valid use of an LSP on the ingress and transit routers in your network, you can determine if there is a problem with Multiprotocol Label Switching (MPLS) in your network. Figure 169 on page 2444 describes the example network used in this topic.

Figure 169: MPLS Topology for Verifying LSP Use



The MPLS network in Figure 169 on page 2444 illustrates a router-only network with SONET interfaces that consist of the following components:

- A full-mesh interior Border Gateway Protocol (IBGP) topology, using AS 65432
- MPLS and Resource Reservation Protocol (RSVP) enabled on all routers
- A **send-statics** policy on routers R1 and R6 that allows a new route to be advertised into the network
- An LSP between routers R1 and R6

The network shown in Figure 169 on page 2444 is a Border Gateway Protocol (BGP) full-mesh network. Since route reflectors and confederations are not used to propagate BGP learned routes, each router must have a BGP session with every other router running BGP.

To verify LSP use in your network, follow these steps:

Verifying an LSP on the Ingress Router

IN THIS SECTION

Purpose | 2445

Action | 2445
Meaning | 2445

Purpose

You can verify the availability of an LSP when it is up by examining the **inet.3** routing table on the ingress router. The **inet.3** routing table contains the host address of each LSP's egress router. This routing table is used on ingress routers to route BGP packets to the destination egress router. BGP uses the **inet.3** routing table on the ingress router to help resolve next-hop addresses.

Action

To verify an LSP on an ingress router, enter the following Junos OS command-line interface (CLI) operational mode command:

```
user@host> show route table inet.3
```

Sample Output

command-name

```
user@R1> show route table inet.3
inet.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.6/32        *[RSVP/7] 4w0d 22:40:57, metric 20
                    > via so-0/0/2.0, label-switched-path R1-to-R6
```

Meaning

The sample output shows the **inet.3** routing table. By default, only BGP and MPLS virtual private networks (VPNs) can use the **inet.3** route table to resolve next-hop information. One destination is listed in the route table, **10.0.0.6**. This destination (**10.0.0.6**) is signaled by RSVP, and is the current active path, as indicated by the asterisk (*****). The protocol preference for this route is **7**, and the metric associated with it is **20**. The label-switched path is **R1-to-R6**, through interface **so-0/0/2.0**, which is the physical next-hop transit interface.

Typically, the penultimate router in the LSP either pops the packet's label or changes the label to a value of 0. If the penultimate router pops the top label and an IPv4 packet is underneath, the egress router routes the IPv4 packet, consulting the IP routing table **inet.0** to determine how to forward the packet. If another type of label (such as one created by Label Distribution Protocol (LDP) tunneling or VPNs, but not IPv4) is underneath the top label, the egress router does not examine the **inet.0** routing table. Instead, it examines the **mpls.0** routing table for forwarding decisions.

If the penultimate router changes the packet's label to a value of 0, the egress router strips off the 0 label, indicating that an IPv4 packet follows. The packet is examined by the **inet.0** routing table for forwarding decisions.

When a transit or egress router receives an MPLS packet, information in the MPLS forwarding table is used to determine the next transit router in the LSP or whether this router is the egress router.

When BGP resolves a next-hop prefix, it examines both the **inet.0** and **inet.3** routing tables, seeking the next hop with the lowest preference; for example, RSVP preference 7 is preferred over OSPF preference 10. The RSVP signaled LSP is used to reach the BGP next hop. This is the default when the BGP next hop equals the LSP egress address. Once the BGP next hop is resolved through an LSP, the BGP traffic uses the LSP to forward BGP transit traffic.

Verifying an LSP on a Transit Router

IN THIS SECTION

- Purpose | **2446**
- Action | **2446**
- Meaning | **2447**

Purpose

You can verify the availability of an LSP when it is up by examining the **mpls.0** routing table on a transit router. MPLS maintains the **mpls.0** routing table, which contains a list of the next label-switched router in each LSP. This routing table is used on transit routers to route packets to the next router along an LSP.

Action

To verify an LSP on a transit router, enter the following Junos OS CLI operational mode command:

```
user@host> show route table mpls.0
```

Sample Output

command-name

```
user@R3> show route table mpls.0
mpls.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                   *[MPLS/0] 7w3d 22:20:56, metric 1
                     Receive
1                   *[MPLS/0] 7w3d 22:20:56, metric 1
                     Receive
2                   *[MPLS/0] 7w3d 22:20:56, metric 1
                     Receive
100064              *[RSVP/7] 2w1d 04:17:36, metric 1
                    > via so-0/0/3.0, label-switched-path R1-to-R6
100064 (S=0)         *[RSVP/7] 2w1d 04:17:36, metric 1
                   > via so-0/0/3.0, label-switched-path R1-to-R6
```

Meaning

The sample output from transit router **R3** shows route entries in the form of MPLS label entries, indicating that there is only one active route, even though there are five active entries.

The first three MPLS labels are reserved MPLS labels defined in RFC 3032. Packets received with these label values are sent to the Routing Engine for processing. Label 0 is the IPv4 explicit null label. Label 1 is the MPLS equivalent of the IP Router Alert label and Label 2 is the IPv6 explicit null label.

The two entries with the **100064** label are for the same LSP, **R1-to-R6.** There are two entries because the stack values in the MPLS header may be different. The second entry, **100064 (S=0)**, indicates that the stack depth is not 1 and additional label values are included in the packet. In contrast, the first entry of **100064** has an inferred S=1 which indicates a stack depth of 1 and makes it the last label in the packet. The dual entry indicates that this is the penultimate router. For more information on MPLS label stacking, see RFC 3032, *MPLS Label Stack Encoding.*

The incoming label is the MPLS header of the MPLS packet, and is assigned by RSVP to the upstream neighbor. Juniper Networks routers dynamically assign labels for RSVP traffic-engineered LSPs in the range from 100,000 through 1,048,575.

The router assigns labels starting at label 100,000, in increments of 16. The sequence of label assignments is 100,000, 100,016, 100,032, 100,048, and so on. At the end of the assigned labels, the label numbers start over at 100001, incrementing in units of 16. Juniper Networks reserves labels for various purposes. Table 1 lists the various label range allocations for incoming labels.

Table 51: MPLS Label Range Allocations

Incoming Label	Status
0 through **15**	Reserved by IETF
16 through **1023**	Reserved for static LSP assignment
1024 through **9999**	Reserved for internal use (for example, CCC labels)
10,000 through **99,999**	Reserved for static LSP assignment
100,000 through **1,048,575**	Reserved for dynamic label assignment

Verify That Load Balancing Is Working

IN THIS SECTION

- Purpose | **2448**
- Action | **2448**
- Meaning | **2449**

Purpose

After configuring load balancing, check that traffic is load-balanced equally across paths. In this section, the command output reflects the load-balancing configuration of the example network shown in "Load-Balancing Network Topology" on page 241. The `clear` commands are used to reset LSP and interface counters to zero so that the values reflect the operation of the load-balancing configuration.

Action

To verify load balancing across interfaces and LSPs, use the following command on the ingress router:

```
user@host# show configuration
```

To verify load balancing across interfaces and LSPs, use the following commands on a transit router:

```
user@host# show route
user@host# show route forwarding-table
user@host# show mpls lsp statistics
user@host# monitor interface traffic
user@host# clear mpls lsp statistics
user@host# clear interface statistics
```

Sample Output

command-name

The following sample output is for the configuration on ingress router **R1**:

```
user@R1> show configuration | no-more
[...Output truncated...]
routing-options {
    [...Output truncated...]
     forwarding-table {
        export lbpp;
    }
}
[...Output truncated...]
policy-options {
     policy-statement lbpp {
        then {
            load-balance per-packet;
        }
    }
}
```

Meaning

The sample output for the `show configuration` command on ingress router **R1** shows that load balancing is correctly configured with the **lbpp** policy statement. Also, the **lbpp** policy is exported into the forwarding table at the `[edit routing-options]` hierarchy level.

Sample Output

The following sample output is from transit router R2:

```
user@R2> show route 192.168.0.1 terse

inet.0: 25 destinations, 27 routes (25 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
A Destination        P Prf   Metric 1   Metric 2  Next hop        AS path
* 192.168.0.1/32     O  10          3             so-0/0/1.0
                                                 >so-0/0/2.0
[...Output truncated...]
```

Meaning

The sample output for the `show route` command issued on transit router **R2** shows the two equal-cost paths (**so-0/0/1** and **so-0/0/2)** through the network to the loopback address to **R0** (**192.168.0.1**). Even though the right angle bracket (>) usually indicates the active route, in this instance it does not, as shown in the following four sample outputs.

Sample Output

The following sample output is from transit router R2:

```
user@R2> monitor interface traffic

R2                                Seconds: 65                 Time: 11:41:14

Interface    Link  Input packets        (pps)     Output packets        (pps)
 so-0/0/0      Up              0          (0)                 0          (0)
 so-0/0/1      Up            126          (0)            164659       (2128)
 so-0/0/2      Up          85219       (1004)            164598       (2128)
 so-0/0/3      Up              0          (0)                 0          (0)
 fe-0/1/0      Up         328954       (4265)             85475       (1094)
 fe-0/1/1      Up              0          (0)                 0          (0)
 fe-0/1/2      Up              0          (0)                 0          (0)
 fe-0/1/3      Up              0          (0)                 0          (0)
[...Output truncated...]
```

Meaning

The sample output for the `monitor interface traffic` command issued on transit router **R2** shows that output traffic is evenly distributed across the two interfaces **so-0/0/1** and **so-0/0/2**.

Sample Output

The following sample output is from transit router R2:

```
user@R2> show mpls lsp statistics
Ingress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 5 sessions
To              From            State     Packets            Bytes LSPname
192.168.0.1     192.168.1.1     Up          87997         17951388 lsp1
192.168.0.1     192.168.1.1     Up          87997         17951388 lsp2
192.168.0.1     192.168.1.1     Up          87997         17951388 lsp3
192.168.0.1     192.168.1.1     Up          87997         17951388 lsp4
192.168.6.1     192.168.0.1     Up              0                0 r0-r1
Total 5 displayed, Up 5, Down 0
```

Meaning

The sample output for the `show mpls lsp statistics` command issued on transit router **R2** shows that output traffic is evenly distributed across the four LSPs configured on ingress router **R6**.

Sample Output

The following sample output is from transit router R2:

```
user@R2> show route forwarding-table destination 10.0.90.14
Routing table: inet
Internet:
Destination        Type RtRef Next hop           Type Index NhRef Netif
10.0.90.12/30      user     0                     ulst 262144     6
```

```
                                                ucst   345     5 so-0/0/1.0
                                                ucst   339     2 so-0/0/2.0
```

Meaning

The sample output for the `show route forwarding-table destination` command issued on transit router **R2** shows **ulst** in the **Type** field, which indicates that load balancing is working. The two unicast (**ucst)** entries in the **Type** field are the two next hops for the LSPs.

Sample Output

The following sample output is from transit router R2:

```
user@R2> show route forwarding-table | find mpls
Routing table: mpls
MPLS:
Destination        Type RtRef Next hop           Type Index NhRef Netif
default            perm     0                    dscd    38     1
0                  user     0                    recv    37     3
1                  user     0                    recv    37     3
2                  user     0                    recv    37     3
100112             user     0                   Swap 100032       so-0/0/1.0
100128             user     0                   Swap 100048       so-0/0/1.0
100144             user     0 10.0.12.13        Swap 100096       fe-0/1/0.0
100160             user     0                   Swap 100112       so-0/0/2.0
100176             user     0                   Swap 100128       so-0/0/2.0
```

Meaning

The sample output for the `show route forwarding-table | find mpls` command issued on transit router R2 shows the MPLS routing table that contains the labels received and used by this router to forward packets to the next-hop router. This routing table is used mostly on transit routers to route packets to the next router along an LSP. The first three labels in the **Destination** column (Label 0, Label 1, and Label 2) are automatically entered by MPLS when the protocol is enabled. These labels are reserved MPLS labels defined in RFC 3032. Label 0 is the IPv4 explicit null label. Label 1 is the MPLS equivalent of the IP Router Alert label, and Label 2 is the IPv6 explicit null label.

The remaining five labels in the **Destination** column are nonreserved labels that the router uses to forward traffic, and the last column **Netif**, shows the interfaces used to send the labeled traffic. For nonreserved labels, the second **Type** column shows the operation performed on matching packets. In this example, all non-reserved packets are swapped for outgoing packet labels. For example, packets

with the label **100112** have their label swapped for **100032** before they are pushed out of interface **so-0/0/1.0**.

Verify the Operation of Uneven Bandwidth Load Balancing

IN THIS SECTION

- Purpose | **2453**
- Action | **2453**
- Meaning | **2455**

Purpose

When a router is performing unequal-cost load balancing between LSPs paths, the `show route detail` command displays a balance field associated with each next hop being used.

Action

To verify that an RSVP LSP is unevenly load-balanced, use the following Junos OS CLI operational mode commands:

```
user@host> show route protocol rsvp detail
user@host> show mpls lsp statistics
```

Sample Output

command-name

```
user@R1> show route protocol rsvp detail

inet.0: 25 destinations, 25 routes (25 active, 0 holddown, 0 hidden)
10.0.90.14/32 (1 entry, 1 announced)
        State: <FlashAll>
        *RSVP   Preference: 7
                Next-hop reference count: 7
                Next hop: 10.0.12.14 via fe-0/1/0.0 weight 0x1  balance 10%
                 Label-switched-path lsp1
```

```
                Label operation: Push 100768
                Next hop: 10.0.12.14 via fe-0/1/0.0 weight 0x1  balance 20%
                 Label-switched-path lsp2
                Label operation: Push 100736
                Next hop: 10.0.12.14 via fe-0/1/0.0 weight 0x1  balance 30%,
                        selected
                 Label-switched-path lsp3
                Label operation: Push 100752
                Next hop: 10.0.12.14 via fe-0/1/0.0 weight 0x1  balance 40%
                 Label-switched-path lsp4
                Label operation: Push 100784
                State: <Active Int>
                Local AS: 65432
                Age: 8:03       Metric: 4
                Task: RSVP
                Announcement bits (2): 0-KRT 4-Resolve tree 1
                AS path: I
inet.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
192.168.0.1/32 (1 entry, 1 announced)
        State: <FlashAll>
        *RSVP   Preference: 7
                Next-hop reference count: 7
                Next hop: 10.0.12.14 via fe-0/1/0.0 weight 0x1 balance 10%
                Label-switched-path lsp1
                Label operation: Push 100768
                Next hop: 10.0.12.14 via fe-0/1/0.0 weight 0x1 balance 20%
                Label-switched-path lsp2
                Label operation: Push 100736
                Next hop: 10.0.12.14 via fe-0/1/0.0 weight 0x1 balance 30%
                Label-switched-path lsp3
                Label operation: Push 100752
                Next hop: 10.0.12.14 via fe-0/1/0.0 weight 0x1 balance 40%, selected
                Label-switched-path lsp4
                Label operation: Push 100784
                State: <Active Int>
                Local AS: 65432
                Age: 8:03       Metric: 4
                Task: RSVP
                Announcement bits (1): 1-Resolve tree 1
                AS path: I

user@R1> show mpls lsp statistics
Ingress LSP: 4 sessions
```

```
To              From            State     Packets            Bytes LSPname
192.168.0.1     192.168.1.1     Up          10067           845628  lsp1
192.168.0.1     192.168.1.1     Up          20026          1682184  lsp2
192.168.0.1     192.168.1.1     Up          29796          2502864  lsp3
192.168.0.1     192.168.1.1     Up          40111          3369324  lsp4
Total 4 displayed, Up 4, Down 0

Egress LSP: 1 sessions
To              From            State     Packets            Bytes LSPname
192.168.1.1     192.168.0.1     Up             NA               NA r0-r1
Total 1 displayed, Up 1, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Meaning

The sample output from ingress router **R1** shows that traffic is distributed according to the LSP bandwidth configuration, as indicated by the **Balance: xx%** field. For example, **lsp1** has 10 Mbps of bandwidth configured, as reflected in the **Balance: 10%** field.

Use the traceroute Command to Verify MPLS Labels

IN THIS SECTION

- Purpose | **2455**
- Action | **2456**
- Meaning | **2456**

Purpose

You can use the `traceroute` command to verify that packets are being sent over the LSP.

Action

To verify MPLS labels, enter the following Junos OS CLI operational mode command, where ***host-name*** is the IP address or the name of the remote host:

```
user@host> traceroute host-name
```

Sample Output 1

command-name

```
user@R1> traceroute 100.100.6.1
traceroute to 100.100.6.1 (100.100.6.1), 30 hops max, 40 byte packets
 1  10.1.12.2 (10.1.12.2)  0.861 ms  0.718 ms  0.679 ms
      MPLS Label=100048 CoS=0 TTL=1 S=1
 2  10.1.24.2 (10.1.24.2)  0.822 ms  0.731 ms  0.708 ms
      MPLS Label=100016 CoS=0 TTL=1 S=1
 3  10.1.46.2 (10.1.46.2)  0.571 ms !N  0.547 ms !N  0.532 ms !N
```

Sample Output 2

command-name

```
user@R1> traceroute 10.0.0.6
traceroute to 10.0.0.6 (10.0.0.6), 30 hops max, 40 byte packets
 1  10.1.13.2 (10.1.13.2)  0.605 ms  0.548 ms  0.503 ms
 2  10.0.0.6 (10.0.0.6)  0.761 ms  0.676 ms  0.675 ms
```

Meaning

Sample Output 1 shows that MPLS labels are used to forward packets through the network. Included in the output is a label value (**MPLS Label=100048**), the time-to-live value (**TTL=1**), and the stack bit value (**S=1**).

The **MPLS Label** field is used to identify the packet to a particular LSP. It is a 20-bit field, with a maximum value of (2^^20-1), or approximately 1,000,000.

The TTL value contains a limit on the number of hops that this MPLS packet can travel through the network (1). It is decremented at each hop, and if the TTL value drops below one, the packet is discarded.

The bottom of the stack bit value (**S=1**) indicates that is the last label in the stack and that this MPLS packet has one label associated with it. The MPLS implementation in the Junos OS supports a stacking depth of 3 on the M-series routers and up to 5 on the T-series platforms. For more information on MPLS label stacking, see RFC 3032, *MPLS Label Stack Encoding.*

MPLS labels appear in Sample Output 1 because the `traceroute` command is issued to a BGP destination where the BGP next hop for that route is the LSP egress address. The Junos OS default behavior uses LSPs for BGP traffic when the BGP next hop equals the LSP egress address.

Sample Output 2 shows that MPLS labels do not appear in the output for the `traceroute` command. If the BGP next hop does not equal the LSP egress address or the destination is an IGP route, the BGP traffic does not use the LSP. Instead of using the LSP, the BGP traffic is using the IGP (IS-IS, in this case) to reach the egress address (**R6**).

Troubleshooting GMPLS and GRE Tunnel

IN THIS SECTION

- Problem | **2457**
- Solution | **2469**

Problem

Description

The logical control channel for GMPLS must be a point-to-point link and must have some form of IP reachability. On broadcast interfaces or when there are multiple hops between control channel peers, use a GRE tunnel for the control channel. For more detailed information on GMPLS and GRE tunnels see the *Junos MPLS Applications Configuration Guide* and the *Junos User Guide*.

A tunnel PIC is *not* required to configure a GRE tunnel for the GMPLS control channel. Instead, use the software-based **gre** interface, rather than the hardware-based **gr-*fpc/pic/port*** interface.

CAUTION: Due to restrictions to the software-based **gre** interface, the GMPLS control channel is the only supported use of the software-based **gre** interface. Any other use is expressly unsupported and might cause an application failure.

The following example shows a basic **gre** interface configuration. In this case, the tunnel source is the loopback address of the local router and the destination address is the loopback destination of the remote router. Traffic that has a next hop of the tunnel destination will use the tunnel. The tunnel is not automatically used by all the traffic passing through the interface. Only traffic with the tunnel destination as the next hop uses the tunnel.

Sample Output

```
user@R1> show configuration interfaces
[...Output truncated...]
gre {
    unit 0 {
        tunnel {
            source 10.0.12.13;
            destination 10.0.12.14;
        }
        family inet {
            address 10.35.1.6/30;
        }
        family mpls;
    }
}
```

Sample Output

The following sample output for the show interfaces command shows the encapsulation type and header, the maximum speed, packets through the logical interface, the destination, and logical address.

```
user@R1> show interfaces gre
Physical interface: gre, Enabled, Physical link is Up
  Interface index: 10, SNMP ifIndex: 8
   Type: GRE, Link-level type: GRE,  MTU: Unlimited, Speed: Unlimited
  Device flags   : Present Running
  Interface flags: Point-To-Point SNMP-Traps
```

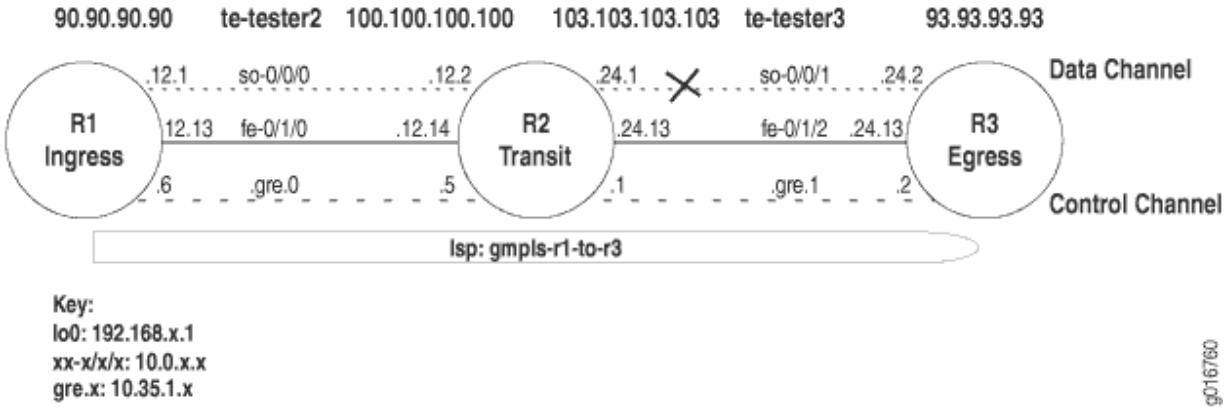
```
    Input packets : 0
    Output packets: 0

  Logical interface gre.0 (Index 70) (SNMP ifIndex 47)
    Flags: Point-To-Point SNMP-Traps 0x4000
     IP-Header 10.0.12.14:10.0.12.13:47:df:64:0000000000000000
     Encapsulation: GRE-NULL
    Input packets : 171734
    Output packets: 194560
    Protocol inet, MTU: 1476
      Flags: None
      Addresses, Flags: Is-Preferred Is-Primary
         Destination: 10.35.1.4/30, Local: 10.35.1.6, Broadcast: 10.35.1.7
    Protocol mpls, MTU: 1464
      Flags: None
```

The following are various requirements when you configure a GMPLS LSP using a GRE tunnel:

- The data channel must start and end on the same type of interface.
- The control channel can be a GRE tunnel that starts and ends on the same or different interface type.
- The GRE tunnel must be configured indirectly with the **peer-interface** *peer-name* statement at the `[edit protocol ospf]` hierarchy level.
- The GRE interface must be disabled at the `[edit protocols ospf]` and `[edit protocols rsvp]` hierarchy levels.
- Data and control channels must be defined correctly in the LMP configuration .
- It is optional to disable Constrained Shortest Path First (CSPF) with the `no-cspf` statement.

This case focuses on the incorrect configuration of the endpoints of the GRE tunnel. However, you can use a similar process and commands to diagnose other GRE tunnel problems. Figure 170 on page 2460 illustrates a network topology with MPLS tunneled through a GRE interface.

Figure 170: GMPLS Network Topology



The MPLS network topology in Figure 170 on page 2460 shows Juniper Networks routers configured with a GRE tunnel that consists of the following components:

- A strict GMPLS LSP path from the ingress router to the egress router.
- On the ingress router, CSPF disabled with the `no-cspf` statement at the [**edit protocol mpls label-switched-path *lsp-name***] hierarchy level.
- Traffic-engineering links and control channels within the `peer` statement at the [**edit protocols link-management**] hierarchy level on all routers.
- OSPF and OSPF traffic engineering configured on all routers.
- A reference to the **peer-interface** in both OSPF and RSVP on all routers.
- A switching-type problem between **R2** and **R3**.

Symptom

The LSP in the network shown in Figure 170 on page 2460 is down, as indicated by the output from the `show mpls lsp` and `show rsvp session` commands, which display very similar information. The `show mpls lsp` command shows all LSPs configured on the router, as well as all transit and egress LSPs. The `show rsvp session` command displays summary information about RSVP sessions. You can use either command to verify the state of the LSP. In this case, LSP **gmpls-r1-to-r3** is down (**Dn**).

Sample Output

```
user@R1> show mpls lsp
Ingress LSP: 1 sessions
```

```
To              From            State Rt ActivePath       P     LSPname
192.168.4.1     192.168.1.1     Dn     0 -                      gmpls-r1-to-r3
Bidir
Total 1 displayed, Up 0, Down 1

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

user@R1>  show rsvp session
Ingress RSVP: 1 sessions
To              From            State   Rt Style Labelin Labelout LSPname
192.168.4.1     192.168.1.1     Dn       0  0  -       -        - gmpls-r1-to-r3
Bidir
Total 1 displayed, Up 0, Down 1

Egress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Cause

The cause of the problem with the GMPLS LSP is the configuration of different interface types at both ends of the GMPLS data channel.

Troubleshooting Commands

The Junos OS includes commands that are useful when troubleshooting a problem. This topic provides a brief description of each command, followed by sample output, and a discussion of the output in relation to the problem.

You can use the following commands when troubleshooting a GMPLS problem:

```
user@host> show mpls lsp extensive
user@host> show rsvp session detail
user@host> show link-management peer
user@host> show link-management te-link
```

```
user@host> show configuration protocols mpls
user@host> monitor start filename
user@host> show log filename
```

Sample Output

Use the show mpls lsp extensive command on transit router R1 to display detailed information about all LSPs transiting, terminating, and configured on the router.

```
user@R1> show mpls lsp extensive
Ingress LSP: 1 sessions

192.168.4.1
  From: 192.168.1.1, State: Dn, ActiveRoute: 0, LSPname: gmpls-r1-to-r3
  Bidirectional
  ActivePath: (none)
  LoadBalance: Random
  Encoding type: SDH/SONET, Switching type: PSC-1, GPID: IPv4
  Primary   p1               State: Dn
    SmartOptimizeTimer: 180
    8 Dec 20 18:08:02 192.168.4.1: MPLS label allocation failure [3 times]
    7 Dec 20 18:07:53 Originate Call
    6 Dec 20 18:07:53 Clear Call
    5 Dec 20 18:07:53 Deselected as active
    4 Dec 20 18:06:13 Selected as active path
    3 Dec 20 18:06:13 Record Route:  100.100.100.100 93.93.93.93
    2 Dec 20 18:06:13 Up
    1 Dec 20 18:06:13 Originate Call
  Created: Wed Dec 20 18:06:12 2006
Total 1 displayed, Up 0, Down 1

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Meaning

The sample output for the `show mpls lsp extensive` command shows an error message (**MPLS label allocation failure)** in the log section of the output. This LSP event indicates that the MPLS protocol or the `family mpls` statement were not configured properly. When the LSP event is preceded by an IP address, the address is typically the router that has the MPLS configuration error. In this case, it appears that the router with the **lo0** address of **192.168.4.1** (**R3**) has an MPLS configuration error.

Sample Output

Use the show rsvp session detail command to display detailed information about RSVP sessions.

```
user@R1> show rsvp session detail
Ingress RSVP: 1 sessions

192.168.4.1
  From: 192.168.1.1,  LSPstate: Dn, ActiveRoute: 0
   LSPname: gmpls-r1-to-r3, LSPpath: Primary
  Bidirectional, Upstream label in: 21253, Upstream label out: -
  Suggested label received: -,  Suggested label sent: 21253
  Recovery label received: -, Recovery label sent: -
   Resv style: 0 - , Label in: -, Label out: -
  Time left:    -, Since: Wed Dec 20 18:07:53 2006
  Tspec: rate 0bps size 0bps peak 155.52Mbps m 20 M 1500
  Port number: sender 2 receiver 46115 protocol 0
  PATH rcvfrom: localclient
  Adspec: sent MTU 1500
  Path MTU: received 0
  PATH sentto: 10.35.1.5 (tester2) 3 pkts
   Explct route: 100.100.100.100 93.93.93.93
   Record route: <self> ...incomplete
Total 1 displayed, Up 0, Down 1

Egress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Meaning

The sample output for the `show rsvp session detail` command shows that LSP **gmpls-r1-to-r3** is down (**LSPstate: Dn**). The route record is incomplete, indicating a problem with the explicit route **100.100.100.100 93.93.93.93**. The address **100.100.100.100** is the data channel on **R2 so-0/0/0**, and the address **93.93.93.93** is the data channel on **R3**.

Sample Output

Use the show link-management peer command to display MPLS peer link information.

```
user@R1> show link-management peer
Peer name: tester2, System identifier: 48428
  State: Up, Control address:  10.35.1.5
   Control-channel                  State
   gre.0                             Active
  TE links:
   tester2

user@R2> show link-management peer
Peer name: tester2, System identifier: 48428
 State: Up, Control address: 10.35.1.6
   Control-channel                  State
   gre.0                             Active
  TE links:
   te-tester2

Peer name: tester3 , System identifier: 48429
  State: Up , Control address: 10.35.1.2
   Control-channel                  State
   gre.1                             Active
  TE links:
   te-tester3

user@R3> show link-management peer
Peer name: tester3, System identifier: 48429
  State: Up, Control address: 10.35.1.1
   Control-channel                  State
   gre.0                             Active
  TE links:
   te-tester3
```

Meaning

The sample output from all routers in the example network in Figure 170 on page 2460 for the `show link-management peer` command shows that all control channels are up and active. A detailed analysis of the output shows the following information:

- Name of the peer, **tester2** or **tester3**, which is the same on neighboring routers for ease of troubleshooting.
- Internal identifier for the peer, **48428** for **tester2** and **48429** for **tester3**. The internal identifier is a range of values from 0 through 64,000.
- The state of the peer, which can be up or down. In this case, all peers are up.
- The address to which a control channel is established, for example, **10.35.1.5.**
- The state of the control channel, which can be up, down, or active.
- The traffic-engineered links that are managed by their peer, indicating that control channel **gre.0** is managed by **tester3**.

Sample Output

Use the show link-management te-link command to display the resources used to set up Multiprotocol Label Switching (MPLS) traffic-engineered forwarding paths.

```
user@R1> show link-management te-link
TE link name:  tester2, State: Up
  Local identifier: 2005, Remote identifier: 21253, Local address: 90.90.90.90, Remote address:
100.100.100.100,
  Encoding: SDH/SONET, Switching: PSC-1, Minimum bandwidth: 155.52Mbps, Maximum bandwidth:
155.52Mbps, Total bandwidth: 155.52Mbps,
  Available bandwidth: 0bps
    Name        State Local ID  Remote ID      Bandwidth  Used   LSP-name
     so-0/0/0    Up       21253      21253      155.52Mbps   Yes   gmpls-r1-to-r3

user@R2> show link-management te-link
 TE link name:  te-tester2, State: Up
  Local identifier: 7002, Remote identifier: 22292, Local address: 100.100.100.100, Remote
address: 90.90.90.90,
  Encoding: SDH/SONET, Switching: PSC-1, Minimum bandwidth: 155.52Mbps, Maximum bandwidth:
155.52Mbps, Total bandwidth: 155.52Mbps,
  Available bandwidth: 0bps
```

```
    Name        State Local ID  Remote ID      Bandwidth  Used   LSP-name
    so-0/0/0    Up       21253      21253     155.52Mbps   Yes   gmpls-r1-to-r3
TE link name:  te-tester3, State: Up
  Local identifier: 7003, Remote identifier: 21254, Local address: 103.103.103.103, Remote
address: 93.93.93.93,
  Encoding: SDH/SONET, Switching: PSC-1, Minimum bandwidth: 155.52Mbps, Maximum bandwidth:
155.52Mbps, Total bandwidth: 155.52Mbps,
  Available bandwidth: 0bps
    Name        State Local ID  Remote ID      Bandwidth  Used   LSP-name
    so-0/0/1    Up       21252      21252     155.52Mbps   Yes  gmpls-r1-to-r3

user@R3> show link-management te-link
 TE link name:  te-tester3, State: Up
  Local identifier: 7003, Remote identifier: 21254,  Local address: 93.93.93.93,
Remote address: 103.103.103.103,
  Encoding: SDH/SONET, Switching: PSC-1, Minimum bandwidth: 0bps, Maximum bandwidth: 0bps, Total
bandwidth: 0bps,
  Available bandwidth: 0bps
    Name        State Local ID  Remote ID      Bandwidth Used  LSP-name
    so-0/0/1    Dn       21252      21252     155.52Mbps    No
```

Meaning

The sample output for the `show link-management te-link` command issued on the three routers in the network in Figure 170 on page 2460 shows the resources allocated to the traffic-engineered links **te-tester2** and **te-tester3**. The resources are the SONET interfaces **so-0/0/0** and **so-0/0/1.** On **R1** and **R2,** the SONET interfaces are used for the LSP **gmpls-r1-to-r3**, as indicated by **Yes** in the **Used** field.However, the SONET interface **so-0/0/1** on **R3** is down (**Dn**) and is not used for the LSP (**Used No**). Further investigation is required to discover why the SONET interface on **R3** is down.

Sample Outut

Use the show log *filename* command to display the contents of the specified log file. In this case, the log file rsvp.log is configured at the [edit protocols rsvp traceoptions] hierarchy level. When the log file is configured, you must issue the monitor start *filename* command to begin logging messages to the file.

```
user@R1> show configuration protocols rsvp
traceoptions {
    file rsvp.log size 3m world-readable;
    flag state detail;
    flag error detail;
```

```
    flag packets detail;
}

user@R1> monitor start rsvp.log
```

NOTE: The **find Error** option entered after the pipe (**|**) searches the output for an instance of the term *Error*.

Sample Output

```
user@R3>
show log rsvp.log | find Error
Dec 28 17:23:32  Error  Len 20 Session preempted flag 0 by 192.168.4.1 TE-link 103.103.103.103
[...Output truncated...]
Dec 28 17:23:32 RSVP new resv state,session 192.168.4.1(port/tunnel ID 46115 Ext-ID
192.168.1.1)Proto 0
Dec 28 17:23:32         RSVP-LMP reset LMP request for gmpls-r1-to-r3
Dec 28 17:23:32         RSVP->LMP request -  resource for LSP gmpls-r1-to-r3
Dec 28 17:23:32         LMP->RSVP resource request gmpls-r1-to-r3 failed cannot find resource
encoding type SDH/SONET remote label 21252 bandwidth bw[0
Dec 28 17:23:32         RSVP-LMP reset LMP request for gmpls-r1-to-r3
Dec 28 17:23:32 RSVP originate PathErr 192.168.4.1->192.168.2.1 MPLS label allocation failure
LSP gmpls-r1-to-r3(2/46115)
Dec 28 17:23:32 RSVP send PathErr 192.168.4.1->192.168.2.1 Len=196 tester3
Dec 28 17:23:32   Session7 Len 16 192.168.4.1(port/tunnel ID 46115 Ext-ID 192.168.1.1) Proto 0
Dec 28 17:23:32   Hop      Len 20 192.168.4.1/0x086e4770 TE-link 103.103.103.103
Dec 28 17:23:32   Error    Len 20 MPLS label allocation failure flag 0 by 192.168.4.1 TE-link
103.103.103.103
Dec 28 17:23:32   Sender7  Len 12 192.168.1.1(port/lsp ID  2)
Dec 28 17:23:32   Tspec    Len 36 rate 0bps size 0bps peak 155.52Mbps m 20 M 1500
Dec 28 17:23:32   ADspec   Len 48 MTU 1500
Dec 28 17:23:32   RecRoute Len 20  103.103.103.103 90.90.90.90
Dec 28 17:23:32   SuggLabel Len  8  21252
Dec 28 17:23:32   UpstrLabel Len  8  21252
```

Meaning

The sample output from the egress router **R3** for the `show log rsvp.log` command is a snippet taken from the log file. The snippet shows a Link Management Protocol (LMP) resource request for the LSP **gmpls-r1-to-r3.** The request has problems with the encoding type (SDH/SONET), indicating a possible error with the SONET interface connecting **R2** and **R3**. Further investigation of the configuration of the LMP on **R2** and **R3** is required.

Sample Output

Use the show configuration *statement-path* command to display a specific configuration hierarchy; in this instance, link-management.

```
user@R2> show configuration protocols link-management
te-link te-tester2 {
    local-address 100.100.100.100;
    remote-address 90.90.90.90;
    remote-id 22292;
    interface so-0/0/0 {
        local-address 100.100.100.100;
        remote-address 90.90.90.90;
        remote-id 21253;
    }
}
te-link te-tester3  {
    local-address 103.103.103.103;
    remote-address 93.93.93.93;
    remote-id 21254;
     interface so-0/0/1  {
        local-address 103.103.103.103;
        remote-address 93.93.93.93;
        remote-id 21252;
    }
}
peer tester2 {
    address 10.35.1.6;
    control-channel gre.0;
    te-link te-tester2;
}
peer tester3 {
    address 10.35.1.2;
    control-channel gre.1;
```

```
    te-link te-tester3;
}

user@R3> show configuration protocols link-management
te-link te-tester3  {
    local-address 93.93.93.93;
    remote-address 103.103.103.103;
    remote-id 21254;
    }
     interface at-0/3/1 {
        local-address 93.93.93.93;
        remote-address 103.103.103.103;
        remote-id 21252;
    }
}
peer tester3 {
    address 10.35.1.1;
    control-channel gre.0;
    te-link te-tester3;
}
```

Meaning

The sample output from transit router **R2** and ingress router **R3** for the `show configuration protocols link-management` command shows that the interface type on the two routers is different. The resource allocated to **te-tester3** on transit router **R2** is a SONET interface, while the resource allocated to **te-tester3** on egress router **R3** is an ATM interface. The interface type on each end of the data or control channels must be of the same type. In this case, both ends should be SONET or ATM.

Solution

Solution

The solution to the problem of different interface or encapsulation types at either end of the GMPLS LSP is to make sure that the interface type is the same at both ends. In this case, the ATM interface was deleted from the link-management configuration on **R3**, and a SONET interface was configured instead.

The following commands illustrate the correct configuration and commands to verify that the GMPLS LSP is up and using the data channel:

```
user@R3> show configuration protocols link-management
user@R3> show mpls lsp
user@R3> show link-management te-link
```

Sample Output

```
user@R3> show configuration protocols link-management
te-link te-tester3 {
    local-address 93.93.93.93;
    remote-address 103.103.103.103;
    remote-id 21254;
     interface so-0/0/1 { # SONET interface replaces the incorrect ATM interface
        local-address 93.93.93.93;
        remote-address 103.103.103.103;
        remote-id 21252;
    }
}
peer tester3 {
    address 10.35.1.1;
    control-channel gre.0;
    te-link te-tester3;
}

user@R3> show mpls lsp
Ingress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
Egress LSP: 1 sessions
To              From            State   Rt Style Labelin Labelout LSPname
192.168.4.1     192.168.1.1     Up       0  1 FF   21252        - gmpls-r1-to-r3
Bidir
Total 1 displayed, Up 1, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

user@R3> show link-management te-link
 TE link name: te-tester3, State: Up
```

```
  Local identifier: 7003, Remote identifier: 21254, Local address: 93.93.93.93, Remote address:
103.103.103.103,
  Encoding: SDH/SONET, Switching: PSC-1, Minimum bandwidth: 155.52Mbps, Maximum bandwidth:
155.52Mbps, Total bandwidth: 155.52Mbps,
  Available bandwidth: 0bps
    Name        State Local ID  Remote ID      Bandwidth Used  LSP-name
     so-0/0/1    Up       21252      21252     155.52Mbps  Yes  gmpls-r1-to-r3
```

Meaning

The sample output for the `show protocols link-management`, `show mpls lsp`, and `show link-management te-link` commands from ingress router **R3** show that the problem is solved. LMP is correctly configured, and the LSP **gmpls-r1-to-r3** is up and using the data channel **so-0/0/1**.

Conclusion

In conclusion, both ends of a GMPLS data channel must be the same encapsulation or interface type. This case illustrates the correct configuration of the data channel. The principles are the same for the control channel.

Router Configurations

Output that shows the configurations of the ingress router in the network. The **no-more** option entered after the pipe **(|)** prevents the output from being paginated if the output is longer than the length of the terminal screen.

Sample Output

The following sample output is for ingress router R1:

```
user@R1>  show configuration | no-more
[...Output truncated...]
interfaces {
    so-0/0/0 {
        unit 0 {
            family inet {
                address 10.0.12.1/32 {
                    destination 10.0.12.2;
                }
            }
            family mpls;
```

```
        }
    }
    fe-0/1/0 {
        unit 0 {
            family inet {
                address 10.0.12.13/30;
            }
            family mpls;
        }
    }
    fxp0 {
        unit 0 {
            family inet {
                address 192.168.70.143/21;
            }
        }
    }
    gre {
        unit 0 {
            tunnel {
                source 10.0.12.13;
                destination 10.0.12.14;
            }
            family inet {
                address 10.35.1.6/30;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 192.168.1.1/32;
            }
        }
    }
}
routing-options {
    static {
        /* corporate and alpha net */
        route 172.16.0.0/12 {
            next-hop 192.168.71.254;
            retain;
```

```
            no-readvertise;
        }
        /* old lab nets */
        route 192.168.0.0/16 {
            next-hop 192.168.71.254;
            retain;
            no-readvertise;
        }
        route 0.0.0.0/0 {
            discard;
            retain;
            no-readvertise;
        }
    }
    router-id 192.168.1.1;
    autonomous-system 65432;
}
protocols {
    rsvp {
        traceoptions {
            file rsvp.log size 3m world-readable;
            flag state detail;
            flag error detail;
            flag packets detail;
        }
        interface fxp0.0 {
            disable;
        }
        interface all;
        interface lo0.0;
        interface gre.0 {
            disable;
        }
        peer-interface tester2;
    }
    mpls {
        label-switched-path gmpls-r1-to-r3 {
            from 192.168.1.1;
            to 192.168.4.1;
            lsp-attributes {
                switching-type psc-1;
                encoding-type sonet-sdh;
            }
```

```
            no-cspf;
            primary p1;
        }
        path p1 {
            100.100.100.100 strict;
            93.93.93.93 strict;
        }
        interface all;
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface lo0.0;
            interface fe-0/1/0.0;
            interface fxp0.0 {
                disable;
            }
            interface gre.0 {
                disable;
            }
            peer-interface tester2;
        }
    }
    link-management {
        te-link tester2 {
            local-address 90.90.90.90;
            remote-address 100.100.100.100;
            remote-id 21253;
            interface so-0/0/0 {
                local-address 90.90.90.90;
                remote-address 100.100.100.100;
                remote-id 21253;
            }
        }
        peer tester2 {
            address 10.35.1.5;
            control-channel gre.0;
            te-link tester2;
        }
    }
}
```

Sample Output

The following sample output is for transit router R2:

```
user@R2>show configuration | no-more
[...Output truncated...]
interfaces {
    so-0/0/0 {
        unit 0 {
            family inet {
                address 10.0.12.2/32 {
                    destination 10.0.12.1;
                }
            }
            family mpls;
        }
    }
    so-0/0/1 {
        unit 0 {
            family inet {
                address 10.0.24.1/32 {
                    destination 10.0.24.2;
                }
            }
            family mpls;
        }
    }
    fe-0/1/0 {
        unit 0 {
            family inet {
                address 10.0.12.14/30;
            }
            family mpls;
        }
    }
    fe-0/1/2 {
        unit 0 {
            family inet {
                address 10.0.24.13/30;
            }
            family mpls;
        }
```

```
    }
    fxp0 {
        unit 0 {
            family inet {
                address 192.168.70.144/21;
            }
        }
    }
    gre {
        unit 0 {
            tunnel {
                source 10.0.12.14;
                destination 10.0.12.13;
            }
            family inet {
                address 10.35.1.5/30;
            }
            family mpls;
        }
        unit 1 {
            tunnel {
                source 10.0.24.13;
                destination 10.0.24.14;
            }
            family inet {
                address 10.35.1.1/30;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 192.168.2.1/32;
            }
        }
    }
}
routing-options {
    static {
        route 172.16.0.0/12 {
            next-hop 192.168.71.254;
            retain;
```

```
            no-readvertise;
        }
        route 192.168.0.0/16 {
            next-hop 192.168.71.254;
            retain;
            no-readvertise;
        }
        route 0.0.0.0/0 {
            discard;
            retain;
            no-readvertise;
        }
    }
    router-id 192.168.2.1;
    autonomous-system 65432;
}
protocols {
    rsvp {
        traceoptions {
            file rsvp.log size 3m world-readable;
            flag packets detail;
            flag state detail;
            flag error detail;
        }
        interface fxp0.0;
        interface lo0.0;
        interface all;
        interface gre.0 {
            disable;
        }
        peer-interface tester2;
        peer-interface tester3;
    }
    mpls {
        interface all;
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface lo0.0;
            interface fxp0.0 {
                disable;
            }
```

```
            interface gre.0 {
                disable;
            }
            interface fe-0/1/0.0;
            interface fe-0/1/2.0;
            interface gre.1 {
                disable;
            }
            peer-interface tester2;
            peer-interface tester3;
        }
    }
    link-management {
        te-link te-tester2 {
            local-address 100.100.100.100;
            remote-address 90.90.90.90;
            remote-id 22292;
            interface so-0/0/0 {
                local-address 100.100.100.100;
                remote-address 90.90.90.90;
                remote-id 21253;
            }
        }
        te-link te-tester3 {
            local-address 103.103.103.103;
            remote-address 93.93.93.93;
            remote-id 21254;
            interface so-0/0/1 {
                local-address 103.103.103.103;
                remote-address 93.93.93.93;
                remote-id 21252;
            }
        }
        peer tester2 {
            address 10.35.1.6;
            control-channel gre.0;
            te-link te-tester2;
        }
        peer tester3 {
            address 10.35.1.2;
            control-channel gre.1;
            te-link te-tester3;
        }
```

```
    }
}
```

Sample Output

The following sample output is for egress router R3:

```
user@R3> show configuration | no-more
[...Output truncated...]
interfaces {
    so-0/0/1 {
        unit 0 {
            family inet {
                address 10.0.24.2/32;
            }
            family mpls;
        }
    }
    fe-0/1/2 {
        unit 0 {
            family inet {
                address 10.0.24.14/30;
            }
            family mpls;
        }
    }
    fxp0 {
        unit 0 {
            family inet {
                address 192.168.70.146/21;
            }
        }
    }
    gre {
        unit 0 {
            tunnel {
                source 10.0.24.14;
                destination 10.0.24.13;
            }
            family inet {
                address 10.35.1.2/30;
```

```
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 192.168.4.1/32;
            }
        }
    }
}
routing-options {
    static {
        route 172.16.0.0/12 {
            next-hop 192.168.71.254;
            retain;
            no-readvertise;
        }
        route 192.168.0.0/16 {
            next-hop 192.168.71.254;
            retain;
            no-readvertise;
        }
        route 0.0.0.0/0 {
            discard;
            retain;
            no-readvertise;
        }
    }
    router-id 192.168.4.1;
    autonomous-system 65432;
}
protocols {
    rsvp {
        traceoptions {
            file rsvp.log size 3m world-readable;
            flag packets detail;
            flag error;
            flag state;
            flag lmp;
        }
        interface fxp0.0 {
```

```
            disable;
        }
        interface all;
        interface lo0.0;
        interface gre.0 {
            disable;
        }
        peer-interface tester3;
    }
    mpls {
        interface all;
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface fxp0.0 {
                disable;
            }
            interface fe-0/1/2.0;
            interface gre.0 {
                disable;
            }
            interface lo0.0;
            peer-interface tester3;
        }
    }
    link-management {
        te-link te-tester3 {
            local-address 93.93.93.93;
            remote-address 103.103.103.103;
            remote-id 21254;
            interface so-0/0/1 {
                local-address 93.93.93.93;
                remote-address 103.103.103.103;
                remote-id 21252;
            }
        }
        peer tester3 {
            address 10.35.1.1;
            control-channel gre.0;
            te-link te-tester3;
        }
```

```
    }
}
```

Determining LSP Status

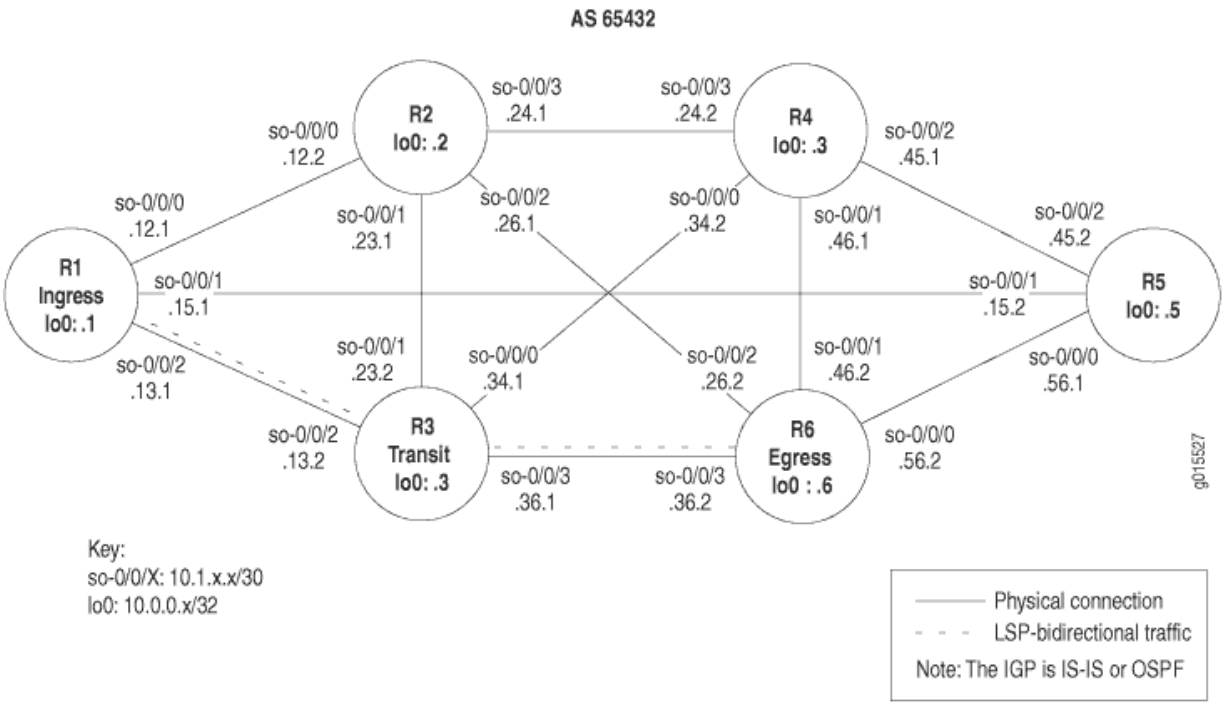
IN THIS SECTION

- Check the Status of the LSP | **2483**
- Display Extensive Status About the LSP | **2484**

Display detailed information about Resource Reservation Protocol (RSVP) objects and the label-switched path (LSP) history to pinpoint a problem with the LSP.

Figure 171 on page 2482 illustrates the network topology used in this topic.

Figure 171: MPLS Network Topology



To determine the LSP state, follow these steps:

Check the Status of the LSP

IN THIS SECTION

- Purpose | 2483
- Action | 2483
- Meaning | 2484

Purpose

Display the status of the label-switched pathe (LSP).

Action

To determine the LSP status, on the ingress router, enter the following Junos OS command-line interface (CLI) operational mode command:

```
user@host>              show mpls lsp
```

Sample Output

command-name

```
user@R1> show mpls lsp
Ingress LSP: 1 sessions
To              From            State Rt ActivePath       P     LSPname
10.0.0.6        10.0.0.1        Up     1                  *     R1-to-R6
Total 1 displayed, Up 1, Down 0

Egress LSP: 1 sessions
To              From            State Rt  Style   Labelin  Labelout LSPname
10.0.0.1        10.0.0.6         Up     0 1  FF        3         -  R6-to-R1
Total 1 displayed, Up 1, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Meaning

The sample output is from the ingress router (**R1**), and shows ingress, egress, and transit LSP information. Ingress information is for the sessions that originate from this router, egress information is for sessions that terminate on this router, and transit information is for sessions that transit through this router.

There is one ingress route from **R1** (**10.0.0.1**) to **R6** (**10.0.0.6**). This route is currently up, and is an active route installed in the routing table (**Rt**). The LSP **R1-to-R6** is the primary path (**P**) as opposed to the secondary path, and is indicated by an asterisk (*****). The route to **R6** does not contain a named path (**ActivePath**).

There is one egress LSP from **R6** to **R1**. The **State** is up, with no routes installed in the routing table. RSVP reservation style (**Style**) consists of two parts. The first is the number of active reservations (**1**). The second is the reservation style, which is **FF** (fixed filter). The reservation style can be **FF**, **SE** (shared explicit), or **WF** (wildcard filter). There are three incoming labels (**Labelin**) and no labels going out (**Labelout**) for this LSP.

There are no transit LSPs.

For more information on checking the LSP state, see Checklist for Working with the Layered MPLS Troubleshooting Model.

Display Extensive Status About the LSP

IN THIS SECTION

- Purpose | **2484**
- Action | **2485**
- Meaning | **2487**

Purpose

Display extensive information about LSPs, including all past state history and the reasons why an LSP might have failed.

Action

To display extensive information about LSPs, on the ingress router, enter the following Junos OS CLI operational mode command:

```
user@host>                          show mpls lsp extensive
```

Sample Output

command-name

```
user@R1> show mpls lsp extensive
Ingress LSP: 1 sessions

10.0.0.6
  From: 10.0.0.1, State:  Up , ActiveRoute:  1 , LSPname:  R1-to-R6
  ActivePath:   (primary)
  LoadBalance:  Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
  *Primary                    State:  Up
     Computed ERO  (S [L] denotes strict [loose] hops): (CSPF metric: 20)
 10.1.13.2 S 10.1.36.2 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt):
          10.1.13.2 10.1.36.2
   91 Aug 17 12:22:52 Selected as active path
   90 Aug 17 12:22:52 Record Route:  10.1.13.2 10.1.36.2
   89 Aug 17 12:22:52 Up
   88 Aug 17 12:22:52 Originate Call
   87 Aug 17 12:22:52 CSPF: computation result accepted
   86 Aug 17 12:22:23 CSPF failed: no route toward 10.0.0.6[13920 times]
   85 Aug 12 19:12:51 Clear Call
   84 Aug 12 19:12:50 10.1.56.2: MPLS label allocation failure
   83 Aug 12 19:12:47 Deselected as active
   82 Aug 12 19:12:47 10.1.56.2: MPLS label allocation failure
   81 Aug 12 19:12:47 ResvTear received
   80 Aug 12 19:12:47 Down
   79 Aug 12 19:12:31 10.1.56.2: MPLS label allocation failure[4 times]
   78 Aug 12 19:09:58 Selected as active path
   77 Aug 12 19:09:58 Record Route:  10.1.15.2 10.1.56.2
   76 Aug 12 19:09:58 Up
```

```
   75 Aug 12 19:09:57 Originate Call
   74 Aug 12 19:09:57 CSPF: computation result accepted
   73 Aug 12 19:09:29 CSPF failed: no route toward 10.0.0.6[11 times]
   72 Aug 12 19:04:36 Clear Call
   71 Aug 12 19:04:23 Deselected as active
   70 Aug 12 19:04:23 ResvTear received
   69 Aug 12 19:04:23 Down
   68 Aug 12 19:04:23 CSPF failed: no route toward 10.0.0.6
   67 Aug 12 19:04:23 10.1.15.2: Session preempted
   66 Aug 12 16:45:35 Record Route:  10.1.15.2 10.1.56.2
   65 Aug 12 16:45:35 Up
   64 Aug 12 16:45:35 Clear Call
   63 Aug 12 16:45:35 CSPF: computation result accepted
   62 Aug 12 16:45:35 ResvTear received
   61 Aug 12 16:45:35 Down
   60 Aug 12 16:45:35 10.1.13.2: Session preempted
   59 Aug 12 14:50:52 Selected as active path
   58 Aug 12 14:50:52 Record Route:  10.1.13.2 10.1.36.2
   57 Aug 12 14:50:52 Up
   56 Aug 12 14:50:52 Originate Call
   55 Aug 12 14:50:52 CSPF: computation result accepted
   54 Aug 12 14:50:23 CSPF failed: no route toward 10.0.0.6[7 times]
   53 Aug 12 14:47:22 Deselected as active
   52 Aug 12 14:47:22 CSPF failed: no route toward 10.0.0.6
   51 Aug 12 14:47:22 Clear Call
   50 Aug 12 14:47:22 CSPF: link down/deleted 10.1.12.1(R1.00/10.0.0.1)-
>10.1.12.2(R2.00/10.0.0.2)
   49 Aug 12 14:47:22 CSPF: link down/deleted 10.1.15.1(R1.00/10.0.0.1)-
>10.1.15.2(R5.00/10.0.0.5)
   48 Aug 12 14:47:22 10.1.15.1: MPLS label allocation failure
   47 Aug 12 14:47:22 Clear Call
   46 Aug 12 14:47:22 CSPF: computation result accepted
   45 Aug 12 14:47:22 10.1.12.1: MPLS label allocation failure
   44 Aug 12 14:47:22 MPLS label allocation failure
   43 Aug 12 14:47:22 Down
   42 Jul 23 11:27:21 Selected as active path
   Created: Sat Jul 10  18:18:44 2004
Total 1 displayed, Up 1, Down 0

Egress LSP: 1 sessions

10.0.0.1
  From: 10.0.0.6,  LSPstate: Up , ActiveRoute: 0
```

```
  LSPname:  R6-to-R1 , LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: -
   Resv style: 1 FF ,  Label in: 3 ,  Label out: -
  Time left:  141, Since: Tue Aug 17 12:23:14 2004
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 39024 protocol 0
  PATH rcvfrom: 10.1.15.2 (so-0/0/1.0) 130 pkts
  Adspec: received MTU 1500
  PATH sentto: localclient
  RESV rcvfrom: localclient
  Record route: 10.1.56.2 10.1.15.2 <self>
Total 1 displayed, Up 1, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Meaning

The sample output is from the ingress router (**R1**), and shows ingress, egress, and transit LSP information in detail, including all past state history and the reasons why an LSP failed. Ingress information is for sessions that originate from this router, egress information is for sessions that terminate on this router, and transit information is for sessions that transit through this router.

There is one ingress route from **R1** (**10.0.0.1**) to **R6** (**10.0.0.6**). This route is currently up (**State**), with one route actively using the LSP, **R1-to-R6**. The LSP active path is the primary path. Even if the LSP does not contain a **primary** or **secondary** keyword, the router still treats the LSP as a primary LSP, indicating that if the LSP fails, the router will attempt to signal inactive LSPs at 30-second intervals, by default.

Load balancing is **Random**, which is the default, indicating that when selecting the physical path for an LSP, the router randomly selects among equal-cost paths that have an equal hop count. Other options that you can configure are **Least-fill** and **Most-fill**. **Least-fill** places the LSP over the least utilized link of the equal-cost paths with equal hop count. **Most-fill** places the LSP over the most utilized link of the equal-cost paths sharing an equal hop count. Utilization is based on the percentage of available bandwidth.

The **Encoding type** field shows Generalized MPLS (GMPLS) signaling parameters (**Packet)**, indicating IPv4. The **Switching type** is **Packet**, and the Generalized Payload Identifier (**GPID**) is IPv4.

The primary path is the active path, as indicated by an asterisk (*). The state of the LSP is **Up**.

The Explicit Route Object (**ERO**) includes the Constrained Shortest Path First (CSPF) cost (**20**) for the physical path that the LSP follows. The presence of the CSPF metric indicates that this is a CSPF LSP. The absence of the CSPF metric indicates a no-CSPF LSP.

The field **10.1.13.2 S** indicates the actual ERO. The RSVP signaling messages went to **10.1.13.2** strictly (as a next hop) and finished at **10.1.36.2** strictly. All ERO addresses are strict hops when the LSP is a CSPF LSP. Loose hops can only display in a no-CSPF LSP.

The received Record Route Object (**RRO**) has the following protection flags:

- **0x01**—Local protection available. The link downstream of this node is protected by a local repair mechanism. This flag can only be set if the Local protection flag was set in the SESSION_ATTRIBUTE object of the corresponding path message.
- **0x02**—Local protection in use. A local repair mechanism is in use to maintain this tunnel (usually because of an outage of the link it was routed over previously).
- **0x04**— Bandwidth protection. The downstream router has a backup path providing the same bandwidth guarantee as the protected LSP for the protected section.
- **0x08**—Node protection. The downstream router has a backup path providing protection against link and node failure on the corresponding path section. If the downstream router can set up only a link-protection backup path, the “Local protection available” bit is set but the “Node protection” bit is cleared.
- **0x10**—Preemption pending. The preempting node sets this flag if a pending preemption is in progress for the traffic engineered LSP. This indicates to the ingress label edge router (LER) of this LSP that it should be rerouted.

For more information on protection flags, see the *Junos Routing Protocols and Policies Command Reference.*

The field **10.1.13.2.10.1.36.2** is the actual received record route (**RRO**). Note that the addresses in the **RRO** field match those in the **ERO** field. This is the normal case for CSPF LSPs. If the RRO and ERO addresses do not match for a CSPF LSP, the LSP has to reroute or detour.

The lines numbered 91 through 42 contain the 49 most recent entries to the history log. Each line is time stamped. The most recent entries have the largest log history number and are at the top of the log, indicating that line 91 is the most recent history log entry. When you read the log, start with the oldest entry (**42**) to the most recent (**91**).

The history log was started on July 10, and displays the following sequence of activities: an LSP was selected as active, was found to be down, MPLS label allocation failed several times, was deleted several times, was preempted because of an ResvTear, was deselected as active, and was cleared. In the end, the router computed a CSPF ERO, signaled the call, the LSP came up with the listed RRO (line 90), and was listed as active.

For more information on error messages, see the *Junos MPLS Network Operations Guide Log Reference.*

The total number of ingress LSPs displayed is **1**, with **1** up and **0** down. The number in the **Up** field plus the number in the **Down** field should equal the total.

There is one egress LSP session from **R6** to **R1**. The **State** is up, with no routes installed in the routing table. RSVP reservation style (**Style**) consists of two parts. The first is the number of active reservations (**1**). The second is the reservation style, which is **FF** (fixed filter). The reservation style can be **FF**, **SE** (shared explicit), or **WF** (wildcard filter). There are three incoming labels (**Labelin**) and no labels going out (**Labelout**) for this LSP.

There are no transit LSPs.

For more information on checking the LSP state, see Checklist for Working with the Layered MPLS Troubleshooting Model.

Checking That RSVP Path Messages Are Sent and Received

IN THIS SECTION

- Purpose | **2489**
- Action | **2489**
- Meaning | **2491**

Purpose

The presence or absence of various RSVP messages can help determine if there is a problem with Multiprotocol Label Switching (MPLS) in your network. For example, if path messages occur in the output without Resv messages, it might indicate that label-switched paths (LSPs) are not being created.

Action

To check that RSVP Path messages are sent and received, enter the following Junos OS command-line interface (CLI) operational mode command:

```
user@host> show rsvp statistics
```

Sample Output

command-name

```
user@R1> show rsvp statistics
  PacketType              Total                 Last 5 seconds
                    Sent      Received        Sent      Received
 Path             114523         80185           1             0
  PathErr              5            10           0             0
   PathTear           12             6           0             0
  Resv FF           80515        111476           0             0
  Resv WF               0             0           0             0
  Resv SE               0             0           0             0
  ResvErr               0             0           0             0
   ResvTear             0             5           0             0
  ResvConf              0             0           0             0
  Ack                   0             0           0             0
  SRefresh              0             0           0             0
  Hello            915851        915881           0             0
  EndtoEnd RSVP         0             0          0             0

  Errors                          Total           Last 5 seconds
  Rcv pkt bad length                0                        0
  Rcv pkt unknown type              0                        0
  Rcv pkt bad version               0                        0
  Rcv pkt auth fail                 0                        0
  Rcv pkt bad checksum              0                        0
  Rcv pkt bad format                0                        0
  Memory allocation fail            0                        0
  No path information               0                        0
  Resv style conflict               0                        0
  Port conflict                     0                        0
  Resv no interface                 0                        0
  PathErr to client                15                        0
  ResvErr to client                 0                        0
  Path timeout                      0                        0
  Resv timeout                      0                        0
  Message out-of-order              0                        0
  Unknown ack msg                   0                        0
  Recv nack                         0                        0
```

```
    Recv duplicated msg-id            0                        0
    No TE-link to recv Hop            0                        0
```

Meaning

The sample output shows RSVP messages sent and received. The total number of RSVP Path messages is 11,4532 sent and 80,185 received. Within the last 5 seconds, no messages have been sent or received.

A total of 5 **PathErr** messages were sent and 10 received. When path errors occur (usually because of parameter problems in a path message), the router sends a unicast PathErr message to the sender that issued the path message. In this case, **R1** sent at least 10 path messages with an error, as indicated by the 10 PathErr messages that **R1** has received. The downstream router sent **R1** five path messages with an error, as indicated by the five PathErr messages that **R1** has sent. PathErr messages transmit in the opposite direction to path messages.

A total of 12 **PathTear** messages were sent and 6 received, none in the last 5 seconds. In contrast to PathErr messages, PathTear messages travel in the same direction as path messages. Since path messages are both sent and received, PathTear messages are also sent and received. However, if only path messages are sent, then only the PathTear messages that are sent appear in the output.

A total of 80,515 reservation (**Resv**) messages with the fixed filter (**FF**) reservation style were sent and 111,476 received, none in the last 5 seconds. An **FF** reservation style indicates that within each session, each receiver establishes its own reservation with each upstream sender, and that all selected senders are listed. No messages for the wildcard filter (**WF**) or shared explicit (**SE**) reservation styles are sent or received. For more information on RSVP reservation styles, see the *Junos MPLS Applications Configuration Guide*.

Other RSVP message types are not sent or received. For information on the ResvErr, ResvTear, and Resvconf message types, see the *Junos MPLS Applications Configuration Guide*.

Ack and summary refresh (SRefresh) messages do not appear in the output. Ack and summary refresh messages are defined in RFC 2961 and are part of the RSVP extensions. Ack messages are used to reduce the amount of RSVP control traffic in the network.

A total of 915,851 hello messages were sent and 915,881 received, with none transmitted or received in the last 5 seconds. The RSVP hello interval is 9 seconds. If more than one hello message is sent or received in the last 5 seconds, it implies that more than one interface supports RSVP.

EndtoEnd RSVP messages are legacy RSVP messages that are not used for RSVP traffic engineering. These counters increment only when RSVP forwards legacy RSVP messages issued by a virtual private network (VPN) customer for transit across the backbone to the other site(s) in the VPN. They are called end-to-end messages because they are intended for the opposite side of the network and only have meaning at the two ends of the provider network.

The **Errors** section of the output shows statistics about RSVP packets with errors. A total of 15 **PathErr to client** packets were sent to the Routing Engine. The total combines the sent and received **PathErr** packets.

PART 11

Configuration Statements and Operational Commands

- Junos CLI Reference Overview | **2494**

Junos CLI Reference Overview

We've consolidated all Junos CLI commands and configuration statements in one place. Read this guide to learn about the syntax and options that make up the statements and commands. Also understand the contexts in which you'll use these CLI elements in your network configurations and operations.

- Junos CLI Reference

Click the links to access Junos OS and Junos OS Evolved configuration statement and command summary topics.

- Configuration Statements
- Operational Commands