JUNOS® Software

Feature Guide

Release 10.0

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Published: 2009-10-09

This product includes the Envoy SNMP Engine, developed by Epilogue Technology, an Integrated Systems Company. Copyright © 1986-1997, Epilogue Technology Corporation. All rights reserved. This program and its documentation were developed at private expense, and no part of them is in the public domain.

This product includes memory allocation software developed by Mark Moraes, copyright © 1988, 1989, 1993, University of Toronto.

This product includes FreeBSD software developed by the University of California, Berkeley, and its contributors. All of the documentation and software included in the 4.4BSD and 4.4BSD-Lite Releases is copyrighted by the Regents of the University of California. Copyright © 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994. The Regents of the University of California. All rights reserved.

GateD software copyright © 1995, the Regents of the University. All rights reserved. Gate Daemon was originated and developed through release 3.0 by Cornell University and its collaborators. Gated is based on Kirton's EGP, UC Berkeley's routing daemon (routed), and DCN's HELLO routing protocol. Development of Gated has been supported in part by the National Science Foundation. Portions of the GateD software copyright © 1988, Regents of the University of California. All rights reserved. Portions of the GateD software copyright © 1991, D. L. S. Associates.

This product includes software developed by Maker Communications, Inc., copyright © 1996, 1997, Maker Communications, Inc.

Juniper Networks, the Juniper Networks logo, JUNOS, NetScreen, ScreenOS, and Steel-Belted Radius are registered trademarks of Juniper Networks, Inc. in the United States and other countries. JUNOSe is a trademark of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, 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.

Products made or sold by Juniper Networks or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

JUNOS® Software Feature Guide,
Release 10.0
Copyright © 2009, Juniper Networks, Inc.
All rights reserved. Printed in USA.

Writing: Roy Spencer, Fawn Damitio, Ines Salazar, Richard Hendricks, and Walter Goralski
Editing: Nancy Kurahashi, Dawn Spencer
Illustration: Faith Bradford, Fawn Damitio, Nathaniel Woodward, Richard Hendricks, and Dawn Spencer
Cover Design: Edmonds Design

Revision History
October 2009—R1 JUNOS 10.0

The information in this document is current as of the date listed in the revision history.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. The JUNOS Software 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

READ THIS END USER LICENSE AGREEMENT ("AGREEMENT") BEFORE DOWNLOADING, INSTALLING, OR USING THE SOFTWARE. BY DOWNLOADING, INSTALLING, OR USING THE SOFTWARE OR OTHERWISE EXPRESSING YOUR AGREEMENT TO THE TERMS CONTAINED HEREIN, YOU (AS CUSTOMER OR IF YOU ARE NOT THE CUSTOMER, AS A REPRESENTATIVE/AGENT AUTHORIZED TO BIND THE CUSTOMER) CONSENT TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT OR CANNOT AGREE TO THE TERMS CONTAINED HEREIN, THEN (A) DO NOT DOWNLOAD, INSTALL, OR USE THE SOFTWARE, AND (B) YOU MAY CONTACT JUNIPER NETWORKS REGARDING LICENSE TERMS.

1. **The Parties.** The parties to this Agreement are (i) Juniper Networks, Inc. (if the Customer's principal office is located in the Americas) or Juniper Networks (Cayman) Limited (if the Customer's principal office is located outside the Americas) (such applicable entity being referred to herein as "Juniper"), and (ii) the person or organization that originally purchased from Juniper or an authorized Juniper reseller the applicable license(s) for use of the Software ("Customer") (collectively, the "Parties").

2. **The Software.** In this Agreement, "Software" means the program modules and features of the Juniper or Juniper-supplied software, for which Customer has paid the applicable license or support fees to Juniper or an authorized Juniper reseller, or which was embedded by Juniper in equipment which Customer purchased from Juniper or an authorized Juniper reseller. "Software" also includes updates, upgrades and new releases of such software. "Embedded Software" means Software which Juniper has embedded in or loaded onto the Juniper equipment and any updates, upgrades, additions or replacements which are subsequently embedded in or loaded onto the equipment.

3. **License Grant.** Subject to payment of the applicable fees and the limitations and restrictions set forth herein, Juniper grants to Customer a non-exclusive and non-transferable license, without right to sublicense, to use the Software, in executable form only, subject to the following use restrictions:

a. Customer shall use Embedded Software solely as embedded in, and for execution on, Juniper equipment originally purchased by Customer from Juniper or an authorized Juniper reseller.

b. Customer shall use the Software on a single hardware chassis having a single processing unit, or as many chassis or processing units for which Customer has paid the applicable license fees; provided, however, with respect to the Steel-Belted Radius or Odyssey Access Client software only, Customer shall use such Software on a single computer containing a single physical random access memory space and containing any number of processors. Use of the Steel-Belted Radius or IMS AAA software on multiple computers or virtual machines (e.g., Solaris zones) requires multiple licenses, regardless of whether such computers or virtualizations are physically contained on a single chassis.

c. Product purchase documents, paper or electronic user documentation, and/or the particular licenses purchased by Customer may specify limits to Customer's use of the Software. Such limits may restrict use to a maximum number of seats, registered endpoints, concurrent users, sessions, calls, connections, subscribers, clusters, nodes, realms, devices, links, ports or transactions, or require the purchase of separate licenses to use particular features, functionalities, services, applications, operations, or capabilities, or provide throughput, performance, configuration, bandwidth, interface, processing, temporal, or geographical limits. In addition, such limits may restrict the use of the Software to managing certain kinds of networks or require the Software to be used only in conjunction with other specific Software. Customer's use of the Software shall be subject to all such limitations and purchase of all applicable licenses.

d. For any trial copy of the Software, Customer's right to use the Software expires 30 days after download, installation or use of the Software. Customer may operate the Software after the 30-day trial period only if Customer pays for a license to do so. Customer may not extend or create an additional trial period by re-installing the Software after the 30-day trial period.

e. The Global Enterprise Edition of the Steel-Belted Radius software may be used by Customer only to manage access to Customer's enterprise network. Specifically, service provider customers are expressly prohibited from using the Global Enterprise Edition of the Steel-Belted Radius software to support any commercial network access services.

The foregoing license is not transferable or assignable by Customer. No license is granted herein to any user who did not originally purchase the applicable license(s) for the Software from Juniper or an authorized Juniper reseller.

4. **Use Prohibitions.** Notwithstanding the foregoing, the license provided herein does not permit the Customer to, and Customer agrees not to and shall not: (a) modify, unbundle, reverse engineer, or create derivative works based on the Software; (b) make unauthorized copies of the Software (except as necessary for backup purposes); (c) rent, sell, transfer, or grant any rights in and to any copy of the Software, in any form, to any third party; (d) remove any proprietary notices, labels, or marks on or in any copy of the Software or any product in which the Software is embedded; (e) distribute any copy of the Software to any third party, including as may be embedded in Juniper equipment sold in the secondhand market; (f) use any 'locked' or key-restricted feature, function, service, application, operation, or capability without first purchasing the applicable license(s) and obtaining a valid key from Juniper, even if such feature, function, service, application, operation, or capability is enabled without a key; (g) distribute any key for the Software provided by Juniper to any third party; (h) use the Software in any manner that extends or is broader than the uses purchased by Customer from Juniper or an authorized Juniper reseller; (i) use Embedded Software on non-Juniper equipment; (j) use Embedded Software (or make it available for use) on Juniper equipment that the Customer did not originally purchase from Juniper or an authorized Juniper reseller; (k) disclose the results of testing or benchmarking of the Software to any third party without the prior written consent of Juniper; or (l) use the Software in any manner other than as expressly provided herein.

5. **Audit.** Customer shall maintain accurate records as necessary to verify compliance with this Agreement. Upon request by Juniper, Customer shall furnish such records to Juniper and certify its compliance with this Agreement.

6. **Confidentiality.** The Parties agree that aspects of the Software and associated documentation are the confidential property of Juniper. As such, Customer shall exercise all reasonable commercial efforts to maintain the Software and associated documentation in confidence, which at a minimum includes restricting access to the Software to Customer employees and contractors having a need to use the Software for Customer's internal business purposes.

7. **Ownership.** Juniper and Juniper's licensors, respectively, retain ownership of all right, title, and interest (including copyright) in and to the Software, associated documentation, and all copies of the Software. Nothing in this Agreement constitutes a transfer or conveyance of any right, title, or interest in the Software or associated documentation, or a sale of the Software, associated documentation, or copies of the Software.

8. **Warranty, Limitation of Liability, Disclaimer of Warranty.** The warranty applicable to the Software shall be as set forth in the warranty statement that accompanies the Software (the "Warranty Statement"). Nothing in this Agreement shall give rise to any obligation to support the Software. Support services may be purchased separately. Any such support shall be governed by a separate, written support services agreement. TO THE MAXIMUM EXTENT PERMITTED BY LAW, JUNIPER SHALL NOT BE LIABLE FOR ANY LOST PROFITS, LOSS OF DATA, OR COSTS OR PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS AGREEMENT, THE SOFTWARE, OR ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. IN NO EVENT SHALL JUNIPER BE LIABLE FOR DAMAGES ARISING FROM UNAUTHORIZED OR IMPROPER USE OF ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. EXCEPT AS EXPRESSLY PROVIDED IN THE WARRANTY STATEMENT TO THE EXTENT PERMITTED BY LAW, JUNIPER DISCLAIMS ANY AND ALL WARRANTIES IN AND TO THE SOFTWARE (WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE), INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT DOES JUNIPER WARRANT THAT THE SOFTWARE, OR ANY EQUIPMENT OR NETWORK RUNNING THE SOFTWARE, WILL OPERATE WITHOUT ERROR OR INTERRUPTION, OR WILL BE FREE OF VULNERABILITY TO INTRUSION OR ATTACK. In no event shall Juniper's or its suppliers' or licensors' liability to Customer, whether in contract, tort (including negligence), breach of warranty, or otherwise, exceed the price paid by Customer for the Software that gave rise to the claim, or if the Software is embedded in another Juniper product, the price paid by Customer for such other product. Customer acknowledges and agrees that Juniper has set its prices and entered into this Agreement in reliance upon the disclaimers of warranty and the limitations of liability set forth herein, that the same reflect an allocation of risk between the Parties (including the risk that a contract remedy may fail of its essential purpose and cause consequential loss), and that the same form an essential basis of the bargain between the Parties.

9. **Termination.** Any breach of this Agreement or failure by Customer to pay any applicable fees due shall result in automatic termination of the license granted herein. Upon such termination, Customer shall destroy or return to Juniper all copies of the Software and related documentation in Customer's possession or control.

10. **Taxes.** All license fees payable under this agreement are exclusive of tax. Customer shall be responsible for paying Taxes arising from the purchase of the license, or importation or use of the Software. If applicable, valid exemption documentation for each taxing jurisdiction shall be provided to Juniper prior to invoicing, and Customer shall promptly notify Juniper if their exemption is revoked or modified. All payments made by Customer shall be net of any applicable withholding tax. Customer will provide reasonable assistance to Juniper in connection with such withholding taxes by promptly: providing Juniper with valid tax receipts and other required documentation showing Customer's payment of any withholding taxes; completing appropriate applications that would reduce the amount of withholding tax to be paid; and notifying and assisting Juniper in any audit or tax proceeding related to transactions hereunder. Customer shall comply with all applicable tax laws and regulations, and Customer will promptly pay or reimburse Juniper for all costs and damages related to any liability incurred by Juniper as a result of Customer's non-compliance or delay with its responsibilities herein. Customer's obligations under this Section shall survive termination or expiration of this Agreement.

11. **Export.** Customer agrees to comply with all applicable export laws and restrictions and regulations of any United States and any applicable foreign agency or authority, and not to export or re-export the Software or any direct product thereof in violation of any such restrictions, laws or regulations, or without all necessary approvals. Customer shall be liable for any such violations. The version of the Software supplied to Customer may contain encryption or other capabilities restricting Customer's ability to export the Software without an export license.

12. **Commercial Computer Software.** The Software is "commercial computer software" and is provided with restricted rights. Use, duplication, or disclosure by the United States government is subject to restrictions set forth in this Agreement and as provided in DFARS 227.7201 through 227.7202-4, FAR 12.212, FAR 27.405(b)(2), FAR 52.227-19, or FAR 52.227-14(ALT III) as applicable.

13. **Interface Information.** To the extent required by applicable law, and at Customer's written request, Juniper shall provide Customer with the interface information needed to achieve interoperability between the Software and another independently created program, on payment of applicable fee, if any. Customer shall observe strict obligations of confidentiality with respect to such information and shall use such information in compliance with any applicable terms and conditions upon which Juniper makes such information available.

14. **Third Party Software.** Any licensor of Juniper whose software is embedded in the Software and any supplier of Juniper whose products or technology are embedded in (or services are accessed by) the Software shall be a third party beneficiary with respect to this Agreement, and such licensor or vendor shall have the right to enforce this Agreement in its own name as if it were Juniper. In addition, certain third party software may be provided with the Software and is subject to the accompanying license(s), if any, of its respective owner(s). To the extent portions of the Software are distributed under and subject to open source licenses obligating Juniper to make the source code for such portions publicly available (such as the GNU General Public License ("GPL") or the GNU Library General Public License ("LGPL")), Juniper will make such source code portions (including Juniper modifications, as appropriate) available upon request for a period of up to three years from the date of distribution. Such request can be made in writing to Juniper Networks, Inc., 1194 N. Mathilda Ave., Sunnyvale, CA 94089, ATTN: General Counsel. You may obtain a copy of the GPL at http://www.gnu.org/licenses/gpl.html, and a copy of the LGPL at http://www.gnu.org/licenses/lgpl.html.

15. **Miscellaneous.** This Agreement shall be governed by the laws of the State of California without reference to its conflicts of laws principles. The provisions of the U.N. Convention for the International Sale of Goods shall not apply to this Agreement. For any disputes arising under this Agreement, the Parties hereby consent to the personal and exclusive jurisdiction of, and venue in, the state and federal courts within Santa Clara County, California. This Agreement constitutes the entire and sole agreement between Juniper and the Customer with respect to the Software, and supersedes all prior and contemporaneous

agreements relating to the Software, whether oral or written (including any inconsistent terms contained in a purchase order), except that the terms of a separate written agreement executed by an authorized Juniper representative and Customer shall govern to the extent such terms are inconsistent or conflict with terms contained herein. No modification to this Agreement nor any waiver of any rights hereunder shall be effective unless expressly assented to in writing by the party to be charged. If any portion of this Agreement is held invalid, the Parties agree that such invalidity shall not affect the validity of the remainder of this Agreement. This Agreement and associated documentation has been written in the English language, and the Parties agree that the English version will govern. (For Canada: Les parties aux présentés confirment leur volonté que cette convention de même que tous les documents y compris tout avis qui s'y rattaché, soient redigés en langue anglaise. (Translation: The parties confirm that this Agreement and all related documentation is and will be in the English language)).

Abbreviated Table of Contents

		About This Guide	xxiii
Part 1		**MPLS Applications**	
	Chapter 1	GMPLS	3
	Chapter 2	Connecting IPv6 Islands with IPv4 MPLS	43
	Chapter 3	Multiple Instances for Label Distribution Protocol	61
	Chapter 4	MPLS LSP Link Protection and Node-Link Protection	101
	Chapter 5	RSVP LSP Tunnels	141
	Chapter 6	Simplified Interinstance Route Sharing	167
Part 2		**Routing Protocols**	
	Chapter 7	Logical Systems	195
	Chapter 8	OSPF Version 3 for IPv6	239
	Chapter 9	Multitopology Routing	271
Part 3		**VPNs**	
	Chapter 10	Layer 2 Circuits	285
	Chapter 11	Multicast over Layer 3 VPNs	339
	Chapter 12	Translational Cross-Connect and Layer 2.5 VPNs	473
	Chapter 13	Virtual Private LAN Service	501
Part 4		**Index**	
		Index	597

Table of Contents

About This Guide xxiii

JUNOS Documentation and Release Notes xxiii
Objectives xxiv
Audience xxiv
Supported Routing Platforms xxv
Using the Indexes xxv
Using the Examples in This Manual xxv
Merging a Full Example xxvi
Merging a Snippet xxvi
Documentation Conventions xxvii
Documentation Feedback xxix
Requesting Technical Support xxix
Self-Help Online Tools and Resources xxix
Opening a Case with JTAC xxx

Part 1 MPLS Applications

Chapter 1 GMPLS 3

GMPLS Overview 4
GMPLS System Requirements 6
GMPLS Phase 2 Implementation 7
GMPLS Operation 9
Configuring GMPLS 9
Configuring Link Management Protocol Traffic Engineering Links 10
Configuring Link Management Protocol Peers 10
Configuring Peer Interfaces in OSPF and RSVP 11
Establishing GMPLS LSP Path Information 12
Defining GMPLS Label-Switched Paths 12
Displaying Local Identifiers and Configuring Remote Identifiers 13
Option: Tearing Down GMPLS LSPs Gracefully 14
Option: Allowing Nonpacket GMPLS LSPs to Establish a Path Through JUNOS-Based Routers 14
Option: Selecting the Peer Model for GMPLS 15
Option: Selecting the Overlay Model for GMPLS 15
Option: GMPLS Graceful Restart 15
Option: Configuring an LMP Control Channel 16

Option: Configuring GMPLS Support for Unnumbered Links 18
GMPLS Configuration Examples 18
Example: GMPLS Configuration 19
Verifying Your Work 23
Router A Status 24
Router C Status 29
Example: Configuring Traffic Engineering Link and Interface Identifiers 30
Example: LMP Control Channel Configuration 31
Verifying Your Work 37
Router 1 Status 37
Router 4 Status 38
For More Information 39
Revision History 40

Chapter 2 Connecting IPv6 Islands with IPv4 MPLS 43

Connecting IPv6 Islands with IPv4 MPLS Overview 43
Connecting IPv6 Islands with IPv4 MPLS System Requirements 45
Configuring an IPv4 MPLS Tunnel to Carry IPv6 Traffic 46
Configuring IPv6 on the Customer and Core-Facing Interfaces 46
Configuring MPLS and RSVP from PE Router to PE Router to Create a Tunnel 47
Enabling IPv6 Tunneling in MPLS 47
Configuring Multiprotocol BGP to Carry IPv6 Traffic 47
Example: Connecting IPv6 Islands over an MPLS Tunnel Configuration 48
Verifying Your Work 54
Router CE1 Status 55
Router PE1 Status 55
Router PE2 Status 56
Router CE2 Status 57
For More Information 58
Revision History 58

Chapter 3 Multiple Instances for Label Distribution Protocol 61

Multiple Instances for Label Distribution Protocol Overview 61
Multiple Instances for Label Distribution Protocol System Requirements 62
Example: Configuring Multiple-Instance LDP 63
Verifying Your Work 82
Router CE3 Status 83
Router PE3 Status 83
Router CE1 Status 85
Router PE1 Status 87
Router PE2 Status 89
Router CE2 Status 94
Router PE4 Status 95
Router CE4 Status 97
Revision History 98

Chapter 4 MPLS LSP Link Protection and Node-Link Protection 101

MPLS LSP Link Protection and Node-Link Protection Overview 101
Link Protection 103
Node-Link Protection 104
MPLS LSP Link Protection and Node-Link Protection System Requirements 105
Configuring MPLS LSP Link Protection or Node-Link Protection 106
Configuring Link Protection or Node-Link Protection on the LSP 106
Configuring Link Protection on the RSVP Interfaces Traversed by the LSP 106
Option: Configuring Multiple Bypass LSPs, Manual Bypass LSPs, and Link Protection Priority 107
Option: Adding Class of Service to a Link-Protected LSP or a Bypass LSP 108
Verifying MPLS LSP Link Protection and Node Link Protection 109
MPLS LSP Link Protection or Node-Link Protection Configuration Examples 109
Example: Configuring MPLS LSP Link Protection 110
Verifying Your Work 115
Case 1: Normal Operation 115
Case 2: When the Link from Router 1 to Router 3 Is Disabled 122
Case 3: When the Link from Router 3 to Router 2 Is Disabled 125
Example: Node-Link Protection Configuration 129
Verifying Your Work 135
For More Information 139
Revision History 139

Chapter 5 RSVP LSP Tunnels 141

RSVP LSP Tunnels Overview 141
RSVP LSP Tunnels System Requirements 142
RSVP LSP Tunneling Operation 143
Configuring an RSVP LSP Tunnel 143
Configuring Link Management Protocol Traffic Engineering Links 144
Configuring Link Management Protocol Peers 144
Configuring Peer Interfaces in OSPF and RSVP 145
Establishing FA-LSP Path Information 145
Defining Label-Switched Paths for the FA-LSP 146
Creating End-to-End LSPs to Traverse the FA-LSP 146
Option: Tearing Down RSVP LSPs Gracefully 146
Example: RSVP LSP Tunnel Configuration 147
Verifying Your Work 159
Router 0 160
Router 1 164
For More Information 165
Revision History 165

Chapter 6 Simplified Interinstance Route Sharing 167

Simplified Interinstance Route Sharing Overview 167
Simplified Interinstance Route Sharing System Requirements 168
Simplified Interinstance Configuration 169
Instance Export Using an IGP Export Policy 171
Configuring Overlapping VPNs 171
Example: Configuring Overlapping VPNs 175
Verifying Your Work 181
Router PE1 Status 181
Configuring Nonforwarding Instances 183
Example: Nonforwarding Instances Configuration 185
Verifying Your Work 188
Router PE2 Status 188
Router CE3 Status 189
For More Information 189
Revision History 189

Part 2 Routing Protocols

Chapter 7 Logical Systems 195

Logical Systems Overview 195
Logical Systems System Requirements 199
Configuring Logical Systems 200
Configuring Logical System Administrators (Master Administrator) 200
Configuring Logical System Interface Properties (Master Administrator) 201
Assigning Logical Interfaces to the Logical System (Master or Logical System Administrator) 201
Configuring Protocols, Routing, and Policy Statements for the Logical System (Master or Logical System Administrator) 201
Configuring Other Logical System Statements 202
Example: Configuring Logical Systems 205
Verifying Your Work 224
Router CE1 Status 225
Router CE2 Status 226
Router CE3 Status 226
Router PE1 Status: Main Router 226
Router PE1 Status: LS1 227
Router PE1 Status: LS2 230
Router P0 Status: Main Router 230
Router P0 Status: LS1 231
Router P0 Status: LS2 231
Router PE2 Status: Main Router 231
Router PE2 Status: LS1 233
Router PE2 Status: LS2 234
Router CE5 Status 235

Router CE6 Status 235
Router CE7 Status 236
Logical System Administrator Verification Output 236
Verifying Routing Instance Connectivity 236
For More Information 237
Revision History 237

Chapter 8 OSPF Version 3 for IPv6 239

OSPF Version 3 for IPv6 Overview 239
OSPF Version 3 for IPv6 System Requirements 241
Configuring OSPFv3 for IPv6 241
Configuring OSPFv3 as the Routing Protocol 242
Configuring Interfaces in OSPFv3 Areas 242
Configuring Virtual Links for OSPFv3 242
Example: Configuring OSPFv3 for IPv6 243
Verifying Your Work 249
Router 0 Status 250
Router 1 Status 253
Router 2 Status 255
Router 3 Status 258
Router 4 Status 262
Router 5 Status 265
For More Information 267
Revision History 267

Chapter 9 Multitopology Routing 271

Multitopology Routing Overview 271
Multitopology Routing System Requirements 274
Configuring Multitopology Routing 274
Configuring Topologies 274
Configuring Filter-Based Forwarding 275
Configuring BGP for Multitopology Routing 275
Option: Configuring OSPF for Multitopology Routing 276
Option: Configuring Static Routes for Multitopology Routing 276
Option: Configuring Route Resolution Policy 277
Example: Multitopology Routing Configuration 277
Verifying Your Work 281
For More Information 282
Revision History 282

Part 3 VPNs

Chapter 10 Layer 2 Circuits 285

Layer 2 Circuits Overview 286
Layer 2 Circuits System Requirements 289
Configuring Layer 2 Circuits 290
Configuring CCC Encapsulation on CE-Facing Ethernet Interfaces 290
Configuring CCC Encapsulation on CE-Facing SONET/SDH Interfaces 291
Configuring a CCC Encapsulation and a Layer 2 Circuit Mode on CE-Facing ATM2 IQ Interfaces 292
Configuring the MPLS Family on Core Interfaces 293
Configuring the Layer 2 Circuit Neighbor Address and Virtual Circuit Identifier 294
Configuring LDP and an IGP to Transport Layer 2 Circuits 295
Option: Applying Traffic Engineering to a Layer 2 Circuit 296
Option: Mapping Layer 2 Protocol Control Information into a Layer 2 Circuit 297
Option: Configuring APS for Layer 2 Circuits 297
Option: Configuring Layer 2 Circuit Trunk Mode on ATM2 IQ Interfaces 298
Option: Reserving LSP Bandwidth for a Layer 2 Circuit 300
Option: Selecting an MTU for a Layer 2 Circuit 301
Option: Configuring Local Interface Switching for a Layer 2 Circuit 302
Option: Configuring Layer 2 Circuits Simultaneously over RSVP and LDP LSPs 302
Layer 2 Circuit Configuration Examples 303
Example: Ethernet-Based Layer 2 Circuit Configuration 303
Verifying Your Work 307
Router PE1 Status 308
Router P0 Status 309
Router PE2 Status 309
Example: SONET/SDH-Based Layer 2 Circuit Configuration 310
Verifying Your Work 314
Example: ATM2 IQ-Based Layer 2 Circuit Configuration 315
Verifying Your Work 321
Example: Layer 2 Circuit Traffic Engineering over Multiple LSPs Configuration 324
Verifying Your Work 333
Example: APS for a Layer 2 Circuit Configuration 334
Verifying Your Work 335
For More Information 336
Revision History 337

Chapter 11 Multicast over Layer 3 VPNs 339

Multicast over Layer 3 VPNs Overview 340
Multiprotocol BGP-Based Multicast VPNs: Next-Generation 341

Dual PIM Multicast VPNs: Draft Rosen 341
System Requirements for Multiprotocol BGP-Based Multicast VPNs: Next-Generation 342
System Requirements for Dual PIM Multicast VPNs: Draft Rosen 343
Building Blocks for Configuring Multiprotocol BGP-Based Multicast VPNs: Next-Generation 344
Creating a Unique Logical Loopback Interface for the Routing Instance for MBGP MVPNs 345
Configuring Interfaces for Layer 3 VPNs 345
Configuring BGP, MPLS, RSVP, and an IGP on the PE and Core Routers for MBGP MVPNs 345
Creating a Routing Instance for a Multiprotocol BGP-Based Multicast VPN 346
Option: Configuring Sender and Receiver Sites 346
Option: Specifying Route Targets 347
Configuring Provider Tunnels 349
Enabling Multicast VPN in BGP 349
Configuring Intra-AS Inclusive Point-to-Multipoint Traffic Engineering LSPs 350
Configuring Intra-AS Selective Provider Tunnels 352
Configuring the Master PIM Instance on the PE Router for BGP-Based Multicast VPNs 354
Configuring the Router's IPv4 Bootstrap Router Priority 355
Configuring MBGP MVPNs to Support IPv6 Multicast Traffic 355
Example: Configuring MBGP MVPN Senders and Receivers using PIM SM and RSVP-TE Provider Tunnels 358
Verifying Your Work 360
show mvpn c-multicast 361
show mvpn instance 362
show mvpn neighbor 364
Example: Configuring MBGP Multicast VPNs 365
Example: Configuring MBGP Multicast VPN Topology Variations 385
MBGP Multicast VPN Extranets Overview 397
MBGP Multicast VPN Extranets Application 397
MBGP Multicast VPN Extranets Configuration Guidelines 398
Example: Configuring MBGP Multicast VPN Extranets 398
Dual PIM Draft-Rosen Multicast VPN Operation 440
Configuring Draft-Rosen Multicast VPNs 442
Configuring BGP, MPLS, RSVP, and an IGP on the PE and Core Routers for Draft Rosen VPNs 442
Creating a Unique Logical Loopback Interface for the Routing Instance for Draft Rosen VPNs 443
Configuring the Master PIM Instance on the PE Router in the Service Provider Network 443
Configuring PIM and the VPN Group Address in a Routing Instance 444
Option: Configuring PIM Sparse Mode Graceful Restart for a Layer 3 VPN 445
Option: Configuring Multicast Distribution Trees for Data 445

Option: Configuring MSDP Within a Layer 3 VPN 446
Draft-Rosen Multicast VPNs Examples 447
Example: Basic IPv4 Multicast over a Layer 3 VPN Configuration 448
Verifying Your Work 451
RP Information 452
PIM Information Before Multicast Transmission 452
Successful PIM Join Verification 454
Example: IPv4 Multicast with Interprovider VPNs Configuration 461
Verifying Your Work 465
Router CE0 Status 465
Router PE0 Status 465
Router P0 Status 467
Router P1 Status 468
Router PE1 Status 469
Router CE1 Status 470
For More Information 471
Revision History 471

Chapter 12 Translational Cross-Connect and Layer 2.5 VPNs 473

Translational Cross-Connect and Layer 2.5 VPNs Overview 473
Translational Cross-Connect and Layer 2.5 VPNs System Requirements 474
Configuring TCC Interface Switching 476
Defining the Encapsulation for Layer 2 TCC Switching 477
Configuring ARP Functionality for Ethernet TCC Circuits 478
Configuring ARP Functionality for Extended VLAN Circuits 478
Configuring Static ARP For Ethernet and Extended VLAN TCCs 479
Defining the Connection for Layer 2 TCC Switching 480
Configuring MPLS 480
TCC Configuration Examples 480
Example: PPP to ATM TCC Configuration 481
Verifying Your Work 482
Example: Frame Relay to Fast Ethernet TCC Configuration 483
Verifying Your Work 485
Configuring Layer 2.5 VPNs 485
Configuring the Encapsulation on Interfaces Participating in the Layer 2.5 VPN 486
Configuring the Layer 2.5 VPN 487
Option: Configuring ISO or MPLS Traffic on T Series and M320 Routers 487
Example: Layer 2.5 VPN Configuration 488
Verifying Your Work 493
Router PE1 Status 494
Router PE2 Status 495
Router P Status 497
For More Information 497
Revision History 498

Chapter 13 Virtual Private LAN Service 501

Virtual Private LAN Service Overview 502
Virtual Private LAN Service System Requirements 505
Configuring VPLS 507
Configuring Routing Protocols on the PE and Core Routers 508
Configuring VPLS Encapsulation on CE-Facing Interfaces 508
Configuring LDP Signaling for VPLS 510
Configuring a VPLS Instance with BGP Signaling 511
Configuring Interworking Between BGP Signaling and LDP Signaling in VPLS Instances 512
Configuring Multihoming on a VPLS Border Router 514
Option: Selecting an LSP for the VPLS Routing Instance to Traverse 515
Option: Configuring VPLS Multihoming with BGP Signaling 516
Option: Configuring VPLS Traffic Flooding over a Point-to-Multipoint LSP 519
Option: Configuring Automatic Site Selection 521
Option: Configuring VPLS to Use LSI Interfaces 522
Option: Configuring Tunnel Services on MX Series Routers 523
Configuring Integrated Routing and Bridging in a VPLS Instance (MX Series Routers Only) 523
Configuring VLAN IDs in a VPLS Instance (MX Series Routers Only) 524
Defining a VPLS Firewall Policer 525
Defining a VPLS Firewall Filter 526
Restricting Broadcast Packets in VPLS 527
Option: Enabling VPLS Class of Service 528
Option: Enabling VPLS Graceful Restart 528
Configuring the VPLS MAC Address Timeout 529
Option: Configuring VPLS Interinstance Bridging and Routing 530
Option: Selecting Interfaces to Process VPLS Traffic 531
Option: Limiting the Number of MAC Addresses Learned on a VPLS Interface 532
Option: Optimizing VPLS Traffic Flows 533
Option: Aggregated Interfaces for VPLS 533
Configuration Steps for VPLS Graceful Routing Engine Switchover and Nonstop Active Routing 534
Configuring VPLS Graceful Routing Engine Switchover and Nonstop Active Routing 534
Synchronizing the Routing Engine Configuration 535
Verifying VPLS Nonstop Active Routing Operation 536
Tracing VPLS Nonstop Active Routing Synchronization Events 536
Option: Configuring the Spanning Tree Protocol and VPLS on MX Series Routers 536

Filtering Layer 2 Packets in a VPLS Instance (MX Series Routers Only) 537
VPLS Configuration Examples 537
Example: VPLS Configuration (BGP Signaling) 538
Verifying Your Work 544
Example: VPLS Configuration (BGP and LDP Interworking) 549
Verifying Your Work 559
Example: Configuring VPLS Nonstop Active Routing 563
Example: Configuring Inter-AS VPLS with MAC Processing at the ASBR 564
For More Information 591
Revision History 591

Part 4 Index

Index 597

List of Figures

Part 1 MPLS Applications

Chapter 1 GMPLS 3

Figure 1: GMPLS LSP Hierarchy 4
Figure 2: GMPLS Topology Diagram 19
Figure 3: Traffic Engineering Link and Interface ID Example 30
Figure 4: LMP Control Channel Topology Diagram 31

Chapter 2 Connecting IPv6 Islands with IPv4 MPLS 43

Figure 5: Connecting IPv6 Islands over MPLS 44
Figure 6: IPv6 over an MPLS Tunnel 48

Chapter 3 Multiple Instances for Label Distribution Protocol 61

Figure 7: Carrier-of-Carriers Example 62
Figure 8: Multiple-Instance LDP Topology Diagram 64

Chapter 4 MPLS LSP Link Protection and Node-Link Protection 101

Figure 9: Link Protection and Node-Link Protection Comparison 104
Figure 10: MPLS LSP Link Protection Topology Diagram 110
Figure 11: Node-Link Protection Topology Diagram 129

Chapter 5 RSVP LSP Tunnels 141

Figure 12: RSVP LSP Tunnel Topology Diagram 147

Chapter 6 Simplified Interinstance Route Sharing 167

Figure 13: Overlapping VPNs Topology Diagram 175
Figure 14: Nonforwarding Instance Concept 183
Figure 15: Nonforwarding Instances Topology Diagram 185

Part 2 Routing Protocols

Chapter 7 Logical Systems 195

Figure 16: Logical Systems Concept 196
Figure 17: Logical System Topology Diagram 205

Chapter 8 OSPF Version 3 for IPv6 239

Figure 18: OSPFv3 for IPv6 Topology Diagram 243

Chapter 9 Multitopology Routing 271

Figure 19: MT-OSPF Area Boundary 272
Figure 20: BGP Route Resolution 273
Figure 21: Route Resolution for MTR 273
Figure 22: Route Resolution in Multitopology Routing 278

Part 3 VPNs

Chapter 10 Layer 2 Circuits 285

Figure 23: Layer 2 Circuit Connection 286
Figure 24: Layer 2 Circuit Concept 287
Figure 25: Ethernet-Based Layer 2 Circuit Topology Diagram 303
Figure 26: SONET/SDH-Based Layer 2 Circuit Topology Diagram 310
Figure 27: ATM2 IQ-Based Layer 2 Circuit Topology Diagram 315
Figure 28: Layer 2 Circuit Traffic Engineering Topology Diagram 324
Figure 29: APS for a Layer 2 Circuit Topology Diagram 334

Chapter 11 Multicast over Layer 3 VPNs 339

Figure 30: Multiprotocol BGP Multicast VPN Senders and Receivers 358
Figure 31: Multicast Over Layer 3 VPN Example Topology 366
Figure 32: MBGP MVPN Topology Variations Diagram 386
Figure 33: MVPN Extranets Topology Diagram 399
Figure 34: Multicast Over Layer 3 VPN Operation 440
Figure 35: Basic IPv4 Multicast over a Layer 3 VPN Topology Diagram 448
Figure 36: IPv4 Multicast with Interprovider VPNs Topology Diagram 461

Chapter 12 Translational Cross-Connect and Layer 2.5 VPNs 473

Figure 37: TCC Concept Example 474
Figure 38: Layer 2 TCC Switching 476
Figure 39: TCC Interface Switching—PPP to ATM 481
Figure 40: TCC Interface Switching—Frame Relay to Fast Ethernet 483
Figure 41: Layer 2.5 VPN Topology Diagram 488

Chapter 13 Virtual Private LAN Service 501

Figure 42: Ethernet Switching Example 503
Figure 43: VPLS Introductory Example 504
Figure 44: Topology for BGP/LDP Interworking in a VPLS Instance 513
Figure 45: Multihoming for Border Area Routers 515
Figure 46: Traditional Flooding in a VPLS Routing Instance 519
Figure 47: VPLS Routing Instance with Point-to-Multipoint LSP Flooding 520
Figure 48: VPLS Topology Diagram 538
Figure 49: Topology for VPLS Configuration Example 549
Figure 50: Inter-AS VPLS with MAC Operations Example Topology 565

List of Tables

About This Guide **xxiii**

Table 1: Notice Icons xxvii

Table 2: Text and Syntax Conventions xxvii

Part 1 MPLS Applications

Chapter 1 GMPLS **3**

Table 3: Default Values for LMP Protocol Fields 17

Chapter 3 Multiple Instances for Label Distribution Protocol **61**

Table 4: Multiple-Instance LDP Example—Routing Protocol Summary 64

Table 5: Multiple-Instance LDP Example—Loopback Addresses 65

Chapter 6 Simplified Interinstance Route Sharing **167**

Table 6: Nonforwarding Instances—Loopback Addresses 185

Part 3 VPNs

Chapter 13 Virtual Private LAN Service **501**

Table 7: Router Interface Addresses for VPLS Configuration Example 549

About This Guide

This preface provides the following guidelines for using the *JUNOS® Software Feature Guide*:

- JUNOS Documentation and Release Notes on page xxiii
- Objectives on page xxiv
- Audience on page xxiv
- Supported Routing Platforms on page xxv
- Using the Indexes on page xxv
- Using the Examples in This Manual on page xxv
- Documentation Conventions on page xxvii
- Documentation Feedback on page xxix
- Requesting Technical Support on page xxix

JUNOS Documentation and Release Notes

For a list of related JUNOS documentation, see http://www.juniper.net/techpubs/software/junos/.

If the information in the latest release notes differs from the information in the documentation, follow the *JUNOS Software Release Notes*.

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at http://www.juniper.net/techpubs/.

Juniper Networks supports a technical book program to publish books by Juniper Networks engineers and subject matter experts with book publishers around the world. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration using JUNOS Software and Juniper Networks devices. In addition, the Juniper Networks Technical Library, published in conjunction with O'Reilly Media, explores improving network security, reliability, and availability using JUNOS configuration techniques. All the books are for sale at technical bookstores and book outlets around the world. The current list can be viewed at http://www.juniper.net/books .

Objectives

Several Juniper Networks customers requested a new series of guides to supplement the current documentation set. Although the JUNOS Software configuration guides thoroughly describe individual topics (such as interface configuration or routing), they might not properly address complex features that span several of the individual configuration guides (such as multicast VPNs or interinstance route sharing).

As a result, the *JUNOS Feature Guide* series is designed to explore the more complicated software features available in Juniper Networks routing platforms. The *Feature Guide* combines all the relevant configuration statements and operational mode commands in one place, precluding the need for users to search multiple manuals for solutions to technical problems.

The general outline for each *Feature Guide* chapter is as follows:

- Overview—Introduces the feature topic to the user.
- System Requirements—Lists the equipment and software needed to implement a feature.
- Terms and Acronyms—Explains the terminology used with each feature example.
- Feature Implementation—Discusses the background needed to understand how to implement the feature.
  - Configuring the feature—Shows the steps that are needed to configure a feature.
  - Example—Gives an actual configuration example, along with verification commands and output.
  - For More Information—Provides additional resources and information for further understanding of the feature.

NOTE: For additional information about the JUNOS Software—either corrections to or information that might have been omitted from this guide—see the software release notes at http://www.juniper.net/.

Audience

This guide is designed for network administrators who are configuring and monitoring a Juniper Networks M Series, MX Series, T Series, EX Series, or J Series router or switch.

To use this guide, you need a broad understanding of networks in general, the Internet in particular, networking principles, and network configuration. You must also be familiar with one or more of the following Internet routing protocols:

- Border Gateway Protocol (BGP)
- Distance Vector Multicast Routing Protocol (DVMRP)

- Intermediate System-to-Intermediate System (IS-IS)
- Internet Control Message Protocol (ICMP) router discovery
- Internet Group Management Protocol (IGMP)
- Multiprotocol Label Switching (MPLS)
- Open Shortest Path First (OSPF)
- Protocol-Independent Multicast (PIM)
- Resource Reservation Protocol (RSVP)
- Routing Information Protocol (RIP)
- Simple Network Management Protocol (SNMP)

Personnel operating the equipment must be trained and competent; must not conduct themselves in a careless, willfully negligent, or hostile manner; and must abide by the instructions provided by the documentation.

Supported Routing Platforms

For the features described in this manual, the JUNOS Software currently supports the following routing platforms:

- J Series
- M Series
- MX Series
- T Series

Using the Indexes

This reference contains two indexes: a standard index with topic entries, and an index of commands.

Using the Examples in This Manual

If you want to use the examples in this manual, you can use the load merge or the load merge relative command. These commands cause the software to merge the incoming configuration into the current candidate configuration. If the example configuration contains the top level of the hierarchy (or multiple hierarchies), the example is a *full example*. In this case, use the load merge command.

If the example configuration does not start at the top level of the hierarchy, the example is a *snippet*. In this case, use the load merge relative command. These procedures are described in the following sections.

Merging a Full Example

To merge a full example, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration example into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following configuration to a file and name the file ex-script.conf. Copy the ex-script.conf file to the /var/tmp directory on your routing platform.

   ```
   system {
       scripts {
           commit {
               file ex-script.xsl;
           }
       }
   }
   interfaces {
       fxp0 {
           disable;
           unit 0 {
               family inet {
                   address 10.0.0.1/24;
               }
           }
       }
   }
   ```

2. Merge the contents of the file into your routing platform configuration by issuing the load merge configuration mode command:

   ```
   [edit]
   user@host#load merge /var/tmp/ex-script.conf
   load complete
   ```

Merging a Snippet

To merge a snippet, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration snippet into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following snippet to a file and name the file ex-script-snippet.conf. Copy the ex-script-snippet.conf file to the /var/tmp directory on your routing platform.

   ```
   commit {
       file ex-script-snippet.xsl; }
   ```

2. Move to the hierarchy level that is relevant for this snippet by issuing the following configuration mode command:

   ```
   [edit]
   user@host#edit system scripts
   [edit system scripts]
   ```

3. Merge the contents of the file into your routing platform configuration by issuing the load merge relative configuration mode command:

   ```
   [edit system scripts]
   user@host#load merge relative /var/tmp/ex-script-snippet.conf
   load complete
   ```

For more information about the load command, see the *JUNOS CLI User Guide*.

Documentation Conventions

Table 1 on page xxvii defines notice icons used in this guide.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.

Table 2 on page xxvii defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the configure command: user@host> **configure**
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host> show chassis alarms` `No alarms currently active`

Table 2: Text and Syntax Conventions *(continued)*

Convention	Description	Examples
Italic text like this	■ Introduces important new terms. ■ Identifies book names. ■ Identifies RFC and Internet draft titles.	■ A policy *term* is a named structure that defines match conditions and actions. ■ *JUNOS System Basics Configuration Guide* ■ RFC 1997, *BGP Communities Attribute*
Italic text like this	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*
Plain text like this	Represents names of configuration statements, commands, files, and directories; IP addresses; configuration hierarchy levels; or labels on routing platform components.	■ To configure a stub area, include the stub statement at the [edit protocols ospf area area-id] hierarchy level. ■ The console port is labeled CONSOLE.
< > (angle brackets)	Enclose optional keywords or variables.	stub <default-metric *metric*>;
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	broadcast | multicast (*string1* | *string2* | *string3*)
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	rsvp { # Required for dynamic MPLS only
[ ] (square brackets)	Enclose a variable for which you can substitute one or more values.	community name members [*community-ids*]

Indention and braces ({ })	Identify a level in the configuration hierarchy.	[edit]
		routing-options {
		  static {
		  route default {
		  nexthop *address*;
		  retain;
		  }
		  }
		}

; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
J-Web GUI Conventions		
Bold text like this	Represents J-Web graphical user interface (GUI) items you click or select.	■ In the Logical Interfaces box, select **All Interfaces**. ■ To cancel the configuration, click **Cancel**.
> (bold right angle bracket)	Separates levels in a hierarchy of J-Web selections.	In the configuration editor hierarchy, select **Protocols > Ospf**.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can send your comments to techpubs-comments@juniper.net, or fill out the documentation feedback form at https://www.juniper.net/cgi-bin/docbugreport/. If you are using e-mail, be sure to include the following information with your comments:

- Document or topic name
- URL or page number
- Software release version (if applicable)

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract, or are covered under warranty, and need postsales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the JTAC User Guide located at http://www.juniper.net/customers/support/downloads/710059.pdf .
- Product warranties—For product warranty information, visit http://www.juniper.net/support/warranty/ .
- JTAC Hours of Operation —The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: http://www.juniper.net/customers/support/
- Search for known bugs: http://www2.juniper.net/kb/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/
- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: https://www.juniper.net/alerts/
- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: https://tools.juniper.net/SerialNumberEntitlementSearch/

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at http://www.juniper.net/cm/ .
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, visit us at http://www.juniper.net/support/requesting-support.html

Part 1
MPLS Applications

- GMPLS on page 3
- Connecting IPv6 Islands with IPv4 MPLS on page 43
- Multiple Instances for Label Distribution Protocol on page 61
- MPLS LSP Link Protection and Node-Link Protection on page 101
- RSVP LSP Tunnels on page 141
- Simplified Interinstance Route Sharing on page 167

Chapter 1

GMPLS

This chapter covers the following topics:

- GMPLS Overview on page 4
- GMPLS System Requirements on page 6
- GMPLS Terms and Acronyms on page 6
- GMPLS Phase 2 Implementation on page 7
- GMPLS Operation on page 9
- Configuring GMPLS on page 9
- Configuring Link Management Protocol Traffic Engineering Links on page 10
- Configuring Link Management Protocol Peers on page 10
- Configuring Peer Interfaces in OSPF and RSVP on page 11
- Establishing GMPLS LSP Path Information on page 12
- Defining GMPLS Label-Switched Paths on page 12
- Displaying Local Identifiers and Configuring Remote Identifiers on page 13
- Option: Tearing Down GMPLS LSPs Gracefully on page 14
- Option: Allowing Nonpacket GMPLS LSPs to Establish a Path Through JUNOS-Based Routers on page 14
- Option: Selecting the Peer Model for GMPLS on page 15
- Option: Selecting the Overlay Model for GMPLS on page 15
- Option: GMPLS Graceful Restart on page 15
- Option: Configuring an LMP Control Channel on page 16
- Option: Configuring GMPLS Support for Unnumbered Links on page 18
- GMPLS Configuration Examples on page 18
- Example: GMPLS Configuration on page 19
- Example: Configuring Traffic Engineering Link and Interface Identifiers on page 30
- Example: LMP Control Channel Configuration on page 31
- For More Information on page 39
- Revision History on page 40

GMPLS Overview

Generalized Multiprotocol Label Switching (GMPLS) is the next-generation implementation of Multiprotocol Label Switching (MPLS). GMPLS extends the functionality of MPLS to include a wider range of label-switched path (LSP) options for a variety of network devices.

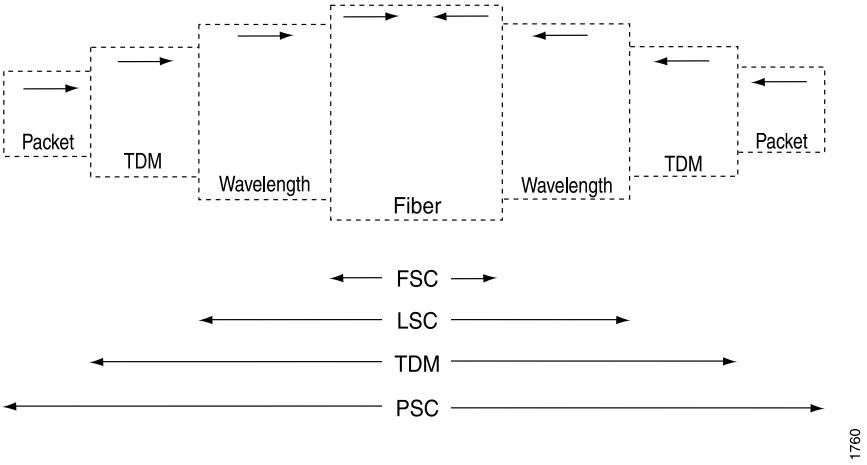
You should have a general understanding of MPLS, label switching concepts, and GMPLS Phase 1. For more information about MPLS, see the *JUNOS MPLS Applications Configuration Guide*. For more information about GMPLS Phase 1, see the *JUNOS 5.5 Feature Guide* at: http://www.juniper.net/techpubs/software/junos/junos55/feature-guide55/feature-guide-55.pdf.

Traditional MPLS is designed to carry Layer 3 IP traffic by establishing IP-based paths and associating these paths with arbitrarily assigned labels. These labels can either be configured explicitly by a network administrator or dynamically assigned by a protocol such as the Label Distribution Protocol (LDP) or Resource Reservation Protocol (RSVP).

In contrast, GMPLS can carry various types of Layer 1 through Layer 3 traffic. GMPLS labels and LSPs can be processed at four levels, as depicted in Figure 1 on page 4. The levels are Fiber-Switched Capable (FSC), Lambda-Switched Capable (LSC), Time-Division Multiplexing Capable (TDM), and Packet-Switched Capable (PSC).

LSPs must start and end on links with the same switching capability. To send an LSP, a label-switched device must communicate with another device at the same layer of the Open System Interconnection (OSI) model. Thus, routers can set up PSC LSPs with other routers at Layer 3, and SONET/SDH add/drop multiplexers (ADMs) can establish TDM LSPs with other ADMs at Layer 1. As seen in Figure 1 on page 4, a router PSC LSP can be carried over a TDM LSP, a TDM LSP can be carried over a wavelength LSC LSP, and so on.

Figure 1: GMPLS LSP Hierarchy



This extension of the MPLS protocol expands the number of devices that can participate in label switching. Lower layer devices, such as optical cross-connects (OXCs) and SONET/SDH ADMs, can now participate in GMPLS signaling and set up paths to transfer data. Additionally, routers can participate in signaling optical paths across a transport network.

GMPLS labeling is also more flexible than MPLS. A GMPLS label can represent a TDM time slot, a Dense Wavelength Division Multiplexing (DWDM) wavelength (also known as a lambda), or a physical port number. The labels can be derived from physical components of the network devices, such as interfaces.

There are two service models for GMPLS. Each model determines how much visibility a client node, such as a router, has into the optical core or transport network. The first model is 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.

To enable multilayer LSPs, GMPLS uses the following mechanisms:

- Separation of the control channel from the data channel—A new protocol called Link Management Protocol (LMP) is used to define and manage both control channels and data channels between GMPLS peers. Messages for GMPLS LSP setup are sent from one device to a peer device over an out-of-band control channel. Once the LSP setup is complete and the path is provisioned, the data channel is established and can be used to carry traffic. In GMPLS, the control channel is always separate from the data channel.
- RSVP-TE extensions for GMPLS—RSVP-TE was designed to signal the setup of packet LSPs only. The protocol has been extended to request path setup for nonpacket LSPs that use wavelengths, time slots, and fibers as potential labels.
- OSPF extensions for GMPLS—OSPF was designed to route packets to physical and logical interfaces related to a PIC. This protocol has been extended to route packets to virtual peer interfaces defined in an LMP configuration.
- Bidirectional LSPs—Unlike unidirectional LSP paths found in the standard, packet-based version of MPLS, data can travel both ways between GMPLS devices over a single LSP path. Nonpacket LSPs in GMPLS are bidirectional by default.

GMPLS is intended to bridge the gap between the traditional transport infrastructure and the IP layer. Since this protocol is supported by several network industry organizations and standards bodies, GMPLS is designed to enable multivendor interoperability and multilayer functionality. In the near future, routers will be able to make dynamic requests for extra bandwidth on demand from the optical network. Consequently, service providers envision GMPLS as a means to set up optical circuits and services dynamically instead of manually. Many industry professionals are cautiously optimistic regarding the advent of GMPLS, and Juniper Networks is pleased to continue its support for this protocol.

GMPLS System Requirements

To implement GMPLS Phase 2, your system must meet these minimum requirements:

- JUNOS Release 8.5 or later for GMPLS support of unnumbered links.
- JUNOS Release 8.1 or later for LMP control channels
- JUNOS Release 7.0 or later for graceful teardown of GMPLS LSPs and graceful restart of GMPLS neighbors
- JUNOS Release 5.6 or later for GMPLS Phase 2
- Two Juniper Networks M Series or T Series routers, and two optical cross-connects (OXCs) that support GMPLS

GMPLS Terms and Acronyms

C

control adjacency — A signaling path between peer devices in a GMPLS network that typically travels across virtual peer interfaces. Protocols are enabled on the control adjacency, which can have one or more associated control channels.

control channel — The actual interfaces where protocol packets are sent and received by GMPLS peers. If more than one control channel is configured, LMP selects which control channel is active.

F

Fiber-Switched Capable (FSC) — LSPs are switched between two fiber-based devices, such as optical cross-connects (OXCs), that operate at the level of individual fibers.

forwarding adjacency — A forwarding path for sending data between peer devices in a GMPLS network.

G

Generalized Multiprotocol Label Switching (GMPLS) — An extension to MPLS that allows data from multiple layers to be switched over label-switched paths (LSPs). GMPLS LSPs are possible between equivalent Layer 1, Layer 2, and Layer 3 devices. For more information about GMPLS and MPLS, see the *JUNOS MPLS Applications Configuration Guide*.

GMPLS label — A fiber port, TDM time slot, DWDM wavelength, or data packet identifier of a GMPLS-enabled device used as a next-hop identifier.

L

Lambda-Switched Capable (LSC) LSPs are switched between two DWDM devices, such as such as OXCs, that operate at the level of individual wavelengths.

Link Management Protocol (LMP) A GMPLS-related protocol defined in RFC 4204 that is used to define control adjacencies and forwarding adjacencies between peers and to maintain and allocate resources on traffic engineering links.

P

Packet-Switched Capable (PSC) LSPs are switched between two packet-based devices, such as routers or ATM switches.

T

TDM-Switched Capable (TDM) LSPs are switched between two TDM devices, such as SONET/SDH ADMs.

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 inherent attributes, such as encoding type, switching capability, and bandwidth. Each traffic engineering link represents a forwarding adjacency between a pair of devices.

GMPLS Phase 2 Implementation

The major changes between GMPLS Phase 1 and GMPLS Phase 2 are as follows:

- You must configure one or more control channels between peers when you configure LMP (in addition to the existing statements for LMP peers and traffic engineering links). The control channels must travel across a point-to-point link or tunnel. To configure a static control channel, include the control-channel statement at the [edit protocols link-management peer *peer-name*] hierarchy level. To configure a control channel that uses control channel management and link property correlation, include the lmp-control-channel statement at the [edit protocols link-management peer *peer-name*] hierarchy level.

NOTE: You can configure either the control-channel statement or the lmp-control-channel statement at the [edit protocols link-management peer *peer-name*] hierarchy level, but not both statements simultaneously.

- OSPF and RSVP have been extended to allow control adjacencies between peers using virtual peer interfaces. The peer interfaces are derived from LMP and can be used for the control adjacency between peers instead of the physical interfaces. To configure OSPF, include the peer-interface statement at the [edit protocols ospf area *area-number*] hierarchy level. To configure RSVP, include the peer-interface statement at the [edit protocols rsvp] hierarchy level. However, when you enable peer interfaces, you must disable RSVP and OSPF on all physical control channel

interfaces. Alternatively, you can omit the physical control channel interfaces when configuring these protocols.

- The Constrained Shortest Path First (CSPF) algorithm has been extended to permit use with nonpacket LSPs. In GMPLS Phase 2, the **no-cspf** statement can be omitted from the LSP configuration because it is no longer mandatory. When this statement is omitted, you must configure the signal type attribute for the LSP. For CSPF to work correctly, OSPF extensions for GMPLS need to be implemented on all devices in the GMPLS network.
- LSP paths now can be strict, loose, or dynamic for GMPLS LSPs because traffic engineering link information is now exchanged by OSPF. (GMPLS Phase 1 required strict LSP paths.)

JUNOS Software supports the following GMPLS functionality:

- Out-of-band signaling controls the setup of LSP paths, enabling a control plane that is separate from the data plane.
- RSVP-TE extensions support additional objects beyond Layer 3 packets, such as ports, time slots, and wavelengths.
- Link Management Protocol (LMP) creates and maintains a database of traffic engineering links, control channels, and peer information. Only the static version of this protocol is supported.
- Bidirectional LSPs are required between nonpacket GMPLS devices.
- Several GMPLS label types are defined in RFC 3471, *Generalized MPLS - Signaling Functional Description*. The MPLS, Generalized, SONET/SDH, Suggested, and Upstream label types are supported.
- Generalized labels do not contain a type field because the nodes are expected to know from the context of their connection what type of label to expect. For example, an encoding type, such as Ethernet or SONET/SDH, is determined by the resources in a traffic engineering link.
- Traffic parameters facilitate GMPLS bandwidth encoding and SONET/SDH formatting.
- Interface Identification/Errored Interface Identification, UNI-style signaling, and Secondary LSP paths are supported.
- Original channelized interfaces (such as channelized OC12 to DS3, channelized OC3 to T1, and channelized STM1 to E1) support GMPLS signaling.
- GMPLS graceful restart for RSVP LSP paths is supported.
- RSVP-TE over unnumbered links is supported.

The following functionality is *not* supported:

- Notify messages
- GMPLS routing extensions for IS-IS
- GMPLS link bundling
- Dynamic LMP

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 not to refer to GMPLS labels as "interfaces." To avoid confusion, refer to them as traffic engineering links and refer to the physical interfaces as resources.

GMPLS Operation

GMPLS requires close interaction between LMP, RSVP, and OSPF. The following sequence of events describes how GMPLS works:

1. LMP notifies RSVP and OSPF of the control peer, the control adjacency, and resources for the traffic engineering link.
2. GMPLS extracts the LSP attributes from the configuration and requests that RSVP signal one or more specific paths, 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 successfully finds a resource matching the attributes, label allocation succeeds. RSVP sends a PathMsg hop by hop until it reaches the target router.
4. The target router, on receiving the RSVP PathMsg, requests that LMP allocate a resource based on the signaled parameters. If label allocation succeeds, it sends back a ResvMsg.
5. If the signaling is successful, an optical path is provisioned.

Configuring GMPLS

To implement GMPLS, you must configure traffic link management protocol traffic engineering links and protocol peers and OSPF and RSVP peer interfaces, establish GMPLS LSP path information, define GMPLS paths, discover local identifiers and configure remote identifiers. This section contains these configuration procedures plus some optional configuration procedures:

- Configuring Link Management Protocol Traffic Engineering Links on page 10
- Configuring Link Management Protocol Peers on page 10
- Configuring Peer Interfaces in OSPF and RSVP on page 11
- Establishing GMPLS LSP Path Information on page 12
- Defining GMPLS Label-Switched Paths on page 12
- Displaying Local Identifiers and Configuring Remote Identifiers on page 13
- Option: Tearing Down GMPLS LSPs Gracefully on page 14

- Option: Allowing Nonpacket GMPLS LSPs to Establish a Path Through JUNOS-Based Routers on page 14
- Option: Selecting the Peer Model for GMPLS on page 15
- Option: Selecting the Overlay Model for GMPLS on page 15
- Option: GMPLS Graceful Restart on page 15
- Option: Configuring an LMP Control Channel on page 16
- Option: Configuring GMPLS Support for Unnumbered Links on page 18

Configuring Link Management Protocol Traffic Engineering Links

To begin your GMPLS configuration, enable LMP to define the data channel interconnection between devices at the [edit protocols link-management] hierarchy level.

To configure data channels in LMP, include the te-link *te-link-name* statement at the [edit protocols link-management] hierarchy level. Define all traffic engineering link options shown. (You will configure remote-id statements at the te-link and interface levels later.) We recommend that you use a different IP address and mask on your traffic engineering link addresses from the ones configured on your physical interfaces. This way, you can identify which addresses are physical and which ones belong to the traffic engineering link (TE link).

```
[edit]
protocols {
   link-management {
      te-link te-link-name { # Collection of physical ports or time slots.
         local-address ip-address; # Local IP address associated with the TE link.
         remote-address ip-address; # Remote IP address mapped to the TE link.
         interface interface-name { # Interface used for data transfer.
            local-address ip-address; # Local IP address for the TE link.
            remote-address ip-address; # Remote IP address for the TE link.
         }
      }
   }
}
```

Configuring Link Management Protocol Peers

After you set up traffic engineering links, configure LMP network peers with the peer statement at the [edit protocols link-management] hierarchy level. A peer is the network device that your router communicates with when setting up the control and data channels. Often, the peer is an OXC. Designate a peer name, configure the peer's router ID as the address (often a loopback address), specify the interface that will be used as a control channel, and apply the traffic engineering link to be associated with this peer.

You can configure one or more control channels for a peer. The control channels must have point-to-point connectivity with the peer (for example, you can use a point-to-point link or a tunnel). You can also configure the generic routing

encapsulation (GRE) tunnel interface of the Routing Engine as a control channel. To configure a static control channel, include the control-channel statement at the [edit protocols link-management peer *peer-name*] hierarchy level. To configure a control channel that uses control channel management and link property correlation, see "Option: Configuring an LMP Control Channel" on page 16. Without 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.
         control-channel interface; # Specify the interface for the control channel.
         te-link te-link-name; # Assign a TE link to this peer.
      }
   }
}
```

NOTE: Although you can configure the gre- tunnel interface on a Routing Engine as a control channel, this interface is not supported, nor is it configurable for other applications.

NOTE: You can configure either the control-channel statement or the lmp-control-channel statement at the [edit protocols link-management peer *peer-name*] hierarchy level, but not both statements simultaneously.

Configuring Peer Interfaces in OSPF and RSVP

After you establish LMP peers, add peer interfaces to OSPF and RSVP. A peer interface is a virtual interface used to support a control adjacency between two peers. OSPF and RSVP form adjacencies between peers by using the peer interfaces instead of the physical interfaces.

Because actual protocol packets are sent and received by peer interfaces, the peer interfaces can be signaled and advertised to peers like any other interface enabled for RSVP and OSPF. The peer interface name must match the peer name configured in LMP. To configure RSVP signaling for LMP peers, include the peer-interface statement at the [edit protocols rsvp] hierarchy level. To configure OSPF routing for LMP peers, include the peer-interface statement at the [edit protocols ospf area *area-number*] 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.
```

```
        }
      }
    }
  }
}
```

NOTE: When adding the virtual peer interfaces to RSVP and OSPF, do not configure the corresponding physical control channel interface in either protocol. If the interface all option is used, you must disable the protocols manually on the control channel interface.

- To disable OSPF, use the disable statement at the [edit protocols ospf area *area-number* interface *interface-name*] hierarchy level.
- To disable RSVP, use the disable statement at the [edit protocols rsvp interface *interface-name*] hierarchy level.

Establishing GMPLS LSP Path Information

When you configure LSP paths for GMPLS, 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);
      }
   }
}
```

Defining GMPLS Label-Switched Paths

Next, define LSP attributes at the [edit protocols mpls label-switched-path] hierarchy level. To enable the proper GMPLS switching parameters, configure the attributes appropriate for your network connection. The default values, which are also appropriate for standard MPLS, are ipv4 for gpid, none for signal-bandwidth, and psc-1 for switching-type.

NOTE: In JUNOS Release 5.6 and later, the signal-bandwidth statement replaces the signal-type statement. Also, virtual tributary (VT) 1.5 and 2.0 SONET/SDH bandwidth options are available at the [edit protocols mpls label-switched-path lsp-name lsp-attributes signal-bandwidth] hierarchy level.

```
[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.
         lsp-attributes { # Attributes determine the selection of an LSP.
            gpid type;  # Payload type, such as Ethernet or PPP.
            signal-bandwidth type; # Bandwidth encoding, such as DS3 or STM1.
            switching-type type; # Switching method, such as psc-1 or lambda.
         }
      }
   }
}
```

NOTE: 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. 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 nonpacket switching type option, such as lambda or fiber, at the [edit mpls label-switched-path *lsp-name* lsp-attributes] hierarchy level.

Displaying Local Identifiers and Configuring Remote Identifiers

Once LMP is enabled on a router, the router automatically assigns two local IDs: one at the te-link level, the other at the interface level. You must configure these port-to-label mappings manually for LMP on both the router and its peer. To configure the mapping, set the local IDs of one device (such as the router) as remote IDs on the peer device (such as an OXC) with the remote-id statement at the [edit protocols link-management te-link *te-link-name*] and [edit protocols link-management te-link *te-link-name* interface *interface-name*] hierarchy levels.

You can view the traffic engineering link and interface local IDs by using the show link-management te-link command. Once you have learned these IDs, configure them as remote-id statements at the corresponding te-link and interface levels on the peer.

Because peers vary, check with your OXC vendor for the configuration statements and location of the local ID information for your specific optical peer device. If you do not manage the peer device, ask the peer's administrator to enable LMP and generate the IDs for you. GMPLS will not work unless these local IDs from both the router and the peer are configured as remote IDs on the opposite device.

To disable an entire traffic engineering link for administrative purposes, include the disable statement at the [edit protocols link-management te-link *te-link-name*] hierarchy level. To disable an interface within a traffic engineering link, include the disable

statement at the [edit protocols link-management te-link *te-link-name* interface *interface-name*] hierarchy level.

```
[edit]
protocols {
   link-management {
      te-link te-link-name {
         disable; # Disable the entire TE link.
         remote-id id-number; # TE link ID number of the peer device.
         interface interface-name { # Name of the interface used for data transfer.
            disable; # Disable an interface in the TE link.
            remote-id id-number; # ID number of the remote device.
         }
      }
   }
}
```

Option: Tearing Down GMPLS LSPs Gracefully

You can tear down a nonpacket GMPLS 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 a GMPLS LSP 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. To configure the timeout, include 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 command.

Option: Allowing Nonpacket GMPLS LSPs to Establish a Path Through JUNOS-Based Routers

When you configure a nonpacket LSP as administratively down, an external device (not a router running JUNOS Software) can either perform a Layer 1 path setup test or help bring up an optical cross-connect through a router running JUNOS Software.

To configure a nonpacket LSP as administratively down, you must set the A-bit in the ADMIN_STATUS field of a RSVP packet. This feature does not affect control path setup or data forwarding for packet LSPs.

To configure the ADMIN_STATUS field for a GMPLS LSP RSVP packet, issue the admin-down statement at the [edit protocols mpls label-switched-path *lsp-name*] hierarchy level or the [edit logical-systems *logical-system-name* protocols mpls label-switched-path *lsp-name*] hierarchy level.

Option: Selecting the Peer Model for GMPLS

To implement the peer model, perform the following configuration tasks:

- Use the default CSPF calculation that is inherent in MPLS LSPs.
- Include a traffic engineering link and a control channel within the peer statement at the [edit protocols link-management] hierarchy level.
- Reference the peer-interface in both OSPF and RSVP.
- Configure OSPF and OSPF traffic engineering to connect with the OXC.
- Optionally, define a GMPLS LSP path.

Option: Selecting the Overlay Model for GMPLS

To implement the overlay model, perform the following configuration tasks:

- Disable CSPF by including the no-cspf statement at the [edit protocols mpls label-switched-path *lsp-name*] hierarchy level.
- Define a strict GMPLS LSP data path to the remote OXC peer.
- Include a traffic engineering link and a control channel within the peer statement at the [edit protocols link-management] hierarchy level.
- Reference the peer-interface in RSVP.

Option: GMPLS Graceful Restart

GMPLS supports graceful restart, a mechanism that allows a restarting router to continue forwarding packets to neighbors without interruption. The restarting router relies on its forwarding table temporarily while it receives updates from “helper” neighbors that assist the restarting router in rebuilding its routing table.

To enable graceful restart for all routing protocols including GMPLS, include the graceful-restart statement at the [edit routing-options] hierarchy level. To disable graceful restart for GMPLS and RSVP, include the disable statement at the [edit protocols rsvp graceful-restart] hierarchy level. To disable GMPLS and RSVP graceful restart helper capability, include the helper-disable statement at the [edit protocols rsvp graceful-restart] hierarchy level.

To configure the maximum amount of time the routing platform retains information for restarting neighbors, include the maximum-helper-recovery-time statement at the

[edit protocols rsvp graceful-restart] hierarchy level. The default helper recovery time is 180 seconds, with a minimum value of 1 second and a maximum value of 3600 seconds. To view the current value configured for the helper recovery time, issue the show rsvp version operational mode command.

To configure the maximum amount of time the routing platform waits until a neighbor is declared dead, include the maximum-helper-restart-time statement at the [edit protocols rsvp graceful-restart] hierarchy level. The default helper restart time is 20 seconds, with a minimum value of 1 second and a maximum value of 1800 seconds. To view the current value configured for the helper restart time, issue the show rsvp version operational command.

```
[edit]
protocols {
    rsvp {
        graceful-restart {
            disable;
            helper-disable;
            maximum-helper-recovery-time seconds;
            maximum-helper-restart-time seconds;
        }
    }
}
```

For more information about graceful restart, see the *JUNOS High Availability Configuration Guide*.

Option: Configuring an LMP Control Channel

In contrast with statically configured control channels that provide basic connectivity, LMP control channels provide control channel management and link property correlation as defined in RFC 4204, *Link Management Protocol (LMP)*. To manage the state of the control channel, LMP uses the following sequence of events:

1. The LMP peer with the highest router ID sends configuration messages to peers.
2. When a peer sends an acknowledgement of a Config message back to the originator, the control channel enters the active state.
3. When a peer sends LMP hello messages and receives them from a neighbor, the control channel transitions to the up state.
4. Once the control channel is up, link summary messages and acknowledgements are sent between LMP peers to share traffic engineering link information.

To configure an LMP control channel, include the lmp-control-channel statement at the [edit protocols link-management peer *peer-name*] hierarchy level and specify the physical IP address of the peer as the remote address. You can also specify a hello interval, a hello dead interval, a retransmission interval, and a retry limit. Default values for these statements are shown in Table 3 on page 17.

Table 3: Default Values for LMP Protocol Fields

LMP Protocol Field	Value
Hello interval	150 milliseconds
Hello dead interval	500 milliseconds
Retransmission interval	500 milliseconds
Retry limit	3 retries

If you plan to use these default values, you do not need to configure them. However, if you choose to manually configure any of these values, you must include all four statements (hello-interval, hello-dead-interval, retransmission-interval, and retry-limit) at the [edit protocols link-management peer *peer-name* lmp-protocol] hierarchy level. Also, the hello dead interval must be at least three times larger than the hello interval.

If you do not want the routing platform to initiate LMP negotiations, include the passive statement at the [edit protocols link-management peer *peer-name* lmp-protocol] hierarchy level. To log hello packets, other LMP messages, and state transitions of the control channels and traffic engineering links, include the hello-packets, packets, and state traceoptions flags at the [edit protocols link-management traceoptions] hierarchy level.

```
[edit]
protocols {
   link-management {
      peer peer-name { # Configure the name of your network peer.
         address ip-address; # Include the router ID of the peer.
         lmp-control-channel interface { # Specify the control channel interface.
            remote-address ip-address; # Configure the peer's physical IP address.
         }
         lmp-protocol { # Manually configure LMP protocol values.
            hello-dead-interval milliseconds;
            hello-interval milliseconds;
            passive; # Respond to LMP peers, but do not initiate LMP negotiations.
            retransmission-interval milliseconds;
            retry-limit number;
         }
         te-link te-link-name; # Assign a TE link to this peer.
      }
      traceoptions {
         flag (hello-packets | packets | state);
      }
   }
}
```

For a full example of an LMP protocol control channel configuration, see "Example: LMP Control Channel Configuration" on page 31.

NOTE: You can configure either the control-channel statement or the lmp-control-channel statement at the [edit protocols link-management peer *peer-name*] hierarchy level, but not both statements simultaneously.

Option: Configuring GMPLS Support for Unnumbered Links

The JUNOS Software supports RSVP traffic engineering over unnumbered interfaces. With this feature, you do not have to configure IP addresses for each interface participating in the RSVP-signaled network. 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)*. Unnumbered links can also be specified in the MPLS traffic engineering signaling, as described in RFC 3477, *Signaling Unnumbered Links in Resource ReSerVation Protocol - Traffic Engineering (RSVP-TE)*.

To configure RSVP for unnumbered interfaces, you must configure a router ID using the router-id statement specified at the [edit routing-options] hierarchy level. The router ID must be routable (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. In addition to the router ID, we recommend that you configure a secondary interface on the loopback:

```
[edit]
routing-options {
    router-id address;
}
```

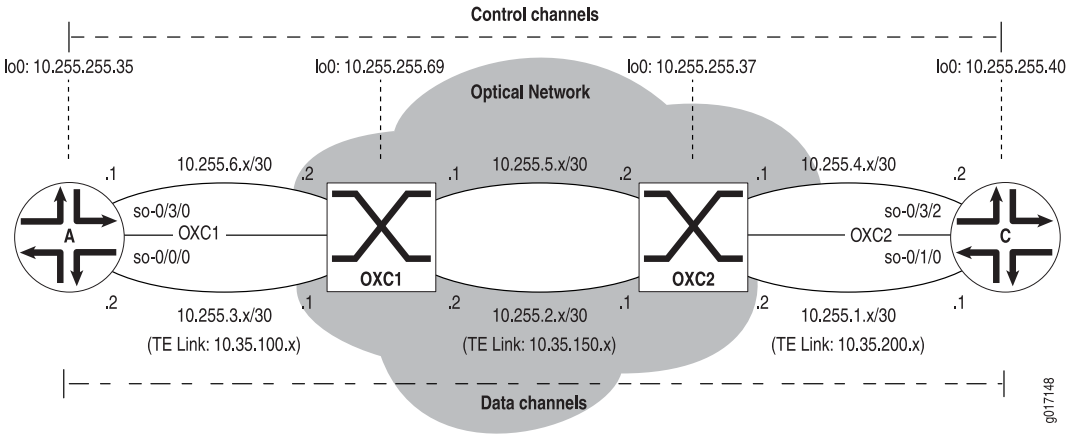
GMPLS Configuration Examples

This section contains configuration examples and commands you can issue to verify your GMPLS configuration:

- Example: GMPLS Configuration on page 19
- Example: Configuring Traffic Engineering Link and Interface Identifiers on page 30
- Example: LMP Control Channel Configuration on page 31

Example: GMPLS Configuration

Figure 2: GMPLS Topology Diagram



In Figure 2 on page 19, a control channel is established between Router A and OXC1, OXC1 and OXC2, and OXC2 and Router C. A data channel is enabled on a second connection between each pair of devices. The optical network cloud can contain OXCs, ADMs, or other lower-layer devices. In this example, OXC1 and OXC2 are in the direct data path between Routers A and C and the two OXCs have point-to-point connectivity with each other and the directly connected peer routers.

Starting with Router A, configure LMP traffic engineering links and peers to create a data channel and a control channel to connect with OXC1. To differentiate the logical traffic engineering link from the physical network, the local and remote addresses in the traffic engineering link are not related to the IP addresses assigned to the physical interfaces.

When you enable LMP peering on both Router A and OXC1, include the control channel interface as one of the peer statements. Use the name of the peer (in this case, oxc1) as the peer interface name when you add the peer-interface statement to RSVP at the [edit protocols rsvp] hierarchy level and OSPF at the [edit protocols ospf area *area-number*] hierarchy level.

The peer-interface statement adds the remote address and local address from your LMP configuration into the routing and signaling processes activated between Router A and OXC1. Make sure the physical control channel is a point-to-point link and has some form of IP reachability through static routes, an interior gateway protocol (IGP), or BGP (this example uses OSPF). Another way to achieve point-to-point links, especially if there are multiple hops between peers, is to use a generic routing encapsulation (GRE) tunnel for the control channel.

Next, configure an MPLS LSP on Router A to reach Router C. For this example, assume your data plane connection uses STM1 and Point-to-Point Protocol (PPP) over a fiber-switched network. Configure these LSP attributes in the LSP. Because this LSP does not use packet switching, a bidirectional LSP is enabled by default. As a result, you do not need to configure a return path LSP on Router C.

Finally, remember to discover the local IDs and configure them on OXC1 with the remote-id statement at the [edit protocols link-management te-link *te-link-name*] and [edit protocols link-management te-link *te-link-name* interface] hierarchy levels. For Router A, use the command show link-management te-link to find Router A's two local IDs (te-link and interface); then configure these IDs as remote IDs on OXC1 at the equivalent hierarchy levels.

Router A

```
[edit]
interfaces {
   so-0/0/0 {
      description "Data channel to OXC1";
      encapsulation ppp;
      unit 0 {
         family inet {
            address 10.255.3.2/30 {
               destination 10.255.3.1;
            }
         }
         family mpls;
      }
   }
   so-0/3/0 {
      description "Control channel to OXC1";
      encapsulation ppp;
      unit 0 {
         family inet {
            address 10.255.6.1/30 {
               destination 10.255.6.2;
            }
         }
         family mpls;
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.255.255.35/32;
         }
      }
   }
}
protocols
rsvp {
   interface all;
   interface so-0/3/0.0 {
      disable;
   }
   peer-interface oxc1;
}
mpls {
   label-switched-path gmpls-lsp1 {
      to 10.255.255.40;
      lsp-attributes {
         signal-bandwidth stm-1;
         switching-type fiber;
         gpid ppp;
```

```
        }
        primary path-lsp1;
    }
    path path-lsp1 {
        10.35.100.1 strict; # This example does not disable CSPF,
        10.35.150.1 strict; # so this step is optional.
        10.35.200.1 strict;
    }
    interface all;
}
ospf {
    area 0.0.0.0 {
        interface lo0.0;
        interface fxp0.0 {
            disable;
        }
        peer-interface oxc1;
    }
}
link-management {
    te-link te-oxc1 {
        local-address 10.35.100.2;
        remote-address 10.35.100.1;
        remote-id 8256;
        interface t3-3/3/0:0 {
            local-address 10.35.100.2;
            remote-address 10.35.100.1;
            remote-id 65536;
        }
    }
    peer oxc1 {
        address 10.255.255.69;
        control-channel so-0/3/0.0;
        te-link te-oxc1;
    }
}
```

On OXC1, complete your configuration of the control channel and the traffic engineering link data channel to Router A. Refer to your OXC vendor's instructions to configure a traffic engineering link on your specific device. Enable LMP peering, configure Router A's local IDs as remote IDs on OXC1, and discover OXC1's local IDs. Finally, configure OXC1's local IDs as remote IDs on Router A.

In the optical network between your OXCs, configure a traffic engineering link and a control channel between OXC1 and OXC2. Refer to the OXC vendor's instructions to configure this link. For the example shown in Figure 2 on page 19, you can assume a traffic engineering link with an address space of 10.255.150.x/30 has been enabled over a physical network with IP addresses 10.255.2.x/30. Also, a control channel has been created over the 10.255.4.x/30 link.

On OXC2, configure a traffic engineering link to Router A. Refer to your OXC vendor's instructions to configure this traffic engineering link on your device. Enable LMP peering, configure Router C's local IDs as remote IDs on OXC2, and discover OXC2's local IDs. Finally, configure OXC2's local IDs as remote IDs on Router C.

Now you are ready to complete this GMPLS example. On Router C, set up your traffic engineering link, LMP peer, and control channel statements to connect to OXC2. As with Router A, the local and remote addresses in the traffic engineering link on Router C are not related to the IP addresses assigned to the physical interface.

Next, configure RSVP, MPLS, and OSPF to match the control channel protocols you configured on Router A. You do not need to set up an LSP on Router C because Router A's nonpacket LSP is bidirectional by default. Also, because RSVP is enabled for all interfaces and you are using a peer interface, you must disable RSVP on the physical control channel interface so-0/3/2.

After you enable LMP on both Router C and OXC2, discover the local IDs and configure them as remote IDs on OXC2. For Router C, use the command show link-management te-link to discover Router C's two local IDs (te-link and interface); then configure these IDs as remote IDs on OXC2 at the equivalent hierarchy levels.

Router C

```
[edit]
interfaces {
   so-0/3/2 {
      description “Control channel to OXC2”;
      unit 0 {
         family inet {
            address 10.255.4.2/30 {
               destination 10.255.4.1;
            }
         }
         family mpls;
      }
   }
   so-0/1/0 {
      description “Data channel to OXC2”;
      encapsulation ppp;
      unit 0 {
         family inet {
            address 10.255.1.1/30 {
               destination 10.255.1.2;
            }
         }
         family mpls;
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.255.255.40/32;
         }
      }
   }
}
protocols
rsvp {
   interface all;
   interface so-0/3/2.0 {
      disabled;
   }
   peer-interface oxc2;
```

```
}
mpls {
    interface all;
}
ospf {
    area 0.0.0.0 {
        interface fxp0.0 {
            disable;
        }
        interface lo0.0;
        peer-interface oxc2;
    }
}
link-management {
    te-link te-oxc2 {
        local-address 10.35.200.1;
        remote-address 10.35.200.2;
        remote id 41060;
        interface so-0/1/0 {
            local-address 10.35.200.1;
            remote-address 10.35.200.2;
            remote-id 22278;
        }
    }
    peer oxc2 {
        address 10.255.255.37;
        control-channel so-0/3/2.0;
        te-link te-oxc2;
    }
}
```

Verifying Your Work

To verify proper operation of GMPLS, you can use the following commands:

- show link-management (te-link | peer)
- show link-management routing (te-link | peer)
- show mpls lsp (bidirectional | unidirectional)
- show mpls lsp (detail | extensive)
- show ospf interface
- show ospf neighbor
- show rsvp interface link-management
- show rsvp session (bidirectional | unidirectional)
- show rsvp session te-link
- show rsvp session detail
- show rsvp neighbor detail

- show ted database extensive
- traceroute (using the lsp flag with RSVP protocol–level trace options)

The following sections show the output of these commands used with the configuration example:

- Router A Status on page 24
- Router C Status on page 29

Router A Status

After you enter the local-address, remote-address, and interface parameters in traffic engineering link te-oxc1 and commit the changes, the router automatically creates a local ID at the te-link and interface levels of the [edit protocols link-management] hierarchy. To view these IDs, issue the show link-management te-link command.

```
user@RouterA> show link-management te-link
 TE link name: te-oxc1 , State: Up
  Local identifier: 8255, Remote identifier: 0 , Local address: 10.35.100.2,
Remote address: 10.35.100.1, Encoding: SDH/SONET,
  Minimum bandwidth: 155.52Mbps, Maximum bandwidth: 155.52Mbps, Total bandwidth:
 155.52Mbps, Available bandwidth: 0bps
   Name          Local ID  Remote ID      Bandwidth In use    LSP
   so-0/0/0         65535          0     155.52Mbps No
```

Once you find these values on Router A, configure them as remote IDs at the same hierarchy levels on OXC1. In this example, 8255 is Router A's local traffic engineering link ID (configure this as the traffic engineering link remote-ID on OXC1) and 65535 is Router A's local interface ID (configure this as the interface remote-ID on OXC1).

After you configure both remote IDs on both peers, the GMPLS traffic engineering links should work. Using the same command as before, you can verify whether the link is functional, with both remote and local IDs in place:

```
user@RouterA> show link-management te-link
 TE link name: te-oxc1, State: Up
  Local identifier: 8255, Remote identifier: 8256, Local address: 10.35.100.2,
Remote address: 10.35.100.1, Encoding: SDH/SONET,
  Minimum bandwidth: 155.52Mbps, Maximum bandwidth: 155.52Mbps, Total bandwidth:
 155.52Mbps, Available bandwidth: 0bps
   Name          Local ID  Remote ID      Bandwidth In use    LSP
   so-0/0/0         65535      65536     155.52Mbps Yes       gmpls-lsp1
```

To further verify proper operation, use the following commands:

```
user@RouterA> show link-management routing peer
Peer name: oxc1, System identifier: 13892
 State: Up, Control address: 10.255.255.69
   Control-channel                  State
   so-0/3/0.0                       Active

user@RouterA> show link-management routing te-link
 TE link name: te-oxc1, State: Up
  Local identifier: 8255, Remote identifier: 8256, Local address: 10.35.100.2,
```

```
Remote address: 10.35.100.1, Encoding: SDH/SONET,
  Minimum bandwidth: 155.52Mbps, Maximum bandwidth: 155.52Mbps, Total bandwidth:
 155.52Mbps, Available bandwidth: 0bps

user@RouterA> show link-management peer
Peer name: oxc1, System identifier: 13892
 State: Up, Control address: 10.255.255.69
   Control-channel                  State
   so-0/3/0.0                       Active
  TE links:
   te-oxc1

user@RouterA>  show mpls lsp bidirectional
Ingress LSP: 1 sessions
To              From            State Rt ActivePath       P     LSPname
10.255.255.40   10.255.255.35   Up     0 path-lsp1        *     gmpls-lsp1 Bidir
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

user@RouterA>  show mpls lsp bidirectional extensive
Ingress LSP: 1 sessions

10.255.255.40
  From: 10.255.255.35, State: Up, ActiveRoute: 0, LSPname: gmpls-lsp1
  Bidirectional
  ActivePath: path-lsp1 (primary)
  LoadBalance: Random
  Signal type: STM-1
  Encoding type: SDH/SONET, Switching type: Fiber, GPID: PPP
 *Primary   path-lsp1             State: Up
    Bandwidth: 155.52Mbps
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 2)
          10.35.100.1 S 10.35.150.1 S 10.35.200.1 S
    Received RRO:
          10.35.100.1 10.35.150.1 10.35.200.1
    7 Nov  7 15:47:11  Selected as active path
    6 Nov  7 15:47:11  Record Route:  10.35.100.1 10.35.150.1 10.35.200.1
    5 Nov  7 15:47:11  Up
    4 Nov  7 15:47:11  Update LSP Encoding Type
    3 Nov  7 15:47:11  Originate Call
    2 Nov  7 15:47:11  CSPF: computation result accepted
    1 Nov  7 15:46:41  CSPF failed: no route toward 10.255.255.40
  Created: Thu Nov  7 15:46:38 2002
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
```

If you configure an LMP peer interface in OSPF, you can see that this virtual interface is treated as a point-to-point link. To view this, use the show ospf interface command.

```
user@RouterA> show ospf interface
```

```
Interface           State     Area            DR ID           BDR ID       Nbrs
lo0.0               DR       0.0.0.0         10.255.255.35   0.0.0.0        0
oxc1                PtToPt   0.0.0.0         0.0.0.0         0.0.0.0        1
```

The next command is useful because it indicates whether RSVP is disabled on the control channel. It also shows the state of the reservations on the traffic engineering links.

```
user@RouterA> show rsvp interface link-management
RSVP interface: 1 active
oxc1  State Up
Active control channel: so-0/3/0.0  RSVP disabled
  TElink: te-oxc1, Local identifier: 8255
  ActiveResv 1, PreemptionCnt 0
  StaticBW: 155.52Mbps, ReservedBW: 155.52Mbps, AvailableBW: 0bps

user@RouterA>  show rsvp session detail
Ingress RSVP: 1 sessions

10.255.255.40
  From: 10.255.255.35, LSPstate: Up, ActiveRoute: 0
  LSPname: gmpls-lsp1, LSPpath: Primary
  Bidirectional, Upstream label in: 27676, Upstream label out: -
  Suggested label received: -, Suggested label sent: 27676
  Recovery label received: -, Recovery label sent: 60444
  Resv style: 1 FF, Label in: -, Label out: 60444
  Time left:    -,  Since: Thu Nov  7 15:47:11 2002
  Tspec: rate 0bps size 0bps peak 1.544Mbps m 20 M 1500
  Port number: sender 1 receiver 17 protocol 0
  PATH rcvfrom: localclient
  PATH sentto: 10.255.255.40 (oxc1) 157 pkts
  RESV rcvfrom: 10.255.255.40  (oxc1) 71 pkts
  Explct route: 10.35.100.1 10.35.150.1 10.35.200.1
  Record route: <self>  10.35.100.1 10.35.150.1 10.35.200.1
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@RouterA>  show rsvp session bidirectional
Ingress RSVP: 1 sessions
To              From            State Rt Style Labelin Labelout LSPname
10.255.255.40   10.255.255.35   Up     0  1 FF       -    60444 gmpls-lsp1 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@RouterA>  show rsvp session te-link te-oxc1
Ingress RSVP: 1 sessions
To              From            State Rt Style Labelin Labelout LSPname
10.255.255.40   10.255.255.35   Up     0  1 FF       -    60444 gmpls-lsp1 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@RouterA>  show ted database extensive
TED database: 0 ISIS nodes 4 INET nodes
NodeID: 10.255.255.35
  Type: Rtr, Age: 2178 secs, LinkIn: 4, LinkOut: 5
  Protocol: OSPF(0.0.0.0)
    To: 10.255.255.69, Local: 10.35.100.2, Remote: 10.35.100.1
      Metric: 1
      Static BW: 155.52Mbps
      Reservable BW: 155.52Mbps
      Available BW [priority] bps:
       [0] 0bps         [1] 0bps         [2] 0bps         [3] 0bps
       [4] 0bps         [5] 0bps         [6] 0bps         [7] 0bps
      Interface Switching Capability Descriptor(1):
        Switching type: Fiber
        Encoding type: SDH/SONET
        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
        Minimum LSP BW: 155.52Mbps
        Interface MTU: 2595
NodeID: 10.255.255.37
  Type: Rtr, Age: 2852 secs, LinkIn: 5, LinkOut: 5
  Protocol: OSPF(0.0.0.0)
    To: 10.255.255.69, Local: 10.35.150.1, Remote: 10.35.150.2
      Metric: 1
      Static BW: 622.08Mbps
      Reservable BW: 622.08Mbps
      Available BW [priority] bps:
       [0] 622.08Mbps   [1] 622.08Mbps   [2] 622.08Mbps   [3] 622.08Mbps
       [4] 622.08Mbps   [5] 622.08Mbps   [6] 622.08Mbps   [7] 622.08Mbps
      Interface Switching Capability Descriptor(1):
        Switching type: Fiber
        Encoding type: SDH/SONET
        Maximum LSP BW [priority] bps:
         [0] 622.08Mbps   [1] 622.08Mbps   [2] 622.08Mbps   [3] 622.08Mbps
         [4] 622.08Mbps   [5] 622.08Mbps   [6] 622.08Mbps   [7] 622.08Mbps
        Minimum LSP BW: 622.08Mbps
        Interface MTU: 2597
    To: 10.255.255.40, Local: 10.35.200.2, Remote: 10.35.200.1
      Metric: 1
      Static BW: 155.52Mbps
      Reservable BW: 155.52Mbps
      Available BW [priority] bps:
       [0] 0bps         [1] 0bps         [2] 0bps         [3] 0bps
       [4] 0bps         [5] 0bps         [6] 0bps         [7] 0bps
      Interface Switching Capability Descriptor(1):
        Switching type: Fiber
        Encoding type: SDH/SONET
        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
        Minimum LSP BW: 155.52Mbps
        Interface MTU: 2600
NodeID: 10.255.255.40
  Type: Rtr, Age: 2854 secs, LinkIn: 2, LinkOut: 2
```

```
  Protocol: OSPF(0.0.0.0)
    To: 10.255.255.37, Local: 10.35.200.1, Remote: 10.35.200.2
      Metric: 1
      Static BW: 155.52Mbps
      Reservable BW: 155.52Mbps
      Available BW [priority] bps:
       [0] 0bps         [1] 0bps         [2] 0bps         [3] 0bps
       [4] 0bps         [5] 0bps         [6] 0bps         [7] 0bps
      Interface Switching Capability Descriptor(1):
        Switching type: Fiber
        Encoding type: SDH/SONET
        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
        Minimum LSP BW: 155.52Mbps
        Interface MTU: 2600
NodeID: 10.255.255.69
  Type: Rtr, Age: 2832 secs, LinkIn: 8, LinkOut: 7
  Protocol: OSPF(0.0.0.0)
    To: 10.255.255.35, Local: 10.35.100.1, Remote: 10.35.100.2
      Metric: 1
      Static BW: 155.52Mbps
      Reservable BW: 155.52Mbps
      Available BW [priority] bps:
       [0] 0bps         [1] 0bps         [2] 0bps         [3] 0bps
       [4] 0bps         [5] 0bps         [6] 0bps         [7] 0bps
      Interface Switching Capability Descriptor(1):
        Switching type: Fiber
        Encoding type: SDH/SONET
        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
        Minimum LSP BW: 155.52Mbps
        Interface MTU: 2595
    To: 10.255.255.37, Local: 10.35.150.2, Remote: 10.35.150.1
      Metric: 1
      Static BW: 622.08Mbps
      Reservable BW: 622.08Mbps
      Available BW [priority] bps:
       [0] 622.08Mbps   [1] 622.08Mbps   [2] 622.08Mbps   [3] 622.08Mbps
       [4] 622.08Mbps   [5] 622.08Mbps   [6] 622.08Mbps   [7] 622.08Mbps
      Interface Switching Capability Descriptor(1):
        Switching type: Fiber
        Encoding type: SDH/SONET
        Maximum LSP BW [priority] bps:
         [0] 622.08Mbps   [1] 622.08Mbps   [2] 622.08Mbps   [3] 622.08Mbps
         [4] 622.08Mbps   [5] 622.08Mbps   [6] 622.08Mbps   [7] 622.08Mbps
        Minimum LSP BW: 622.08Mbps
        Interface MTU: 2597

user@RouterA>  show rsvp neighbor detail
RSVP neighbor: 1 learned
Address: 10.255.255.40   via: oxc1    status: Up
  Last changed time: 50:52, Idle: 0 sec, Up cnt: 1, Down cnt: 0
  Message received: 145
  Hello: sent 338, received: 338, interval: 9 sec
  Remote instance: 0x643087c7, Local instance: 0x3271e0a4
  Refresh reduction:  not operational
  Link protection:  disabled
    Bypass LSP: does not exist,  Backup routes: 0,  Backup LSPs: 0
```

Router C Status

After you enter the local-address, remote-address, and interface parameters in traffic engineering link te-oxc2 and commit the changes, the router automatically creates a local ID at the te-link and interface levels of the [edit protocols link-management] hierarchy. To view these IDs, issue the show link-management te-link command.

```
user@RouterC> show link-management te-link
  TE link name: te-oxc2,  State: Up
   Local identifier: 41059, Remote identifier: 0,  Local address: 10.35.200.1,
Remote address: 10.35.200.2, Encoding: SDH/SONET,
  Minimum bandwidth: 155.52Mbps, Maximum bandwidth: 155.52Mbps, Total bandwidth:
 155.52Mbps, Available bandwidth: 0bps
   Name          Local ID  Remote ID      Bandwidth In use    LSP
   so-0/1/0         22277          0     155.52Mbps No
```

Once you see what these values are, configure them as remote IDs at the same hierarchy levels on OXC2 where you found them on Router C. In this example, 41059 is Router C's local traffic engineering link ID (configure this as the traffic engineering link remote-ID on OXC2) and 22277 is Router C's local interface ID (configure this as the interface remote-ID on OXC2).

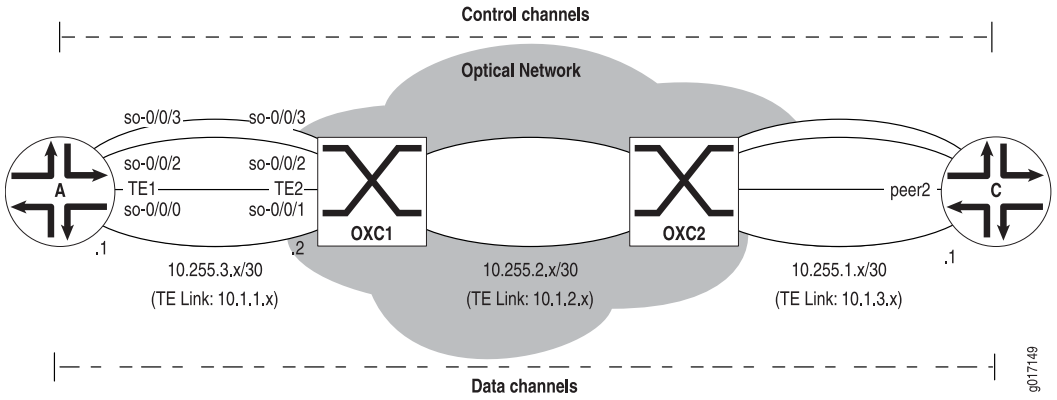
After you configure both remote IDs on both peers, the GMPLS traffic engineering links should work. Using the same command as before, you can determine whether the link is functional, with both remote and local IDs in place:

```
user@RouterC> show link-management te-link
 TE link name: te-oxc2, State: Up
  Local identifier: 41059, Remote identifier: 41060, Local address: 10.35.200.1,
 Remote address: 10.35.200.2, Encoding: SDH/SONET,
  Minimum bandwidth: 155.52Mbps, Maximum bandwidth: 155.52Mbps, Total bandwidth:
 155.52Mbps, Available bandwidth: 0bps
   Name          Local ID  Remote ID      Bandwidth In use    LSP
   so-0/1/0         22277      22278     155.52Mbps Yes       gmpls-lsp1
```

The other show commands operate like those in "Router A Status" on page 24.

Example: Configuring Traffic Engineering Link and Interface Identifiers

Figure 3: Traffic Engineering Link and Interface ID Example



Router A Details
TE link: 10.1.1.1
Peer name: TE1
te-link Local ID: 12345 *
interface Local ID: 23456 *
te-link Remote ID: 98765 **
interface Remote ID: 54321 **
Control channels:
so-0/0/2 and so-0/0/3

OXC1 Details
TE link: 10.1.1.2
Peer name: TE2
te-link Local ID: 98765 *
interface Local ID: 54321 *
te-link Remote ID: 12345 **
interface Remote ID: 23456 **
Control channels:
so-0/0/2 and so-0/0/3

* Automatically generated by the device when the TE link is enabled
** Configured by the user after the TE link is enabled

Figure 3 on page 30 shows where the IDs come from and where you must assign them. This example highlights the connections between Router A and OXC1, but the same configuration concepts apply to all pairs of peers.

First, you configure a traffic engineering link named TE1 on Router A, which contains the local address 10.1.1.1, remote address 10.1.1.2, data channel interface so-0/0/0, and control channel interfaces so-0/0/2 and so-0/0/3. You also configure a traffic engineering link named TE2 on OXC1, which contains the local address 10.1.1.2, remote address 10.1.1.1, data channel interface so-0/0/1, and control channel interfaces so-0/0/2 and so-0/0/3. When the traffic engineering links are enabled on Router A and OXC1, these two peer devices each generate two local IDs: one for the traffic engineering link itself and one for the logical interface.

If Router A has a local ID of 12345 for its traffic engineering link and a local ID of 23456 for its interface, you must configure 12345 as the traffic engineering link remote-ID and 23456 as the interface remote-ID on the TE2 traffic engineering link of OXC1. Similarly, if OXC1 has local IDs of 98765 for its traffic engineering link and 54321 for its interface, you configure Router A's TE1 traffic engineering link with 98765 as the traffic engineering link remote-ID and 54321 as the interface remote-ID.

To complete the full data path, you need to enable LMP on each link in the path. This means you must configure remote-ID and local-ID pairs between linked devices.

Example: LMP Control Channel Configuration

Figure 4: LMP Control Channel Topology Diagram

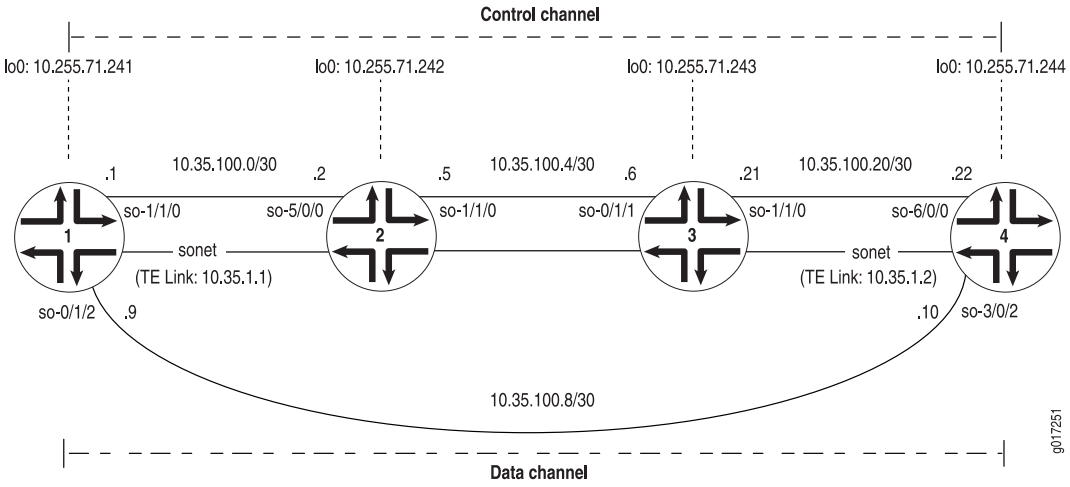


Figure 4 on page 31 shows a series of four routers that are used to establish an LMP control channel and traffic engineering link between Router R1 and Router R4. The control channel originates on the so-1/1/0 interface of Router 1 and ends at the so-6/0/0 interface of Router 4. The traffic engineering link data channel is a point-to-point link from the so-0/1/2 interface of Router R1 to the so-3/0/2 interface of Router R4.

On Router 1, configure IS-IS, MPLS, and RSVP to support the LMP control channel. For the control channel, use the so-1/1/0 interface as the origin and specify 10.35.100.22 (the so-6/0/0 interface on Router 4) as the remote address. For the traffic engineering link, use the so-0/1/2 interface as the origin, include a local address and remote address pair of your choice, and configure the remote IDs to match the local IDs generated by the remote peer.

Configure a bidirectional GMPLS LSP to reach Router 4. Use the loopback address of 10.255.71.244 as the destination for the LSP, disable CSPF, and configure a strict path to the remote address of the traffic engineering link.

Router 1

```
[edit]
interfaces {
   so-0/1/2 {
      description “Data channel to Router 4”;
      unit 0 {
         family inet {
            address 10.35.100.9/30;
         }
         family iso;
         family mpls;
      }
   }
   so-1/1/0 {
      description “Control channel to Router 4”;
```

```
        unit 0 {
            family inet {
                address 10.35.100.1/30;
            }
            family iso;
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.71.241/32;
            }
            family iso {
                address 47.0005.0000.0000.0000.0000.0000.0102.5507.1241.00;
            }
        }
    }
}
routing-options {
    router-id 10.255.71.241;
}
protocols {
    rsvp {
        interface all;
    }
    mpls {
        label-switched-path gmpls-router1-router4 {
            to 10.255.71.244;
            lsp-attributes {
                switching-type fiber;
            }
            no-cspf;
            primary path1;
        }
        path path1 {
            10.35.1.2 strict;
        }
        interface so-1/1/0.0;
    }
    isis {
        interface so-1/1/0.0;
        interface fxp0.0 {
            disable;
        }
        interface lo0.0 {
            passive;
        }
    }
    link-management {
        te-link sonet {
            local-address 10.35.1.1;
            remote-address 10.35.1.2;
            remote-id 8070;
            interface so-0/1/2 {
                remote-id 21303;
```

```
        }
      }
      peer router4 {
        address 10.255.71.244;
        lmp-control-channel so-1/1/0.0 {
          remote-address 10.35.100.22;
        }
        te-link sonet;
      }
      traceoptions {
        file lmp.logs size 5m files 10 world-readable;
        flag hello-packets;
        flag packets;
        flag state;
      }
    }
  }
```

On Router 2, configure IS-IS, MPLS, and RSVP to provide backbone connectivity for GMPLS and LMP between Routers 1 and 3.

Router 2

```
[edit]
interfaces {
  so-1/1/0 {
    description “Connection to Router 3”;
    unit 0 {
      family inet {
        address 10.35.100.5/30;
      }
      family iso;
      family mpls;
    }
  }
  so-5/0/0 {
    description “Connection to Router 1”;
    unit 0 {
      family inet {
        address 10.35.100.2/30;
      }
      family iso;
      family mpls;
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.255.71.242/32;
      }
      family iso {
        address 47.0005.0000.0000.0000.0000.0000.0102.5507.1242.00;
      }
    }
  }
}
routing-options {
```

```
    router-id 10.255.71.242;
}
protocols {
    rsvp {
        interface all;
    }
    mpls {
        interface all;
    }
    isis {
        interface so-1/1/0.0;
        interface so-5/0/0.0;
        interface fxp0.0 {
            disable;
        }
        interface lo0.0 {
            passive;
        }
    }
    link-management {
        traceoptions {
            file lmp.logs size 5m files 10 world-readable;
            flag hello-packets;
            flag packets;
            flag state;
        }
    }
}
```

The configuration of Router 3 is very similar to the configuration on Router 2. Configure IS-IS, MPLS, and RSVP to provide backbone connectivity for GMPLS and LMP between Routers 2 and 4.

Router 3

```
[edit]
interfaces {
    so-0/1/1 {
        description “Connection to Router 2”;
        unit 0 {
            family inet {
                address 10.35.100.6/30;
            }
            family iso;
            family mpls;
        }
    }
    so-1/1/0 {
        description “Connection to Router 4”;
        unit 0 {
            family inet {
                address 10.35.100.21/30;
            }
            family iso;
            family mpls;
        }
    }
```

```
    lo0 {
        unit 0 {
            family inet {
                address 10.255.71.243/32;
            }
            family iso {
                address 47.0005.0000.0000.0000.0000.0000.0102.5507.1243.00;
            }
        }
    }
}
routing-options {
    router-id 10.255.71.243;
}
protocols {
    rsvp {
        interface all;
    }
    mpls {
        interface all;
    }
    isis {
        interface so-0/1/1.0;
        interface so-1/1/0.0;
        interface fxp0.0 {
            disable;
        }
        interface lo0.0 {
            passive;
        }
    }
    link-management {
        traceoptions {
            file lmp.logs size 5m files 10 world-readable;
            flag hello-packets;
            flag packets;
            flag state;
        }
    }
}
```

On Router 4, complete the example by configuring IS-IS, MPLS, and RSVP to support the LMP control channel. For the control channel, use the **so-6/0/0** interface as the origin and specify **10.35.100.1** (the **so-1/1/0** interface on Router 1) as the remote address. For the traffic engineering link, use the **so-3/0/2** interface as the origin, swap the local address and remote address pair configured on Router 1 and add them here, and configure the remote IDs to match the local IDs generated by the remote peer.

Because GMPLS LSPs are bidirectional by default, you do not need to configure a return path to Router 1.

Router 4

```
[edit]
interfaces {
    so-3/0/2 {
```

```
        description “Data channel to Router 1”;
        unit 0 {
            family inet {
                address 10.35.100.10/30;
            }
            family iso;
            family mpls;
        }
    }
    so-6/0/0 {
        description “Control channel to Router 1”;
        unit 0 {
            family inet {
                address 10.35.100.22/30;
            }
            family iso;
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.71.244/32;
            }
            family iso {
                address 47.0005.0000.0000.0000.0000.0000.0102.5507.1244.00;
            }
        }
    }
}
routing-options {
    router-id 10.255.71.244;
}
protocols {
    rsvp {
        interface all;
    }
    mpls {
        interface all;
    }
    isis {
        interface so-6/0/0.0;
        interface fxp0.0 {
            disable;
        }
        interface lo0.0 {
            passive;
        }
    }
    link-management {
        te-link sonet {
            local-address 10.35.1.2;
            remote-address 10.35.1.1;
            remote-id 8070;
            interface so-3/0/2 {
                remote-id 22279;
```

```
        }
      }
      peer router1 {
        address 10.255.71.241;
        lmp-control-channel so-6/0/0.0 {
          remote-address 10.35.100.1;
        }
        te-link sonet;
      }
      traceoptions {
        file lmp.logs size 5m files 10 world-readable;
        flag hello-packets;
        flag packets;
        flag state;
      }
    }
  }
```

Verifying Your Work

To verify proper operation of LMP control channels, use the following commands:

- show link-management peer
- show link-management statistics
- show link-management te-link

The following sections show the output of these commands used with the configuration example:

- Router 1 Status on page 37
- Router 4 Status on page 38

Router 1 Status

On Router 1, issue the show link-management commands to verify that the control channel, traffic engineering links, and LMP negotiations are working as expected. The show link-management peer command indicates peer addresses, names, and identifiers, as well as control channel identifiers. The show link-management statistics command shows the number of LMP hellos and messages that have been exchanged. The show link-management te-link command displays names and identifiers configured for the traffic engineering link.

```
user@router1> show link-management peer
Peer name: router4, System identifier: 37495
 State: Up, Control address: 10.255.70.103
   CC local ID CC remote ID State       TxSeqNum  RcvSeqNum Flags
         22515        38882 Up              1692       1691
  TE links:
   sonet

user@router1> show link-management statistics
Statistics for peer router4
```

```
    Received packets
        ConfigAck: 1
        Hello: 748
        LinkSummary: 13
        LinkSummaryAck: 1
    Small packets: 0
    Wrong protocol version: 0
    Messages for unknown peer: 0
    Messages for bad state: 0
    Stale acknowledgements: 0
    Stale negative acknowledgements: 0
    Sent packets
        Config: 24
        Hello: 748
        LinkSummary: 13
        LinkSummaryAck: 13
    Retransmitted packets
        Config: 18
        LinkSummary: 9
    Dropped packets
        Config: 5
        LinkSummary: 3

user@router1> show link-management te-link
 TE link name: sonet, State: Up
  Local identifier: 8070, Remote identifier: 8070, Local address: 10.35.1.1,
  Remote address: 10.35.1.2, Encoding: SDH/SONET, Switching: PSC-1,
  Minimum bandwidth: 622.08Mbps, Maximum bandwidth: 622.08Mbps,
  Total bandwidth: 622.08Mbps, Available bandwidth: 622.08Mbps
    Name        State Local ID  Remote ID      Bandwidth Used  LSP-name
    so-0/1/2    Up       22279      21303     622.08Mbps   No
```

Router 4 Status

On Router 4, issue the show link-management commands to verify that the control channel, traffic engineering links, and LMP negotiations are being reciprocated by Router 1.

```
user@router4> show link-management peer
Peer name: router1, System identifier: 56483
 State: Up, Control address: 10.255.71.242
   CC local ID CC remote ID State       TxSeqNum  RcvSeqNum Flags
         38882        22515 Up              1451       1450
  TE links:
   sonet

user@router4> show link-management statistics
Statistics for peer router1
    Received packets
        Config: 1
        Hello: 255
        LinkSummary: 1
        LinkSummaryAck: 1
    Small packets: 0
    Wrong protocol version: 0
    Messages for unknown peer: 0
    Messages for bad state: 0
    Stale acknowledgements: 0
    Stale negative acknowledgements: 0
```

```
    Sent packets
        Config: 31
        ConfigAck: 1
        Hello: 255
        LinkSummary: 13
        LinkSummaryAck: 1
    Retransmitted packets
        Config: 23
        LinkSummary: 9
    Dropped packets
        Config: 7
        LinkSummary: 3

user@router4> show link-management te-link
TE link name: sonet, State: Up
  Local identifier: 8070, Remote identifier: 8070, Local address: 10.35.1.2,
  Remote address: 10.35.1.1, Encoding: SDH/SONET, Switching: PSC-1,
  Minimum bandwidth: 622.08Mbps, Maximum bandwidth: 622.08Mbps,
  Total bandwidth: 622.08Mbps, Available bandwidth: 622.08Mbps
    Name        State Local ID  Remote ID      Bandwidth Used  LSP-name
    so-3/0/2    Up       21303      22279     622.08Mbps   No
```

For More Information

For additional information about implementing GMPLS, see the following:

- *JUNOS MPLS Applications Configuration Guide*
- RFC 2205, *Resource ReSerVation Protocol (RSVP)*
- RFC 3471, *Generalized Multi-Protocol Label Switching (GMPLS)—Signaling Functional Description*
- RFC 3473, *Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions*
- RFC 4204, *Link Management Protocol (LMP)* (The JUNOS Software supports sections 3 and 4)
- 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, *GMPLS Extensions for SONET and SDH Control* (expires August 2003)
- Internet draft draft-ietf-ccamp-ospf-gmpls-extensions-12.txt, *OSPF Extensions in Support of Generalized MPLS* (expires April 2004)
- Internet draft draft-ietf-mpls-bundle-04.txt, *Link Bundling in MPLS Traffic Engineering* (expires January 2003)
- Internet draft draft-ietf-mpls-lsp-hierarchy-08.txt, *LSP Hierarchy with Generalized MPLS TE* (expires March 2003)
- Internet draft draft-ietf-ccamp-gmpls-routing-09.txt, *Routing Extensions in Support of Generalized MPLS* (expires April 2004)

Revision History

5 October 2009—10.0R1 Release. Roy Spencer.

13 June 2009—9.6R1 Release. Roy Spencer.

13 April 2009—9.5R1 Release. Roy Spencer.

15 January 2009—9.4R1 Release. Roy Spencer.

10 October 2008—9.3R1 Release. Roy Spencer.

10 July 2008—9.2R1 Release. Roy Spencer.

10 April 2008—9.1R1 Release. Roy Spencer.

1 February 2008—9.0R1 Release. Fawn Damitio.

27 March 2007—8.3R1 Release. Fawn Damitio.

12 January 2007—Added support for nonpacket LSP path establishment through JUNOS Software-based routers. 8.2R1 Release. Fawn Damitio.

15 September 2006—Added support for LMP control channels, 8.1R1 Release. Richard Hendricks.

29 June 2006—8.0R1 Release. Richard Hendricks.

27 March 2006—7.6R1 Release. Richard Hendricks.

9 January 2006—7.5R1 Release. Richard Hendricks.

14 September 2005—7.4R1 Release. Richard Hendricks.

13 June 2005—7.3R1 Release. Richard Hendricks.

5 April 2005—Added the steps necessary to configure GMPLS with the peer model and the overlay model, 7.2R1 Release. Richard Hendricks.

2 February 2005—7.1R1 Release. Richard Hendricks.

6 October 2004—Added support for graceful teardown of GMPLS LSPs and graceful restart of GMPLS neighbors, 7.0R1 Release. Richard Hendricks.

6 July 2004—6.4R1 Release. Richard Hendricks.

5 April 2004—6.3R1 Release. Richard Hendricks.

22 December 2003—6.2R1 Release. Richard Hendricks.

22 September 2003—6.1R1 Release. Richard Hendricks.

30 June 2003—6.0R1 Release. Richard Hendricks.

2 April 2003—5.7R1 Release. Richard Hendricks.

27 December 2002—Added OSPF and CSPF updates, 5.6R1 Release. Richard Hendricks.

30 September 2002—Added one note for the 5.5R1 Release. Richard Hendricks.

19 July 2002—5.4R1 Release. Richard Hendricks.

28 June 2002—Revised commands and statements. Richard Hendricks.

6 May 2002—Initial document written. Richard Hendricks.

Chapter 2

Connecting IPv6 Islands with IPv4 MPLS

This chapter covers the following topics:

- Connecting IPv6 Islands with IPv4 MPLS Overview on page 43
- Connecting IPv6 Islands with IPv4 MPLS System Requirements on page 45
- Connecting IPv6 Islands with IPv4 MPLS Terms and Acronyms on page 46
- Configuring an IPv4 MPLS Tunnel to Carry IPv6 Traffic on page 46
- Configuring IPv6 on the Customer and Core-Facing Interfaces on page 46
- Configuring MPLS and RSVP from PE Router to PE Router to Create a Tunnel on page 47
- Enabling IPv6 Tunneling in MPLS on page 47
- Configuring Multiprotocol BGP to Carry IPv6 Traffic on page 47
- Example: Connecting IPv6 Islands over an MPLS Tunnel Configuration on page 48
- For More Information on page 58
- Revision History on page 58

Connecting IPv6 Islands with IPv4 MPLS Overview

Many service providers are looking for ways to provide new revenue-generating services to their customers. One such service is Internet Protocol version 6 (IPv6). Some enterprise customers are beginning to experiment with this new version of IP, but are reluctant to deploy it broadly. Interconnecting multiple sites that use IPv6 can be challenging. Also, most service providers would prefer to carry this traffic without making major modifications to their core network.

A technique available in JUNOS Release 5.4 allows you to connect IPv6 sites over an IPv4 Multiprotocol Label Switching (MPLS) enabled backbone. Juniper Networks supports the Multiprotocol Border Gateway Protocol (MP-BGP) over IPv4 approach detailed in the Internet Engineering Task Force (IETF) Internet draft draft-ooms-v6ops-bgp-tunnel-06.txt, *Connecting IPv6 Islands over IPv4 MPLS using IPv6 Provider Edge Routers (6PE)* (expires July 2006). With this technique, IPv6 islands are connected to each other across an IPv4 backbone enabled with MPLS label stacking while MP-BGP is used to announce the IPv6 routes across these MPLS tunnels. This feature can be implemented with label-switched paths (LSPs) using the Label Distribution Protocol (LDP) or Resource Reservation Protocol (RSVP).

IPv6 packets are carried over an IPv4 MPLS tunnel. To enable this service, you need to deploy provider edge (PE) routers that can run IPv4, MPLS, and BGP toward the core and IPv6 toward the edge. Since only the PE routers need to run a dual stack of IPv4 and IPv6, the other provider (P) core routers do not need to be upgraded. As a result, this MPLS tunneling technique allows for interoperability with routers from other vendors.

Because of this flexible method of implementation, it is now more attractive for providers to carry IPv6 traffic over their existing core networks and for customers to roll out IPv6 to more sites.

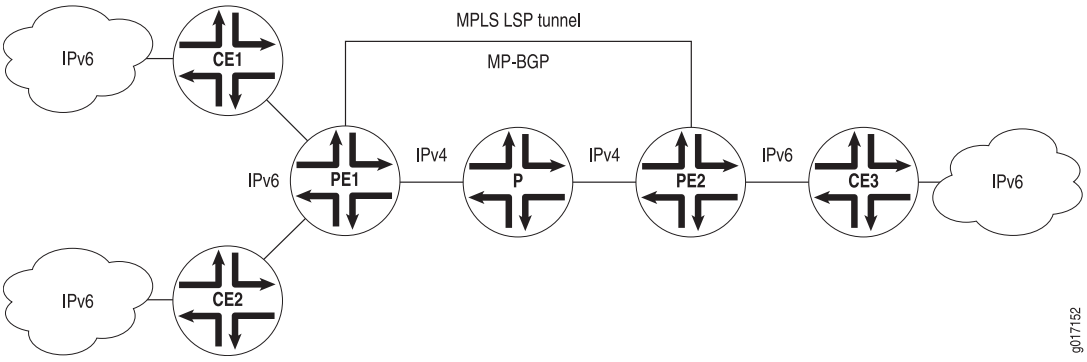
In Figure 5 on page 44, PE1 and PE2 are dual-stack Border Gateway Protocol (DS-BGP) routers. They implement IPv4 and IPv6 stacks simultaneously. The IPv6 clouds are separate islands that are connected to PE routers through a customer edge (CE) router.

This example shows how to enable IPv6 connectivity between the various IPv6 islands, not how to create an IPv6 VPN service. One of the IPv6 islands can be the global IPv6 Internet.

The connection between the CE and PE routers can use any network layer protocol that carries IPv6 traffic. The provider router can exchange information with the customer routers using IPv6-enabled routing protocols, such as RIPng or MP-BGP, or static routes. The PE routers use IPv6 on the CE-facing interfaces, but use IPv4, BGP, and MPLS to connect to the core.

You must configure appropriate export policies on the PE router to share route information between IBGP and EBGP, and between BGP and other protocols.

Figure 5: Connecting IPv6 Islands over MPLS



Because MP-BGP requires that a BGP next hop use the same address family as the Network Layer Reachability Information (NLRI), the IPv4 address needs to be embedded in an IPv6 format. Such IPv4-mapped IPv6 addresses are defined in RFC 3513, *IP Version 6 Addressing Architecture*. After the PE routers learn the IPv6 routes from their directly attached CE neighbors, each PE router uses its own IPv4 address as the next hop for the IPv6 routes that are advertised in the BGP session.

The two PE routers establish an MP-BGP session with each other using IPv4 addresses. In the session, the routers exchange IPv6 routes with an IPv6 address family identifier (AFI) value of 2 and a subsequent AFI (SAFI) label with a value of 4. Labels with a

value of 2 are explicit null labels for IPv6, as defined in RFC 3032. Before sending IPv6 traffic across the IPv4 MPLS tunnel, the PE attaches the two labels. The inner label is 2 (another value if the advertising PE router is not a Juniper Networks router) and the outer label is the LSP label.

A PE router must have MPLS LSPs pointing to the other peer PE router's IPv4 address. The LSPs are signaled across the IPv4 control plane using either LDP or RSVP. These LSPs resolve the next-hop addresses of the IPv6 routes learned through MP-BGP. The next hops are actually IPv4-mapped IPv6 addresses, whereas the LSPs are associated with IPv4 addresses. Because of this mapping technique, the IPv6 traffic can travel over the IPv4 LSP transparently.

In Figure 5 on page 44, PE1 receives an IPv6 packet from CE1 and performs a lookup in the IPv6 forwarding table. If the destination matches a prefix that was learned from CE2, no labels are necessary and the IPv6 packet is sent to CE2. If the destination matches a prefix that was learned from PE2, then PE1 places two labels on the packet and sends it to P. The inner label is 2 and the outer label is the LSP label needed to reach PE2. Since P is the penultimate-hop router for the LSP to PE2 and the received packet has more than one label, Router P pops the outer label and sends the packet to PE2. When PE2 receives the packet, it has a single label with a value of 2. PE2 strips off the label and treats the remaining packet as an IPv6 packet (since 2 is the IPv6 explicit null label) and performs a lookup in the IPv6 forwarding table.

Although the MP-BGP over IPv4 approach can operate using a single level of labels, there is an advantage in using two labels. The penultimate-hop router for the MPLS LSP (P in this case) can pop the outer label and send the packet with the inner label as an MPLS packet. When the packet arrives at egress Router PE2, the second label using the explicit null value is popped and the remaining IPv6 packet is sent to the directly connected IPv6 network. Thus, the benefit of using two labels is that penultimate hop-popping (PHP) routers do not require IPv6 capabilities or the need for an upgrade.

Interconnecting IPv6 islands over an IPv4 MPLS tunnel requires:

- An exchange of IPv6 reachability information between DS-BGP routers. Using MP-BGP, the DS-BGP (PE) routers exchange IPv6 reachability information over the IPv4 core network with other similarly enabled DS-BGP PE peers. As a result, the egress DS-BGP (PE) router announces itself as the BGP next hop.
- IPv6 packets are tunneled from the ingress DS-BGP router to the egress DS-BGP router by means of MPLS. The ingress DS-BGP router tunnels an IPv6 packet over the IPv4 network toward the egress DS-BGP router identified as the BGP next hop for the packet's destination IPv6 address.

Connecting IPv6 Islands with IPv4 MPLS System Requirements

To carry IPv6 traffic over IPv4 MPLS tunnels, your system must meet these minimum requirements:

- JUNOS Release 8.2 or later for MX Series routers
- JUNOS Release 7.2 or later for J Series Services Routers

- JUNOS Release 5.4 or later for M Series and T Series routers
- Two Juniper Networks J Series, M Series, MX Series, or T Series routers to act as the DS-BGP ingress and egress devices

Connecting IPv6 Islands with IPv4 MPLS Terms and Acronyms

D

dual-stack BGP (DS-BGP) A router that processes IPv4 and IPv6 packets in a BGP-connected network.

M

Multiprotocol BGP (MP-BGP) A router enabled for MP-BGP processes packets from a variety of protocols in a BGP-connected network.

S

Subsequent Address Family Identifier (SAFI) A field in Multiprotocol BGP messages that identifies MPLS network layer reachability information (NLRI). Common values include 1 (unicast), 2 (multicast), and 4 (MPLS label).

Configuring an IPv4 MPLS Tunnel to Carry IPv6 Traffic

To enable IPv6 to be carried over an IPv4 MPLS tunnel, perform the following tasks:

- Configuring IPv6 on the Customer and Core-Facing Interfaces on page 46
- Configuring MPLS and RSVP from PE Router to PE Router to Create a Tunnel on page 47
- Enabling IPv6 Tunneling in MPLS on page 47
- Configuring Multiprotocol BGP to Carry IPv6 Traffic on page 47

Configuring IPv6 on the Customer and Core-Facing Interfaces

Configure family inet6 on all the CE-facing interfaces and on all the core-facing interfaces running MPLS. This enables the router 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.

```
[edit]
interfaces {
    interface-name {
        unit unit-number {
            family inet6 {
                address inet6-address;
```

```
        }
      }
    }
}
```

Configuring MPLS and RSVP from PE Router to PE Router to Create a Tunnel

This guide assumes you already have experience configuring MPLS and RSVP. For more information about these topics, see the *JUNOS MPLS Applications Configuration Guide*.

Enabling IPv6 Tunneling in MPLS

Enter the ipv6-tunneling option on your PE routers at the [edit protocols mpls] hierarchy level:

```
[edit]
protocols {
    mpls {
        ipv6-tunneling;
    }
}
```

Configuring Multiprotocol BGP to Carry IPv6 Traffic

You can specify the family inet6 statement on a per-neighbor, per-group, or global basis. The statement allows BGP to carry IPv6 traffic.

At the appropriate global, group, or neighbor hierarchy level in BGP (shown below), configure the family inet6 statement with the labeled-unicast parameter and the explicit-null option. These additional parameters enable the IPv4 MPLS label to be removed at the destination PE router. The remaining IPv6 packet without a label can then be forwarded to the connected IPv6 network.

```
[edit protocols bgp] OR
[edit protocols bgp group group-name] OR
[edit protocols bgp group group-name neighbor neighbor-name]
family inet6 {
    labeled-unicast {
        explicit-null;
    }
}
```

Example: Connecting IPv6 Islands over an MPLS Tunnel Configuration

Figure 6: IPv6 over an MPLS Tunnel

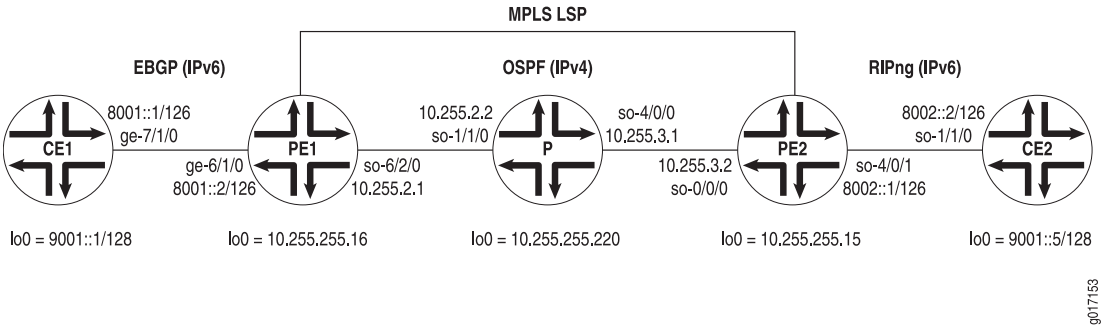


Figure 6 on page 48 shows a standard CE-PE-P-PE-CE MPLS-style network. CE1 and CE2 are the end customer CE routers using IPv6; PE1 and PE2 are the provider edge routers; and P is a provider core router. The IPv4 MPLS tunnel travels between PE1 and PE2, connecting IPv6 sites CE1 and CE2.

Since the CE-to-PE configuration can use a variety of routing protocols, this example requires that you use EBGP between CE1 and PE1 and RIPng between PE2 and CE2. You must establish policies on PE2 to import and export routes between BGP and RIPng.

To start the configuration, set up the IPv6 connection between CE1 and PE1. In your BGP routing policy, you must advertise the IPv6 loopback address of the CE1 router address to the PE1 router.

Router CE1

```
[edit]
interfaces {
    ge-7/1/0 {
        unit 0 {
            family inet6 {
                address 8001::1/126;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet6 {
                address 9001::1/128;
            }
        }
    }
}
routing-options {
    autonomous-system 200;
}
protocols {
    bgp {
        group to_PE1 {
```

```
        type external;
        local-address 8001::1;
        family inet6 {
          unicast;
        }
        export policy1;
        peer-as 100;
        neighbor 8001::2;
      }
    }
    policy-options {
      policy-statement policy1 {
        term 1 {
          from {
            family inet6;
            route-filter 9001::1/128 exact;
          }
          then accept;
        }
        term 2 {
          then reject;
        }
      }
    }
}
```

Once you move to PE1, your tasks become more complex. You must complete the IPv6 EBGP connection to CE1 and build the first part of the MPLS tunnel. You must set the inet, inet6, and mpls families on the core-facing interface, configure an inet6 address for the CE-facing interface attached to CE1, and ensure the IPv4 loopback address is advertised in OSPF, since this is the MPLS LSP target for PE2. You must also add the ipv6-tunneling parameter in MPLS, include the labeled-unicast and explicit-null options at the [edit protocols bgp family inet6] hierarchy level, and create an external BGP group pointing to CE1 and an internal group pointing to PE2.

Router PE1

```
[edit]
interfaces {
  ge-6/1/0 {
    unit 0 {
      family inet6 {
        address 8001::2/126;
      }
    }
  }
  so-6/2/0 {
    unit 0 {
      family inet {
        address 10.255.2.1/24;
      }
      family inet6;
      family mpls;
    }
  }
  lo0 {
    unit 0 {
```

```
            family inet {
               address 10.255.255.16/32;
            }
         }
      }
   }
   routing-options {
      autonomous-system 100;
   }
   protocols {
      rsvp {
         interface so-6/2/0.0;
      }
      mpls {
         ipv6-tunneling;
         label-switched-path to_PE2 {
            to 10.255.255.15;
         }
         interface so-6/2/0.0;
      }
      bgp {
         group to_PE2 {
            type internal;
            local-address 10.255.255.16;
            family inet6 {
               labeled-unicast {
                  explicit-null;
               }
            }
            neighbor 10.255.255.15;
         }
         group to_CE1 {
            local-address 8001::2;
            family inet6 {
               unicast;
            }
            peer-as 200;
            neighbor 8001::1;
         }
      }
      ospf {
         traffic-engineering;
         area 0.0.0.0 {
            interface so-6/2/0.0;
            interface lo0.0 {
               passive;
            }
         }
      }
   }
```

On Router P, connect the MPLS tunnel between PE1 and PE2. Enable RSVP, MPLS, and IPv4 connectivity on the interfaces and ensure that IP connectivity is available through the routing protocol (in this case, OSPF).

Router P

```
[edit]
interfaces {
    so-1/1/0 {
        unit 0 {
            family inet {
                address 10.255.2.2/24;
            }
            family mpls;
        }
    }
    so-4/0/0 {
        unit 0 {
            family inet {
                address 10.255.3.1/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.255.220/32;
            }
        }
    }
}
routing-options {
    autonomous-system 100;
}
protocols {
    rsvp {
        interface so-1/1/0.0;
        interface so-4/0/0.0;
    }
    mpls {
        interface so-1/1/0.0;
        interface so-4/0/0.0;
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface so-1/1/0.0;
            interface so-4/0/0.0;
            interface lo0.0 {
                passive;
            }
        }
    }
}
```

At PE2, you must complete a mirror image of the MPLS tunnel configuration started at PE1 and configure a RIPng connection to CE2. Set the inet, inet6, and mpls families on the core-facing interface, configure an inet6 address for the CE facing interface attached to CE2, and ensure the IPv4 loopback address is advertised in OSPF, since this is the MPLS LSP target for PE1. You must also add the ipv6-tunneling parameter

in MPLS and include the labeled-unicast and explicit-null options at the [edit protocols bgp family inet6] hierarchy level. Finally, create and apply policies that export BGP routes into RIPng and import RIPng routes to BGP.

Router PE2

```
[edit]
interfaces {
   so-0/0/0 {
      unit 0 {
         family inet {
            address 10.255.3.2/24;
         }
         family inet6;
         family mpls;
      }
   }
   so-4/0/1 {
      unit 0 {
         family inet6 {
            address 8002::1/126;
         }
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.255.255.15/32;
         }
      }
   }
}
routing-options {
   autonomous-system 100;
}
protocols {
   rsvp {
      interface so-0/0/0.0;
   }
   mpls {
      ipv6-tunneling;
      label-switched-path to_PE1 {
         to 10.255.255.16;
      }
      interface so-0/0/0.0;
   }
   bgp {
      group to_PE1 {
         type internal;
         local-address 10.255.255.15;
         family inet6 {
            labeled-unicast {
               explicit-null;
            }
         }
         export red-export;
         neighbor 10.255.255.16;
      }
```

```
    }
    ospf {
      traffic-engineering;
      area 0.0.0.0 {
        interface so-0/0/0.0;
        interface lo0.0 {
          passive;
        }
      }
    }
    ripng {
      group to_CE2 {
        export red-import;
        neighbor so-4/0/1.0;
      }
    }
  }
  policy-options {
    policy-statement red-export {
      term 1 {
        from protocol ripng;
        then accept;
      }
      term 2 {
        then reject;
      }
    }
    policy-statement red-import {
      from protocol bgp;
      then accept;
    }
  }
```

Finally, on Router CE2, configure IPv6 addresses on the SONET/SDH and loopback interfaces, enable RIPng, and create and apply a policy for RIPng that permits the IPv6 loopback address to be exported to Router PE2. Once these tasks are accomplished, your IPv6 connection to Router CE1 should be ready for use.

Router CE2

```
[edit]
interfaces {
  so-1/1/0 {
    unit 0 {
      family inet6 {
        address 8002::2/126;
      }
    }
  }
  lo0 {
    unit 0 {
      family inet6 {
        address 9001::5/128;
      }
    }
  }
}
```

```
routing-options {
   autonomous-system 300;
}
protocols {
   ripng {
      group to_PE2 {
         export policy1;
         neighbor so-1/1/0.0;
      }
   }
}
policy-options {
   policy-statement policy1 {
      term 1 {
         from {
            family inet6;
            route-filter 9001::5/128 exact;
         }
         then accept;
      }
      term 2 {
         then reject;
      }
   }
}
```

Verifying Your Work

To verify that IPv6 traffic is being transported over the IPv4 MPLS tunnel, use the following commands:

- ping
- show bgp summary
- show route protocol
- show route advertising-protocol
- show route receive-protocol
- show route table
- show route table (inet6.0 | inet6.3)
- show interfaces terse

The following sections show the output of these commands used with the configuration example:

- Router CE1 Status on page 55
- Router PE1 Status on page 55
- Router PE2 Status on page 56
- Router CE2 Status on page 57

Router CE1 Status

```
user@CE1> show bgp summary
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet6.0                1          1          0          0          0          0
Peer              AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Damped...
8001::2           100         58         56       0       0       26:25 Establ
  inet6.0: 1/1/0

user@CE1>  show route protocol bgp
inet.0: 13 destinations, 13 routes (12 active, 0 holddown, 1 hidden)
iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
inet6.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
9001::5/128         *[BGP/170] 00:04:18, localpref 100
                      AS path: 100 I
                    >  to 8001::2 via ge-7/1/0.0

user@CE1>  ping 9001::5 source 9001::1
PING6(56=40+8+8 bytes) 9001::1 --> 9001::5
16 bytes from 9001::5, icmp_seq=0 hlim=62 time=0.945 ms
16 bytes from 9001::5, icmp_seq=1 hlim=62 time=0.831 ms
^C
--- 9001::5 ping6 statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.831/0.887/0.945 ms
```

Router PE1 Status

```
user@PE1> show bgp summary
Groups: 2 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet6.0                2          2          0          0          0          0
Peer              AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Damped...
8001::1           200         56         61       0       0       27:18 Establ
  inet6.0: 1/1/0
10.255.255.15      100         13         14       0       1        5:28 Establ

  inet6.0: 1/1/0

user@PE1>  show route advertising-protocol bgp 10.255.255.15 detail
inet6.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
9001::1/128 (1 entry, 1 announced)
 BGP group to_PE2 type Internal
     Route Label: 2
     Nexthop: Self
     Localpref: 100
     AS path: 200 I
     Communities:

user@PE1>  show route 9001::5
inet6.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
9001::5/128        *[BGP/170] 00:05:48, MED 2, localpref 100, from 10.255.255.15
                      AS path: I
                    > via so-6/2/0.0, label-switched-path to_PE2

user@PE1> show route table inet6.0
inet6.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
8001::/126         *[Direct/0] 00:29:01
                    > via ge-6/1/0.0
8001::2/128        *[Local/0] 00:29:01
                      Local via ge-6/1/0.0
9001::1/128        *[BGP/170] 00:28:46, localpref 100
                      AS path: 200 I
                    > to 8001::1 via ge-6/1/0.0
9001::2/128        *[Direct/0] 00:29:01
                    > via lo0.0
9001::5/128        *[BGP/170] 00:06:56, MED 2, localpref 100, from 10.255.255.15
                      AS path: I
                    > via so-6/2/0.0, label-switched-path to_PE2
fe80::/64          *[Direct/0] 00:29:01
                    > via ge-6/1/0.0
fe80::280:42ff:fe10:d30c/128
                   *[Direct/0] 00:29:01
                    > via lo0.0
fe80::290:69ff:fe0f:1633/128
                   *[Local/0] 00:29:01
                      Local via ge-6/1/0.0

user@PE1>  show route table inet6.3
inet6.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
::ffff:10.255.255.15/128 *[RSVP/7] 00:06:37, metric 2, metric2 0
                    > via so-6/2/0.0, label-switched-path to_PE2
```

Router PE2 Status

```
user@PE2> show bgp summary
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet6.0               1          1          0          0          0          0
Peer               AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Damped...
10.255.255.16      100         18         20       0       0        8:06 Establ
  inet6.0: 1/1/0

user@PE2> show interfaces terse so-4/0/1
Interface       Admin Link Proto Local                 Remote
so-4/0/1        up    up
so-4/0/1.0      up    up   inet  100.1.4.1/24
                           inet6 8002::1/126
                                 fe80::280:42ff:fe10:d312/64

user@PE2>  show route receive-protocol bgp 10.255.255.16 detail

inet.0: 18 destinations, 19 routes (17 active, 0 holddown, 1 hidden)

inet.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
```

```
iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

mpls.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)

inet6.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
9001::1/128 (1 entry, 1 announced)
     Route Label: 2
     Nexthop: ::ffff:10.255.255.16
     Localpref: 100
     AS path: 200 I

inet6.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
user@PE2> show route advertising-protocol ripng fe80::280:42ff:fe10:d312 detail
inet6.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
9001::1/128 (1 entry, 1 announced)
        *BGP    Preference: 170/-101
                Source: 10.255.255.16
                Next hop: via so-0/0/0.0, weight 1, selected
                Label-switched-path to_PE1
                Label operation: Push 2, Push 100015(top)
                Protocol next hop: ::ffff:10.255.255.16
                Push 2
                 Indirect next hop: 8451440 50
                State: <Active Int Ext>
                Local AS:   100 Peer AS:   100
                Age: 2:27       Metric2: 2
                Task: BGP_100.10.255.255.16+179
                Announcement bits (3): 0-KRT 1-RIPng 3-Resolve inet6.0
                AS path: 200 I
                Route Label: 2
```

Router CE2 Status

```
user@CE2> show ripng neighbor
                    Source                          Dest              In
Neighbor     State  Address                         Address  Send Recv Met
--------     -----  -------                         -------  ---- ---- ---
so-1/1/0.0      Up  fe80::2a0:a5ff:fe12:34d9        ff02::9  yes yes    1

user@CE2> show route protocol ripng

inet.0: 12 destinations, 12 routes (11 active, 0 holddown, 1 hidden)

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

inet6.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

9001::1/128        *[RIPng/100] 00:04:10, metric 2, tag 0
                    > to fe80::280:42ff:fe10:d312 via so-1/1/0.0
ff02::9/128        *[RIPng/100] 02:42:33, metric 1
                      MultiRecv
```

For More Information

For additional information about connecting IPv6 islands with IPv4 MPLS, see the following:

- *JUNOS MPLS Applications Configuration Guide*
- *JUNOS Routing Protocols Configuration Guide*
- RFC 3032, *MPLS Label Stack Encoding*
- RFC 3107, *Carrying Label Information in BGP-4*
- RFC 3513, *IP Version 6 Addressing Architecture*
- 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 (6PE)* (expires July 2006)

Revision History

5 October 2009—10.0R1 Release. Roy Spencer.

13 June 2009—9.6R1 Release. Roy Spencer.

13 April 2009—9.5R1 Release. Roy Spencer.

15 January 2009—9.4R1 Release. Roy Spencer.

10 October 2008—9.3R1 Release. Roy Spencer.

10 July 2008—9.2R1 Release. Roy Spencer.

10 April 2008—9.1R1 Release. Roy Spencer.

1 February 2008—9.0R1 Release. Fawn Damitio.

27 March 2007—8.3R1 Release. Fawn Damitio.

12 January 2007—Added support for MX960 Ethernet Services Routers. 8.2R1. Fawn Damitio.

15 September 2006—8.1R1 Release. Richard Hendricks.

29 June 2006—8.0R1 Release. Richard Hendricks.

27 March 2006—7.6R1 Release. Richard Hendricks.

9 January 2006—7.5R1 Release. Richard Hendricks.

14 September 2005—7.4R1 Release. Richard Hendricks.

13 June 2005—7.3R1 Release. Richard Hendricks.

5 April 2005—Added support for J Series Services Routers, 7.2R1 Release. Richard Hendricks.

2 February 2005—7.1R1 Release. Richard Hendricks.

6 October 2004—7.0R1 Release. Richard Hendricks.

6 July 2004—6.4R1 Release. Richard Hendricks.

5 April 2004—6.3R1 Release. Richard Hendricks.

22 December 2003—6.2R1 Release. Richard Hendricks.

22 September 2003—6.1R1 Release. Richard Hendricks.

30 June 2003—6.0R1 Release. Richard Hendricks.

2 April 2003—5.7R1 Release. Richard Hendricks.

27 December 2002—5.6R1 Release. Richard Hendricks.

30 September 2002—5.5R1 Release. Richard Hendricks.

19 July 2002—5.4R1 Release. Richard Hendricks.

6 May 2002—Initial document written. Richard Hendricks.

Chapter 3

Multiple Instances for Label Distribution Protocol

This chapter covers the following topics:

- Multiple Instances for Label Distribution Protocol Overview on page 61
- Multiple Instances for Label Distribution Protocol System Requirements on page 62
- Multiple Instances for Label Distribution Protocol Terms and Acronyms on page 63
- Example: Configuring Multiple-Instance LDP on page 63
- Revision History on page 98

Multiple Instances for Label Distribution Protocol Overview

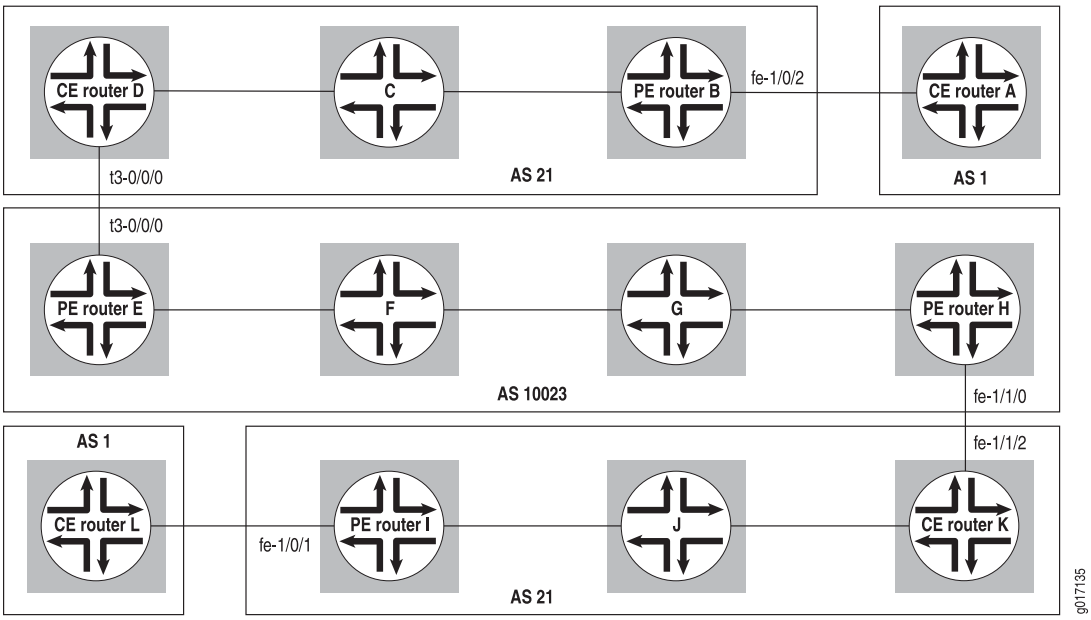
Previous versions of JUNOS Software support multiple VPN routing and forwarding (VRF) instances of Border Gateway Protocol (BGP), Intermediate System-to-Intermediate System (IS-IS), Open Shortest Path First (OSPF), Protocol Independent Multicast (PIM), and Routing Information Protocol (RIP). JUNOS Release 5.4 and later adds support for multiple instances of the Label Distribution Protocol (LDP).

This support allows LDP to be used to advertise labels in a carrier-of-carriers scenario from a core provider edge (PE) router to a customer carrier 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 at large. Multiple-instance LDP is also useful when a carrier customer wants to provide Layer 2 VPN or Layer 3 VPN services to its customers.

Using multiple-instance LDP lets you circumvent one of the requirements of RFC 3107: the need to run full-mesh internal BGP (IBGP) within the carrier customer's autonomous system (AS). When you use multiple-instance LDP, full-mesh IBGP is unnecessary.

In Figure 7 on page 62, the customer carrier in AS 21 can configure one instance of LDP for all routers in AS 21 instead of using full-mesh IBGP.

Figure 7: Carrier-of-Carriers Example



In general, if there are a limited number of customer carrier sites and few internal routes in the customer carrier AS, it is simpler and quicker to use LDP than to configure a full IBGP mesh.

An instance of LDP operates essentially in the same way as a master instance. Each instance of LDP must be enabled on all the desired interfaces and a separate set of LDP data structures are maintained for each instance. Instance information includes a set of LDP interfaces, neighbors, sessions, and databases.

For more information about carrier-of-carriers VPNs, see the *JUNOS VPNs Configuration Guide*.

For more information about LDP, see the *JUNOS MPLS Applications Configuration Guide*.

Multiple Instances for Label Distribution Protocol System Requirements

To implement the multiple-instance LDP feature, your system must meet these minimum requirements:

- JUNOS Release 8.2 for support on MX Series routers.
- JUNOS Release 5.4 or later for support on M Series and T Series routers.
- Two Juniper Networks M Series, MX Series, or T Series routers for basic multiple-instance LDP; and a minimum of four Juniper Networks routers to act as PE routers in a carrier-of-carriers network.

Multiple Instances for Label Distribution Protocol Terms and Acronyms

C

carrier-of-carriers VPN A VPN that transports data traffic between two or more telecommunications carrier sites across a core provider network. The core provider becomes a carrier for the customer carrier, which, in turn, provides Internet or VPN services to end customers. For more information about carrier-of-carriers VPNs, see the *JUNOS VPNs Configuration Guide*.

L

Label Distribution Protocol (LDP) A protocol used to distribute labels in an MPLS-enabled network. For more information about LDP, see the *JUNOS MPLS Applications Configuration Guide*.

V

VPN routing and forwarding (VRF) instance A unique routing table created to maintain VPN routing and forwarding information. One routing table is created per instance, which keeps prefix information and data private from other instances. For more information about VRF instances, see the *JUNOS VPNs Configuration Guide*.

Example: Configuring Multiple-Instance LDP

The master 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 master 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 8: Multiple-Instance LDP Topology Diagram

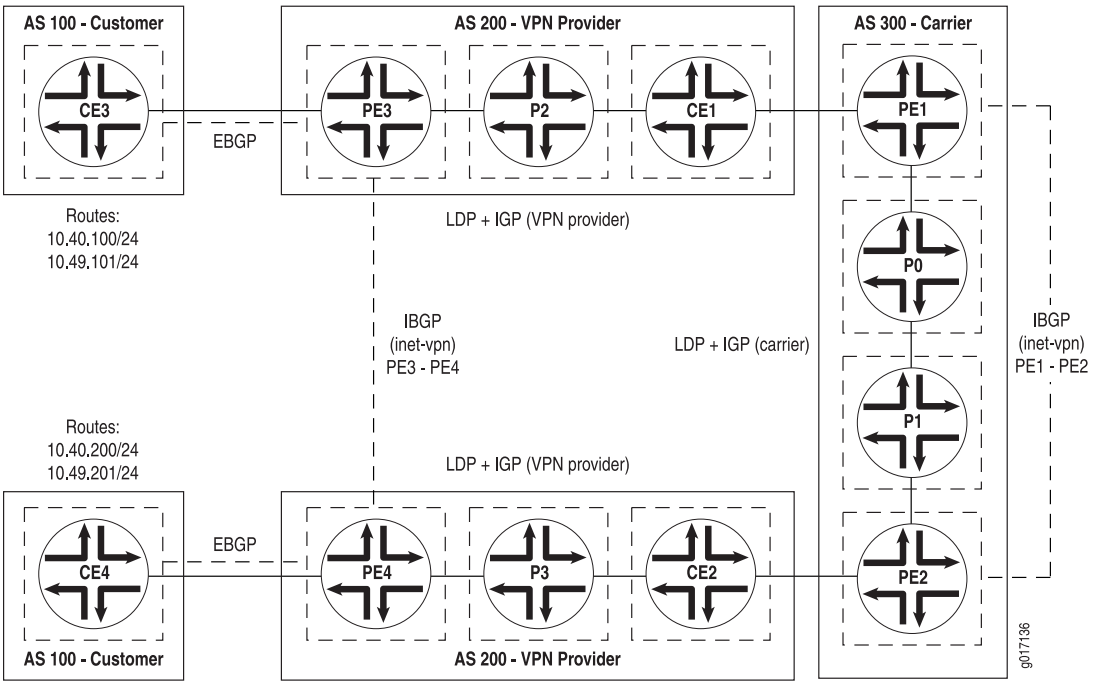


Figure 8 on page 64 shows an example of a carrier-of-carriers network. CE3 and CE4 are end customer CE routers residing in AS 100. The VPN provider in AS 200 has three types of routers: PE3 and PE4 are PE routers that connect to the end customer, CE1 and CE2 act as the intermediate carrier CE routers, and P2 and P3 are internal transit routers. PE1 and PE2 in AS 300 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 PE3 and PE 4).
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 4 on page 64 and Table 5 on page 65.

Table 4: Multiple-Instance LDP Example—Routing Protocol Summary

Connection	Protocols
CE3 - PE3	EBGP family inet
PE3 - P2 - CE1	OSPF and LDP

Table 4: Multiple-Instance LDP Example—Routing Protocol Summary *(continued)*

Connection	Protocols
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

Table 5: Multiple-Instance LDP Example—Loopback Addresses

Router	Loopback Address
PE1	10.255.255.171
PE2	10.255.255.172
P0	10.255.255.173
P1	10.255.255.174
P2	10.255.255.175
P3	10.255.255.176
PE3	10.255.255.177
PE4	10.255.255.178
CE1	10.255.255.179
CE2	10.255.255.180
CE3	10.255.255.181 10.49.100.1
CE4	10.255.255.182 10.49.200.1

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 200, 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

```
[edit]
interfaces {
   so-1/2/0 {
      description "to pe3 so-1/2/0";
      unit 0 {
         family inet {
            address 192.255.198.14/30;
         }
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.255.255.181/32;
            address 10.49.100.1/32;
         }
      }
   }
}
routing-options {
   static {
      route 10.49.100.0/24 reject;
      route 10.49.101.0/24 reject;
   }
   autonomous-system 100;
}
protocols {
   bgp {
      group provider {
         type external;
         export static-to-bgp;
         peer-as 200;
         neighbor 192.255.198.13;
      }
   }
}
policy-options {
   policy-statement static-to-bgp {
      term 1 {
         from {
            protocol static;
            route-filter 10.49.100.0/24 exact;
            route-filter 10.49.101.0/24 exact;
         }
         then accept;
      }
      term 2 {
         from protocol direct;
         then accept;
      }
      term 3 {
         then reject;
```

```
    }
  }
}
```

On PE3, the configuration tasks are more involved. You need to complete the EBGP connection to CE3 in a VRF instance, enable MPLS and LDP on the interface pointing toward the VPN provider CE1 router, and configure a master instance of IBGP to reach PE4 at the far edge of AS 200.

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 PE4.

Router PE3

```
[edit]
interfaces {
  so-1/2/0 {
    unit 0 {
      family inet {
        address 192.255.198.13/30;
      }
      family mpls;
    }
  }
  so-1/2/1 {
    description "to p2 so-1/2/1";
    unit 0 {
      family inet {
        address 192.255.198.9/30;
      }
      family mpls;
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.255.255.177/32;
      }
    }
  }
}
routing-options {
  autonomous-system 200;
}
protocols {
  mpls {
    interface so-1/2/0.0;
  }
  bgp {
    group internal {
      type internal;
      local-address 10.255.255.177;
      peer-as 200;
      neighbor 10.255.255.178 {
        family inet-vpn {
          unicast;
```

```
            }
          }
        }
      }
      ospf {
        area 0.0.0.0 {
          interface so-1/2/1.0;
          interface lo0.0 {
            passive;
          }
        }
      }
      ldp {
        interface so-1/2/1.0;
      }
}
policy-options {
      policy-statement vpn-customer-import {
        term 1 {
          from {
            protocol bgp;
            community vpn-customer-comm;
          }
          then accept;
        }
        term 2 {
          then reject;
        }
      }
      policy-statement vpn-customer-export {
        term 1 {
          from protocol [bgp direct];
          then {
            community add vpn-customer-comm;
            accept;
          }
        }
        term 2 {
          then reject;
        }
      }
      community vpn-customer-comm members target:200:100;
}
routing-instances {
      vpn-customer {
        instance-type vrf;
        interface so-1/2/0.0;
        route-distinguisher 10.255.255.177:1;
        vrf-import vpn-customer-import;
        vrf-export vpn-customer-export;
        protocols {
          bgp {
            group customer {
              type external;
              peer-as 100;
              as-override;
```

```
            neighbor 192.255.198.14;
          }
        }
      }
    }
}
```

On 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

```
[edit]
interfaces {
    so-1/2/0 {
        description "to ce1 so-1/2/0";
        unit 0 {
            family inet {
                address 192.255.198.2/30;
            }
            family mpls;
        }
    }
    so-1/2/1 {
        description "to pe3 so-1/2/1";
        unit 0 {
            family inet {
                address 192.255.198.10/30;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.255.175/32;
            }
        }
    }
}
routing-options {
    autonomous-system 200;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface lo0.0 {
                passive;
            }
            interface so-1/2/0.0;
            interface so-1/2/1.0;
        }
    }
    ldp {
        interface so-1/2/0.0;
        interface so-1/2/1.0;
```

```
    }
}
```

For Router CE1, configure LDP and OSPF in the same manner that you configured the P2 router.

Router CE1

```
[edit]
interfaces {
    t3-0/1/0 {
        description "to pe1 t3-0/2/1";
        unit 0 {
            family inet {
                address 192.255.197.18/30;
            }
            family mpls;
        }
    }
    so-1/2/0 {
        description "to p2 so-1/2/0";
        unit 0 {
            family inet {
                address 192.255.198.1/30;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.255.179/32;
            }
        }
    }
}
routing-options {
    autonomous-system 200;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-1/2/0.0;
            interface lo0.0 {
                passive;
            }
            interface t3-0/1/0.0;
        }
    }
    ldp {
        interface t3-0/1/0.0;
        interface so-1/2/0.0;
    }
}
```

On core carrier Router PE1, configure a master instance for OSPF, LDP, MPLS, and IBGP (with the family inet-vpn option) to connect the router to neighbor PE2. Next, implement multiple-instance LDP by establishing a secondary instance. Enable LDP and OSPF in this instance for PE1 to communicate with CE1. MPLS is not required in the secondary instance.

Finally, set up an outbound VRF policy that places all LDP traffic coming from CE1 into a BGP community, an export policy that sends this community traffic to PE2, and an inbound VRF policy that accepts similar BGP community traffic from PE2. This step tunnels the VPN provider's LDP traffic into the carrier's BGP session.

Router PE1

```
[edit]
interfaces {
   so-0/0/0 {
      description "to p0 so-0/1/0";
      unit 0 {
         family inet {
            address 192.255.197.21/30;
         }
         family mpls;
      }
   }
   t3-0/2/1 {
      description "to ce1 t3-0/1/0";
      unit 0 {
         family inet {
            address 192.255.197.17/30;
         }
         family mpls;
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.255.255.171/32;
         }
      }
   }
}
routing-options {
   autonomous-system 300;
}
protocols {
   mpls {
      interface t3-0/2/1.0;
   }
   bgp {
      group pe {
         type internal;
         local-address 10.255.255.171;
         family inet-vpn {
            unicast;
         }
         peer-as 300;
         neighbor 10.255.255.172;
      }
```

```
    }
    ospf {
      area 0.0.0.0 {
        interface lo0.0 {
          passive;
        }
        interface so-0/0/0.0;
      }
    }
    ldp {
      interface so-0/0/0.0;
    }
}
policy-options {
    policy-statement vpn-provider-import {
      term 1 {
        from {
          protocol bgp;
          community vpn-provider-comm;
        }
        then accept;
      }
      term 2 {
        then reject;
      }
    }
    policy-statement vpn-provider-export {
      term 1 {
        from protocol ldp;
        then {
          community add vpn-provider-comm;
          accept;
        }
      }
      term 2 {
        then reject;
      }
    }
    policy-statement bgp-routes-to-export {
      term 1 {
        from {
          protocol bgp;
          community vpn-provider-comm;
        }
        then accept;
      }
      term 2 {
        then reject;
      }
    }
    community vpn-provider-comm members target:300:200;
}
routing-instances {
    vpn-provider {
      instance-type vrf;
      interface t3-0/2/1.0;
```

```
        route-distinguisher 10.255.255.171:1;
        vrf-import vpn-provider-import;
        vrf-export vpn-provider-export;
        protocols {
            ospf {
                export bgp-routes-to-export;
                area 0.0.0.0 {
                    interface t3-0/2/1.0;
                }
            }
            ldp {
                egress-policy bgp-routes-to-export;
                interface t3-0/2/1.0;
            }
        }
    }
}
```

On P0, enable LDP and OSPF in the same manner that you configured these protocols on P2. You will repeat these tasks on routers P1 and P3.

Router P0

```
[edit]
interfaces {
    so-0/1/0 {
        description "to pe1 so-0/0/0";
        unit 0 {
            family inet {
                address 192.255.197.22/30;
            }
            family mpls;
        }
    }
    so-1/0/0 {
        description "to p1 so-1/0/0";
        unit 0 {
            family inet {
                address 192.255.197.85/30;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.255.173/32;
            }
        }
    }
}
routing-options {
    autonomous-system 300;
}
protocols {
    ospf {
        area 0.0.0.0 {
```

```
            interface so-0/1/0.0;
            interface so-1/0/0.0;
            interface lo0.0 {
               passive;
            }
         }
      }
      ldp {
         interface so-0/1/0.0;
         interface so-1/0/0.0;
      }
   }
```

On P1, enable LDP and the IGP used for transporting labels (OSPF in this case).

Router P1

```
[edit]
interfaces {
   so-0/0/0 {
      description "to pe2 so-0/2/0";
      unit 0 {
         family inet {
            address 192.255.197.74/30;
         }
         family mpls;
      }
   }
   so-1/0/0 {
      description "to p0 so-1/0/0";
      unit 0 {
         family inet {
            address 192.255.197.86/30;
         }
         family mpls;
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.255.255.174/32;
         }
      }
   }
}
routing-options {
   autonomous-system 300;
}
protocols {
   ospf {
      area 0.0.0.0 {
         interface so-0/0/0.0;
         interface so-1/0/0.0;
         interface lo0.0 {
            passive;
         }
      }
```

```
    }
    ldp {
      interface so-0/0/0.0;
      interface so-1/0/0.0;
    }
  }
```

Core carrier Router PE2 is a mirror image of PE1. First, configure a master instance for OSPF, LDP, MPLS, and IBGP (with the family inet-vpn option) to connect PE2 to neighbor PE1. Next, implement multiple-instance LDP by establishing a secondary instance. Enable LDP and OSPF in this instance for PE2 to communicate with CE2. MPLS is not required in the secondary instance.

Finally, set up an outbound VRF policy that places all LDP traffic coming from CE2 into a BGP community, an export policy that sends this community traffic to PE1, and an inbound VRF policy that accepts similar BGP community traffic from PE1. This step tunnels the VPN provider's LDP traffic into the carrier's BGP session.

Router PE2

```
[edit]
interfaces {
  so-0/2/0 {
    description "to p1 so-0/0/0";
    unit 0 {
      family inet {
        address 192.255.197.73/30;
      }
      family mpls;
    }
  }
  t1-3/0/0 {
    description "to ce2 t1-0/0/0";
    unit 0 {
      family inet {
        address 192.255.197.37/30;
      }
      family mpls;
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.255.255.172/32;
      }
    }
  }
}
routing-options {
  autonomous-system 300;
}
protocols {
  mpls {
    interface t1-3/0/0.0;
  }
  bgp {
    group pe {
```

```
            type internal;
            local-address 10.255.255.172;
            family inet-vpn {
               unicast;
            }
            peer-as 300;
            neighbor 10.255.255.171;
         }
      }
      ospf {
         area 0.0.0.0 {
            interface so-0/2/0.0;
            interface lo0.0 {
               passive;
            }
         }
      }
      ldp {
         interface so-0/2/0.0;
      }
   }
   policy-options {
      policy-statement vpn-provider-import {
         term 1 {
            from {
               protocol bgp;
               community vpn-provider-comm;
            }
            then accept;
         }
         term 2 {
            then reject;
         }
      }
      policy-statement vpn-provider-export {
         term 1 {
            from protocol ldp;
            then {
               community add vpn-provider-comm;
               accept;
            }
         }
         term 2 {
            then reject;
         }
      }
      policy-statement bgp-routes-to-export {
         term 1 {
            from {
               protocol bgp;
               community vpn-provider-comm;
            }
            then accept;
         }
         term 2 {
            then reject;
```

```
        }
      }
      community vpn-provider-comm members target:300:200;
    }
    routing-instances {
      vpn-provider {
        instance-type vrf;
        interface t1-3/0/0.0;
        route-distinguisher 10.255.255.172:1;
        vrf-import vpn-provider-import;
        vrf-export vpn-provider-export;
        protocols {
          ospf {
            export bgp-routes-to-export;
            area 0.0.0.0 {
              interface t1-3/0/0.0;
            }
          }
          ldp {
            egress-policy bgp-routes-to-export;
            interface t1-3/0/0.0;
          }
        }
      }
    }
```

For Router CE2, configure LDP and OSPF as you did on CE1 and the transit P routers.

Router CE2

```
[edit]
interfaces {
  t1-0/0/0 {
    description "to pe2 t1-3/0/0";
    unit 0 {
      family inet {
        address 192.255.197.38/30;
      }
      family mpls;
    }
  }
  t3-0/3/3 {
    description "to p3 t3-0/0/3";
    unit 0 {
      family inet {
        address 192.255.198.26/30;
      }
      family mpls;
    }
    lo0 {
      unit 0 {
        family inet {
          address 10.255.255.180/32;
        }
      }
    }
  }
```

```
    routing-options {
        autonomous-system 200;
    }
    protocols {
        ospf {
            area 0.0.0.0 {
                interface t1-0/0/0.0;
                interface t3-0/3/3.0;
                interface lo0.0 {
                    passive;
                }
            }
        }
        ldp {
            interface t1-0/0/0.0;
            interface t3-0/3/3.0;
        }
    }
}
```

Since P3 is another core provider router, enable LDP and OSPF on all transit interfaces.

Router P3

```
[edit]
interfaces {
    t3-0/0/3 {
        description "to ce2 t3-0/3/3";
        unit 0 {
            family inet {
                address 192.255.198.25/30;
            }
            family mpls;
        }
    }
    t1-0/1/1 {
        description "to pe4 t1-0/1/1";
        unit 0 {
            family inet {
                address 192.255.198.37/30;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.255.176/32;
            }
        }
    }
}
routing-options {
    autonomous-system 200;
}
protocols {
```

```
    ospf {
      area 0.0.0.0 {
        interface t3-0/0/3.0;
        interface t1-0/1/1.0;
        interface lo0.0 {
          passive;
        }
      }
    }
    ldp {
      interface t3-0/0/3.0;
      interface t1-0/1/1.0;
    }
}
```

On PE4, complete the IBGP connection initiated on PE3 to connect the edge routers in AS 200. Also, enable LDP and MPLS on the t1-0/0/1 interface pointing toward the VPN provider CE2 router and establish an EBGP connection to 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 PE3.

Router PE4

```
[edit]
interfaces {
    t3-0/0/3 {
        description to ce4 t3-0/0/3";
        unit 0 {
            family inet {
                address 192.255.198.21/30;
            }
            family mpls;
        }
    }
    t1-0/1/1 {
        unit 0 {
            family inet {
                address 192.255.198.38/30;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.255.178/32;
            }
        }
    }
}
routing-options {
    autonomous-system 200;
}
protocols {
```

```
    mpls {
        interface t3-0/0/3.0;
    }
    bgp {
        group internal {
            type internal;
            local-address 10.255.255.178;
            peer-as 200;
            neighbor 10.255.255.177 {
                family inet-vpn {
                    unicast;
                }
            }
        }
    }
    ospf {
        area 0.0.0.0 {
            interface t1-0/1/1.0;
            interface lo0.0 {
                passive;
            }
        }
    }
    ldp {
        interface t1-0/1/1.0;
    }
}
policy-options {
    policy-statement vpn-customer-import {
        term 1 {
            from {
                protocol bgp;
                community vpn-customer-comm;
            }
            then accept;
        }
        term 2 {
            then reject;
        }
    }
    policy-statement vpn-customer-export {
        term 1 {
            from protocol [bgp direct];
            then {
                community add vpn-customer-comm;
                accept;
            }
        }
        term 2 {
            then reject;
        }
    }
    community vpn-customer-comm members target:200:100;
}
routing-instances {
    vpn-customer {
```

```
        instance-type vrf;
        interface t3-0/0/3.0;
        route-distinguisher 10.255.255.178:1;
        vrf-import vpn-customer-import;
        vrf-export vpn-customer-export;
        protocols {
            bgp {
                group customer {
                    type external;
                    peer-as 100;
                    as-override;
                    neighbor 192.255.198.22;
                }
            }
        }
    }
}
```

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

```
[edit]
interfaces {
    t3-0/0/3 {
        description "to pe4 t3-0/0/3";
        unit 0 {
            family inet {
                address 192.255.198.22/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.255.182/32;
                address 10.49.200.1/32;
            }
        }
    }
}
routing-options {
    static {
        route 10.49.200.0/24 reject;
        route 10.49.201.0/24 reject;
    }
    autonomous-system 100;
}
protocols {
    bgp {
        group provider {
            type external;
            export static-to-bgp;
            peer-as 200;
```

```
            neighbor 192.255.198.21;
        }
    }
}
policy-options {
    policy-statement static-to-bgp {
        term 1 {
            from {
                protocol static;
                route-filter 10.49.200.0/24 exact;
                route-filter 10.49.201.0/24 exact;
            }
            then accept;
        }
        term 2 {
            from protocol direct;
            then accept;
        }
        term 3 {
            then reject;
        }
    }
}
```

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 Software 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 master instance.

The following sections show the output of these commands used with the configuration example:

- Router CE3 Status on page 83
- Router PE3 Status on page 83
- Router CE1 Status on page 85
- Router PE1 Status on page 87
- Router PE2 Status on page 89
- Router CE2 Status on page 94
- Router PE4 Status on page 95
- Router CE4 Status on page 97

Router CE3 Status

```
user@CE3> show bgp summary

Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0               10          5          0          0          0          0
Peer               AS      InPkt     OutPkt    OutQ   Flaps Last Up/DwnState|#Active/Received/Damped...
192.255.198.13    200        440        433       0       0     3:34:34 5/10/0 0/0/0

user@CE3>  show route protocol bgp
inet.0: 23 destinations, 28 routes (22 active, 0 holddown, 6 hidden)
+ = Active Route, - = Last Active, * = Both
10.49.200.0/24     *[BGP/170] 00:19:20, localpref 100
                      AS path: 200 200 I
                    > to 192.255.198.13 via so-1/2/0.0
10.49.200.1/32     *[BGP/170] 00:19:20, localpref 100
                      AS path: 200 200 I
                    > to 192.255.198.13 via so-1/2/0.0
10.49.201.0/24     *[BGP/170] 00:19:20, localpref 100
                      AS path: 200 200 I
                    > to 192.255.198.13 via so-1/2/0.0
10.255.255.182/32  *[BGP/170] 00:19:20, localpref 100
                      AS path: 200 200 I
                    > to 192.255.198.13 via so-1/2/0.0
192.255.198.20/30  *[BGP/170] 00:19:20, localpref 100
                      AS path: 200 I
                    > to 192.255.198.13 via so-1/2/0.0
```

Router PE3 Status

```
user@PE3> show bgp summary

Groups: 2 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l3vpn.0            6          6          0          0          0          0
Peer               AS      InPkt     OutPkt    OutQ   Flaps Last Up/DwnState|#Active/Received/Damped...
192.255.198.14    100        432        441       0       0     3:34:55 Establ
  vpn-customer.inet.0: 5/6/0
```

```
10.255.255.178    200         62         63       0       2       27:23 Establ
  bgp.l3vpn.0: 6/6/0
  vpn-customer.inet.0: 5/6/0

user@PE3>  show route protocol ldp
inet.0: 19 destinations, 20 routes (18 active, 0 holddown, 1 hidden)
inet.3: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.255.255.175/32  *[LDP/9] 03:35:45, metric 1
                    > via so-1/2/1.0
10.255.255.176/32  *[LDP/9] 00:29:32, metric 1
                    > via so-1/2/1.0, Push 100007
10.255.255.178/32  *[LDP/9] 00:29:32, metric 1
                    > via so-1/2/1.0, Push 100008
10.255.255.179/32  *[LDP/9] 03:34:39, metric 1
                    > via so-1/2/1.0, Push 100001
10.255.255.180/32  *[LDP/9] 03:31:15, metric 1
                    > via so-1/2/1.0, Push 100002
vpn-customer.inet.0: 12 destinations, 14 routes (12 active, 0 holddown, 0 hidden)
mpls.0: 13 destinations, 13 routes (13 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
100000             *[LDP/9] 03:35:45, metric 1
                    > via so-1/2/1.0, Pop
100000(S=0)        *[LDP/9] 03:35:45, metric 1
                    > via so-1/2/1.0, Pop
100001             *[LDP/9] 03:34:39, metric 1
                    > via so-1/2/1.0, Swap 100001
100002             *[LDP/9] 03:31:15, metric 1
                    > via so-1/2/1.0, Swap 100002
100011             *[LDP/9] 00:29:32, metric 1
                    > via so-1/2/1.0, Swap 100007
100012             *[LDP/9] 00:29:32, metric 1
                    > via so-1/2/1.0, Swap 100008
bgp.l3vpn.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)

user@PE3>  show route protocol bgp
inet.0: 19 destinations, 20 routes (18 active, 0 holddown, 1 hidden)
inet.3: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
vpn-customer.inet.0: 12 destinations, 14 routes (12 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.49.100.0/24     *[BGP/170] 03:34:59, MED 0, localpref 100
                      AS path: 100 I
                    > to 192.255.198.14 via so-1/2/0.0
10.49.100.1/32     *[BGP/170] 03:34:59, localpref 100
                      AS path: 100 I
                    > to 192.255.198.14 via so-1/2/0.0
10.49.101.0/24     *[BGP/170] 03:34:59, MED 0, localpref 100
                      AS path: 100 I
                    > to 192.255.198.14 via so-1/2/0.0
10.49.200.0/24     *[BGP/170] 00:26:39, MED 0, localpref 100, from 10.255.255.178
                      AS path: 100 I
                    > via so-1/2/1.0, Push 100019, Push 100008(top)
10.49.200.1/32     *[BGP/170] 00:26:39, localpref 100, from 10.255.255.178
                      AS path: 100 I
                    > via so-1/2/1.0, Push 100019, Push 100008(top)
10.49.201.0/24     *[BGP/170] 00:26:39, MED 0, localpref 100, from 10.255.255.178
                      AS path: 100 I
                    > via so-1/2/1.0, Push 100019, Push 100008(top)
10.255.255.181/32  *[BGP/170] 03:34:59, localpref 100
                      AS path: 100 I
                    > to 192.255.198.14 via so-1/2/0.0
```

```
10.255.255.182/32  *[BGP/170] 00:26:39, localpref 100, from 10.255.255.178
                      AS path: 100 I
                    > via so-1/2/1.0, Push 100019, Push 100008(top)
192.255.14.0/24    *[BGP/170] 03:34:59, localpref 100
                      AS path: 100 I
                    > to 192.255.198.14 via so-1/2/0.0
                    [BGP/170] 00:26:39, localpref 100, from 10.255.255.178
                      AS path: 100 I
                    > via so-1/2/1.0, Push 100019, Push 100008(top)
192.255.198.12/30   [BGP/170] 03:34:59, localpref 100
                      AS path: 100 I
                    > to 192.255.198.14 via so-1/2/0.0
192.255.198.20/30  *[BGP/170] 00:26:39, localpref 100, from 10.255.255.178
                      AS path: I
                    > via so-1/2/1.0, Push 100020, Push 100008(top)
mpls.0: 13 destinations, 13 routes (13 active, 0 holddown, 0 hidden)
bgp.l3vpn.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.255.255.178:1:10.49.200.0/24
                   *[BGP/170] 00:27:27, MED 0, localpref 100, from 10.255.255.178
                      AS path: 100 I
                    > via so-1/2/1.0, Push 100019, Push 100008(top)
10.255.255.178:1:10.49.200.1/32
                   *[BGP/170] 00:27:27, localpref 100, from 10.255.255.178
                      AS path: 100 I
                    > via so-1/2/1.0, Push 100019, Push 100008(top)
10.255.255.178:1:10.49.201.0/24
                   *[BGP/170] 00:27:27, MED 0, localpref 100, from 10.255.255.178
                      AS path: 100 I
                    > via so-1/2/1.0, Push 100019, Push 100008(top)
10.255.255.178:1:10.255.255.182/32
                   *[BGP/170] 00:27:27, localpref 100, from 10.255.255.178
                      AS path: 100 I
                    > via so-1/2/1.0, Push 100019, Push 100008(top)
10.255.255.178:1:192.255.14.0/24
                   *[BGP/170] 00:27:27, localpref 100, from 10.255.255.178
                      AS path: 100 I
                    > via so-1/2/1.0, Push 100019, Push 100008(top)
10.255.255.178:1:192.255.198.20/30
                   *[BGP/170] 00:27:27, localpref 100, from 10.255.255.178
                      AS path: I
                    > via so-1/2/1.0, Push 100020, Push 100008(top)
```

Router CE1 Status

```
user@CE1> show ldp neighbor
Address            Interface          Label space ID         Hold time
192.255.197.17     t3-0/1/0.0         192.255.197.17:0         11
192.255.198.2      so-1/2/0.0         10.255.255.175:0         14

user@CE1>  show route

inet.0: 21 destinations, 23 routes (20 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both
0.0.0.0/0          *[Static/5] 07:53:10, metric 0
                       Discard
10.255.255.175/32  *[OSPF/10] 00:31:44, metric 1
                    > via so-1/2/0.0
```

```
10.255.255.176/32  *[OSPF/150] 00:31:44, metric 1, tag 3489661228
                    > via t3-0/1/0.0
10.255.255.177/32  *[OSPF/10] 00:31:44, metric 2
                    > via so-1/2/0.0
10.255.255.178/32  *[OSPF/150] 00:31:44, metric 1, tag 3489661228
                    > via t3-0/1/0.0
10.255.255.179/32  *[Direct/0] 07:53:10
                    > via lo0.0
10.255.255.180/32  *[OSPF/150] 00:31:44, metric 1, tag 3489661228
                    > via t3-0/1/0.0
172.16.0.0/12      *[Static/5] 07:53:10
                    > to 192.255.14.254 via fxp0.0
192.255.0.0/18     *[Static/5] 07:53:10
                    > to 192.255.14.254 via fxp0.0
192.255.14.0/24    *[Direct/0] 07:53:10
                    > via fxp0.0
192.255.14.179/32  *[Local/0] 07:53:10
                      Local via fxp0.0
192.255.40.0/22    *[Static/5] 03:38:37
                    > to 192.255.14.254 via fxp0.0
192.255.64.0/18    *[Static/5] 03:38:37
                    > to 192.255.14.254 via fxp0.0
192.255.197.16/30  *[Direct/0] 03:37:42
                    > via t3-0/1/0.0
                    [OSPF/10] 00:31:44, metric 2
                    > via t3-0/1/0.0
192.255.197.18/32  *[Local/0] 07:52:01
                      Local via t3-0/1/0.0
192.255.198.0/30   *[Direct/0] 07:51:18
                    > via so-1/2/0.0
                    [OSPF/10] 00:31:44, metric 1
                    > via so-1/2/0.0
192.255.198.1/32   *[Local/0] 07:51:59
                      Local via so-1/2/0.0
192.255.198.8/30   *[OSPF/10] 00:31:44, metric 2
                    > via so-1/2/0.0
207.17.136.192/32  *[Static/5] 07:53:10
                    > to 192.255.14.254 via fxp0.0
224.0.0.5/32       *[OSPF/10] 07:53:14, metric 1
                      MultiRecv
inet.3: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.255.255.175/32  *[LDP/9] 01:00:52, metric 1
                    > via so-1/2/0.0
10.255.255.176/32  *[LDP/9] 00:33:24, metric 1
                    > via t3-0/1/0.0, Push 100020
10.255.255.177/32  *[LDP/9] 01:00:52, metric 1
                    > via so-1/2/0.0, Push 100000
10.255.255.178/32  *[LDP/9] 00:33:24, metric 1
                    > via t3-0/1/0.0, Push 100021
10.255.255.180/32  *[LDP/9] 01:00:52, metric 1
                    > via t3-0/1/0.0, Push 100015
mpls.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
100000             *[LDP/9] 03:38:31, metric 1
                    > via so-1/2/0.0, Pop
100000(S=0)        *[LDP/9] 03:38:31, metric 1
                    > via so-1/2/0.0, Pop
100001             *[LDP/9] 03:38:31, metric 1
                    > via so-1/2/0.0, Swap 100000
100002             *[LDP/9] 03:35:06, metric 1
```

```
                    > via t3-0/1/0.0, Swap 100015
100007              *[LDP/9] 00:33:24, metric 1
                    > via t3-0/1/0.0, Swap 100020
100008              *[LDP/9] 00:33:24, metric 1
                    > via t3-0/1/0.0, Swap 100021
```

Router PE1 Status

```
user@PE1> show ldp neighbor instance vpn-provider
Address            Interface          Label space ID         Hold time
192.255.197.18     t3-0/2/1.0         10.255.255.179:0         11

user@PE1>  show ldp database instance vpn-provider
Input label database, 192.255.197.17:0--10.255.255.179:0
  Label     Prefix
      3     10.255.255.179/32
 100002     10.255.255.180/32
 100007     10.255.255.176/32
 100001     10.255.255.177/32
 100008     10.255.255.178/32
 100000     10.255.255.175/32
Output label database, 192.255.197.17:0--10.255.255.179:0
  Label     Prefix
 100007     10.255.255.175/32
 100020     10.255.255.176/32
 100008     10.255.255.177/32
 100021     10.255.255.178/32
 100006     10.255.255.179/32
 100015     10.255.255.180/32

user@PE1>  show ldp interface instance vpn-provider
Interface            Label space ID        Nbr count   Next hello
t3-0/2/1.0           192.255.197.17:0         1           0
user@PE1>  show ldp path instance vpn-provider
Output Session (label)          Input Session (label)
10.255.255.179:0(100006)(                          ) 10.255.255.179:0(3)( )
10.255.255.179:0(100007)        10.255.255.179:0(100000)
10.255.255.179:0(100008)        10.255.255.179:0(100001)
10.255.255.179:0(100015)        ( )
10.255.255.179:0(100020)        ( )
10.255.255.179:0(100021)        ( )

user@PE1>  show ldp route instance vpn-provider
Destination         Next-hop intf/lsp               Next-hop address
 10.255.255.175/32  t3-0/2/1.0
 10.255.255.176/32  so-0/0/0.0
 10.255.255.177/32  t3-0/2/1.0
 10.255.255.178/32  so-0/0/0.0
 10.255.255.179/32  t3-0/2/1.0
 10.255.255.180/32  so-0/0/0.0
 192.255.197.16/30  t3-0/2/1.0
 192.255.197.17/32
 192.255.198.0/30   t3-0/2/1.0
 192.255.198.8/30   t3-0/2/1.0
 224.0.0.5/32

user@PE1>  show ldp session instance vpn-provider
  Address           State        Connection     Hold time
10.255.255.179      Operational  Open             24
```

```
user@PE1>  show ldp statistics instance vpn-provider
Message type              Total                    Last 5 seconds
                      Sent      Received          Sent      Received
Hello                 2838          2839             1             2
Initialization           1             1             0             0
Keepalive             1240          1239             0             0
Notification             0             0             0             0
Address                  1             1             0             0
Address withdraw         0             0             0             0
Label mapping           10            10             0             0
Label request            0             0             0             0
Label withdraw           4             4             0             0
Label release            4             4             0             0
Label abort              0             0             0             0
All UDP               2837          2839             1             2
All TCP               1258          1251             0             0
Event type                          Total        Last 5 seconds
Sessions opened                         1                 0
Sessions closed                         0                 0
Topology changes                       21                 0
No router id                            0                 0
No address                              0                 0
No interface                            0                 0
No session                              0                 0
No adjacency                            0                 0
Unknown version                         0                 0
Malformed PDU                           0                 0
Malformed message                       0                 0
Unknown message type                    0                 0
Inappropriate message                   0                 0
Malformed TLV                           0                 0
Bad TLV value                           0                 0
Missing TLV                             0                 0
PDU too large                           0                 0
PDU too small                           0                 0

user@PE1>  show ldp traffic-statistics instance vpn-provider
FEC                 Type            Packets          Bytes    Shared
 10.255.255.175/32  Transit               0              0    No
 10.255.255.175/32  Ingress               0              0    No
 10.255.255.176/32  Transit               0              0    No
 10.255.255.177/32  Transit            2798         241984    No
 10.255.255.177/32  Ingress               0              0    No
 10.255.255.178/32  Transit            1365         125580    No
 10.255.255.179/32  Transit               0              0    No
 10.255.255.179/32  Ingress            2427         149076    No
 10.255.255.180/32  Transit               0              0    No

user@PE1>  show bgp summary
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l3vpn.0            3          3          0          0          0          0
Peer               AS      InPkt     OutPkt    OutQ   Flaps Last
Up/DwnState|#Active/Received/Damped...
10.255.255.172    300        428        422       0       0     3:28:37 Establ
  bgp.l3vpn.0: 3/3/0
  vpn-provider.inet.0: 3/3/0
```

Router PE2 Status

```
user@PE2> show ldp neighbor instance vpn-provider
Address            Interface          Label space ID         Hold time
192.255.197.38     t1-3/0/0.0         10.255.255.180:0         11

user@PE2>  show ldp database instance vpn-provider
Input label database, 192.255.197.37:0--10.255.255.180:0
  Label     Prefix
      3     10.255.255.180/32
 100003     10.255.255.177/32
 100010     10.255.255.178/32
 100009     10.255.255.176/32
 100002     10.255.255.175/32
 100004     10.255.255.179/32
Output label database, 192.255.197.37:0--10.255.255.180:0
  Label     Prefix
 100026     10.255.255.175/32
 100028     10.255.255.179/32
 100027     10.255.255.177/32
 100021     10.255.255.180/32
 100039     10.255.255.178/32
 100037     10.255.255.176/32

user@PE2>  show ldp interface instance vpn-provider
Interface            Label space ID        Nbr count   Next hello
t1-3/0/0.0           192.255.197.37:0
        1           1

user@PE2>  show ldp path instance vpn-provider
Output Session (label)          Input Session (label)
10.255.255.180:0(100021)(                          ) 10.255.255.180:0(3)( )
10.255.255.180:0(100026)        ( )
10.255.255.180:0(100027)        ( )
10.255.255.180:0(100028)        ( )
10.255.255.180:0(100037)        10.255.255.180:0(100009)
10.255.255.180:0(100039)        10.255.255.180:0(100010)

user@PE2>  show ldp route instance vpn-provider
Destination         Next-hop intf/lsp               Next-hop address
 10.255.255.175/32  so-0/2/0.0
 10.255.255.176/32  t1-3/0/0.0
 10.255.255.177/32  so-0/2/0.0
 10.255.255.178/32  t1-3/0/0.0
 10.255.255.179/32  so-0/2/0.0
 10.255.255.180/32  t1-3/0/0.0
 192.255.197.36/30  t1-3/0/0.0
 192.255.197.37/32
 192.255.198.24/30  t1-3/0/0.0
 192.255.198.36/30  t1-3/0/0.0
 224.0.0.5/32

user@PE2>  show ldp session instance vpn-provider
  Address           State        Connection     Hold time
10.255.255.180      Operational  Open             29

user@PE2>  show ldp statistics instance vpn-provider
Message type              Total                     Last 5 seconds
```

```
                    Sent      Received          Sent      Received
Hello               2948          2939             1             1
Initialization         1             1             0             0
Keepalive           1285          1285             0             0
Notification           0             0             0             0
Address                1             1             0             0
Address withdraw       0             0             0             0
Label mapping         10            10             0             0
Label request          0             0             0             0
Label withdraw         4             4             0             0
Label release          4             4             0             0
Label abort            0             0             0             0
All UDP             2947          2939             1             1
All TCP             1297          1299             0             0
Event type                          Total       Last 5 seconds
Sessions opened                         1                    0
Sessions closed                         0                    0
Topology changes                       33                    0
No router id                            0                    0
No address                              0                    0
No interface                            0                    0
No session                              0                    0
No adjacency                            0                    0
Unknown version                         0                    0
Malformed PDU                           0                    0
Malformed message                       0                    0
Unknown message type                    0                    0
Inappropriate message                   0                    0
Malformed TLV                           0                    0
Bad TLV value                           0                    0
Missing TLV                             0                    0
PDU too large                           0                    0
PDU too small
                            0                    0
user@PE2>  show ldp traffic-statistics instance vpn-provider
FEC                 Type            Packets          Bytes    Shared
 10.255.255.175/32  Transit               0              0    No
 10.255.255.176/32  Transit               0              0    No
 10.255.255.176/32  Ingress               0              0    No
 10.255.255.177/32  Transit            3131         274830    No
 10.255.255.178/32  Transit            1966         178256    No
 10.255.255.178/32  Ingress               0              0    No
 10.255.255.179/32  Transit               1             44    No
 10.255.255.180/32  Transit               0              0    No
 10.255.255.180/32  Ingress            2330         144838    No

user@PE2>  show bgp summary

Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 0          0          0          0          0          0
bgp.l3vpn.0            3          3          0          0          0          0
Peer               AS      InPkt     OutPkt    OutQ   Flaps Last
Up/DwnState|#Active/Received/Damped...
10.255.255.171    300        429        438       0       0     3:33:32 Establ
  bgp.l3vpn.0: 3/3/0
  vpn-provider.inet.0: 3/3/0

user@PE2>  show route protocol bgp
inet.0: 18 destinations, 19 routes (17 active, 0 holddown, 1 hidden)
inet.3: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
```

```
vpn-provider.inet.0: 11 destinations, 15 routes (11 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.255.255.175/32  *[BGP/170] 00:27:59, MED 1, localpref 100, from 10.255.255.171

                     AS path: I
                    > via so-0/2/0.0, Push 100012, Push 100028(top)
10.255.255.177/32  *[BGP/170] 00:27:59, MED 1, localpref 100, from 10.255.255.171

                     AS path: I
                    > via so-0/2/0.0, Push 100013, Push 100028(top)
10.255.255.179/32  *[BGP/170] 00:27:59, MED 1, localpref 100, from 10.255.255.171

                     AS path: I
                    > via so-0/2/0.0, Push 100014, Push 100028(top)
vpn-provider.inet.3: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
mpls.0: 11 destinations, 11 routes (11 active, 0 holddown, 0 hidden)
vpn-provider.mpls.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
bgp.l3vpn.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.255.255.171:1:10.255.255.175/32
                   *[BGP/170] 03:33:34, MED 1, localpref 100, from 10.255.255.171

                     AS path: I
                    > via so-0/2/0.0, Push 100012, Push 100028(top)
10.255.255.171:1:10.255.255.177/32
                   *[BGP/170] 03:33:34, MED 1, localpref 100, from 10.255.255.171

                     AS path: I
                    > via so-0/2/0.0, Push 100013, Push 100028(top)
10.255.255.171:1:10.255.255.179/32
                   *[BGP/170] 03:33:34, MED 1, localpref 100, from 10.255.255.171

                     AS path: I
                    > via so-0/2/0.0, Push 100014, Push 100028(top)
Address            Interface          Label space ID         Hold time
192.255.197.38     t1-3/0/0.0         10.255.255.180:0         11

user@PE2>  show ldp database instance vpn-provider
Input label database, 192.255.197.37:0--10.255.255.180:0
  Label     Prefix
      3     10.255.255.180/32
 100003     10.255.255.177/32
 100010     10.255.255.178/32
 100009     10.255.255.176/32
 100002     10.255.255.175/32
 100004     10.255.255.179/32
Output label database, 192.255.197.37:0--10.255.255.180:0
  Label     Prefix
 100026     10.255.255.175/32
 100028     10.255.255.179/32
 100027     10.255.255.177/32
 100021     10.255.255.180/32
 100039     10.255.255.178/32
 100037     10.255.255.176/32

user@PE2>  show ldp interface instance vpn-provider
Interface            Label space ID        Nbr count   Next hello
t1-3/0/0.0           192.255.197.37:0         1           1

user@PE2>  show ldp path instance vpn-provider
Output Session (label)          Input Session (label)
```

```
10.255.255.180:0(100021)(                          ) 10.255.255.180:0(3)( )
10.255.255.180:0(100026)        ( )
10.255.255.180:0(100027)        ( )
10.255.255.180:0(100028)        ( )
10.255.255.180:0(100037)        10.255.255.180:0(100009)
10.255.255.180:0(100039)        10.255.255.180:0(100010)

user@PE2>  show ldp route instance vpn-provider
Destination         Next-hop intf/lsp               Next-hop address
 10.255.255.175/32  so-0/2/0.0
 10.255.255.176/32  t1-3/0/0.0
 10.255.255.177/32  so-0/2/0.0
 10.255.255.178/32  t1-3/0/0.0
 10.255.255.179/32  so-0/2/0.0
 10.255.255.180/32  t1-3/0/0.0
 192.255.197.36/30  t1-3/0/0.0
 192.255.197.37/32
 192.255.198.24/30  t1-3/0/0.0
 192.255.198.36/30  t1-3/0/0.0
 224.0.0.5/32

user@PE2>  show ldp session instance vpn-provider
  Address           State        Connection     Hold time
10.255.255.180      Operational  Open             29

user@PE2>  show ldp statistics instance vpn-provider
Message type              Total                    Last 5 seconds
                     Sent      Received          Sent      Received
Hello                2948          2939            1             1
Initialization          1             1            0             0
Keepalive            1285          1285            0             0
Notification            0             0            0             0
Address                 1             1            0             0
Address withdraw        0             0            0             0
Label mapping          10            10            0             0
Label request           0             0            0             0
Label withdraw          4             4            0             0
Label release           4             4            0             0
Label abort             0             0            0             0
All UDP              2947          2939            1             1
All TCP              1297          1299            0             0
Event type                          Total        Last 5 seconds
Sessions opened                         1                 0
Sessions closed                         0                 0
Topology changes                       33                 0
No router id                            0                 0
No address                              0                 0
No interface                            0                 0
No session                              0                 0
No adjacency                            0                 0
Unknown version                         0                 0
Malformed PDU                           0                 0
Malformed message                       0                 0
Unknown message type                    0                 0
Inappropriate message                   0                 0
Malformed TLV                           0                 0
Bad TLV value                           0                 0
Missing TLV                             0                 0
PDU too large                           0                 0
PDU too small                           0                 0
```

```
user@PE2>  show ldp traffic-statistics instance vpn-provider
FEC                 Type            Packets          Bytes    Shared
 10.255.255.175/32  Transit               0              0    No
 10.255.255.176/32  Transit               0              0    No
 10.255.255.176/32  Ingress               0              0    No
 10.255.255.177/32  Transit            3131         274830    No
 10.255.255.178/32  Transit            1966         178256    No
 10.255.255.178/32  Ingress               0              0    No
 10.255.255.179/32  Transit               1             44    No
 10.255.255.180/32  Transit               0              0    No
 10.255.255.180/32  Ingress            2330         144838    No

user@PE2>  show bgp summary
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 0          0          0          0          0          0
bgp.l3vpn.0            3          3          0          0          0          0
Peer               AS      InPkt     OutPkt    OutQ   Flaps Last
Up/DwnState|#Active/Received/Damped...
10.255.255.171    300        429        438       0       0     3:33:32 Establ
  bgp.l3vpn.0: 3/3/0
  vpn-provider.inet.0: 3/3/0

user@PE2>  show route protocol bgp
inet.0: 18 destinations, 19 routes (17 active, 0 holddown, 1 hidden)
inet.3: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
vpn-provider.inet.0: 11 destinations, 15 routes (11 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.255.255.175/32  *[BGP/170] 00:27:59, MED 1, localpref 100, from 10.255.255.171

                     AS path: I
                    > via so-0/2/0.0, Push 100012, Push 100028(top)
10.255.255.177/32  *[BGP/170] 00:27:59, MED 1, localpref 100, from 10.255.255.171

                     AS path: I
                    > via so-0/2/0.0, Push 100013, Push 100028(top)
10.255.255.179/32  *[BGP/170] 00:27:59, MED 1, localpref 100, from 10.255.255.171

                     AS path: I
                    > via so-0/2/0.0, Push 100014, Push 100028(top)
vpn-provider.inet.3: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
mpls.0: 11 destinations, 11 routes (11 active, 0 holddown, 0 hidden)
vpn-provider.mpls.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
bgp.l3vpn.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.255.255.171:1:10.255.255.175/32
                   *[BGP/170] 03:33:34, MED 1, localpref 100, from 10.255.255.171

                     AS path: I
                    > via so-0/2/0.0, Push 100012, Push 100028(top)
10.255.255.171:1:10.255.255.177/32
                   *[BGP/170] 03:33:34, MED 1, localpref 100, from 10.255.255.171

                     AS path: I
                    > via so-0/2/0.0, Push 100013, Push 100028(top)
10.255.255.171:1:10.255.255.179/32
                   *[BGP/170] 03:33:34, MED 1, localpref 100, from 10.255.255.171

                     AS path: I
                    > via so-0/2/0.0, Push 100014, Push 100028(top)
```

Router CE2 Status

```
user@CE2> show ldp neighbor
Address            Interface          Label space ID         Hold time
192.255.197.37     t1-0/0/0.0         192.255.197.37:0         12
192.255.198.25     t3-0/3/3.0         10.255.255.176:0         13

user@CE2>  show route
inet.0: 21 destinations, 23 routes (20 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both
0.0.0.0/0          *[Static/5] 07:53:49, metric 0
                      Discard
10.255.255.175/32  *[OSPF/150] 00:29:56, metric 1, tag 3489661228
                    > via t1-0/0/0.0
10.255.255.176/32  *[OSPF/10] 00:29:56, metric 2
                    > via t3-0/3/3.0
10.255.255.177/32  *[OSPF/150] 00:29:56, metric 1, tag 3489661228
                    > via t1-0/0/0.0
10.255.255.178/32  *[OSPF/10] 00:29:56, metric 67
                    > via t3-0/3/3.0
10.255.255.179/32  *[OSPF/150] 00:29:56, metric 1, tag 3489661228
                    > via t1-0/0/0.0
10.255.255.180/32  *[Direct/0] 07:53:49
                    > via lo0.0
172.16.0.0/12      *[Static/5] 07:53:49
                    > to 192.255.14.254 via fxp0.0
192.255.0.0/18     *[Static/5] 07:53:49
                    > to 192.255.14.254 via fxp0.0
192.255.14.0/24    *[Direct/0] 07:53:49
                    > via fxp0.0
192.255.14.180/32  *[Local/0] 07:53:49
                      Local via fxp0.0
192.255.40.0/22    *[Static/5] 06:07:28
                    > to 192.255.14.254 via fxp0.0
192.255.64.0/18    *[Static/5] 07:49:39
                    > to 192.255.14.254 via fxp0.0
192.255.197.36/30  *[Direct/0] 03:38:03
                    > via t1-0/0/0.0
                    [OSPF/10] 00:29:56, metric 65
                    > via t1-0/0/0.0
192.255.197.38/32  *[Local/0] 07:52:52
                      Local via t1-0/0/0.0
192.255.198.24/30  *[Direct/0] 03:33:17
                    > via t3-0/3/3.0
                    [OSPF/10] 00:29:56, metric 2
                    > via t3-0/3/3.0
192.255.198.26/32  *[Local/0] 07:52:49
                      Local via t3-0/3/3.0
192.255.198.36/30  *[OSPF/10] 00:29:56, metric 67
                    > via t3-0/3/3.0
207.17.136.192/32  *[Static/5] 07:53:49
                    > to 192.255.14.254 via fxp0.0
224.0.0.5/32       *[OSPF/10] 03:38:55, metric 1
                      MultiRecv
inet.3: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.255.255.175/32  *[LDP/9] 03:35:53, metric 1
                    > via t1-0/0/0.0, Push 100026
```

```
10.255.255.176/32  *[LDP/9] 00:34:13, metric 1
                    > via t3-0/3/3.0
10.255.255.177/32  *[LDP/9] 03:35:53, metric 1
                    > via t1-0/0/0.0, Push 100027
10.255.255.178/32  *[LDP/9] 00:34:13, metric 1
                    > via t3-0/3/3.0, Push 100014
10.255.255.179/32  *[LDP/9] 03:35:53, metric 1
                    > via t1-0/0/0.0, Push 100028
mpls.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
100002             *[LDP/9] 03:35:53, metric 1
                    > via t1-0/0/0.0, Swap 100026
100003             *[LDP/9] 03:35:53, metric 1
                    > via t1-0/0/0.0, Swap 100027
100004             *[LDP/9] 03:35:53, metric 1
                    > via t1-0/0/0.0, Swap 100028
100009             *[LDP/9] 00:34:13, metric 1
                    > via t3-0/3/3.0, Pop
100009(S=0)        *[LDP/9] 00:34:13, metric 1
                    > via t3-0/3/3.0, Pop
100010             *[LDP/9] 00:34:13, metric 1
                    > via t3-0/3/3.0, Swap 100014
```

Router PE4 Status

```
user@PE4> show bgp summary

Groups: 2 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l3vpn.0            6          6          0          0          0          0
inet.0                12         10          0          0          0          0
Peer               AS      InPkt     OutPkt    OutQ   Flaps Last Up/DwnState|#Active/Received/Damped...
192.255.198.22    100        420        429       0       0     3:28:57 Establ
  vpn-customer.inet.0: 5/6/0
10.255.255.177    200        394        406       0       2       28:35 Establ
  bgp.l3vpn.0: 6/6/0
  vpn-customer.inet.0: 5/6/0

user@PE4>  show route protocol bgp
inet.0: 20 destinations, 21 routes (19 active, 0 holddown, 1 hidden)
inet.3: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
vpn-customer.inet.0: 12 destinations, 14 routes (12 active, 0 holddown,
0 hidden)
+ = Active Route, - = Last Active, * = Both
10.49.100.0/24     *[BGP/170] 00:23:27, MED 0, localpref 100, from 10.255.255.177
                      AS path: 100 I
                    > via t1-0/1/1.0, Push 100013, Push 100012(top)
10.49.100.1/32     *[BGP/170] 00:23:27, localpref 100, from 10.255.255.177
                      AS path: 100 I
                    > via t1-0/1/1.0, Push 100013, Push 100012(top)
10.49.101.0/24     *[BGP/170] 00:23:27, MED 0, localpref 100, from 10.255.255.177
                      AS path: 100 I
                    > via t1-0/1/1.0, Push 100013, Push 100012(top)
10.49.200.0/24     *[BGP/170] 03:29:00, MED 0, localpref 100
                      AS path: 100 I
                    > to 192.255.198.22 via t3-0/0/3.0
10.49.200.1/32     *[BGP/170] 03:29:00, localpref 100
                      AS path: 100 I
                    > to 192.255.198.22 via t3-0/0/3.0
```

```
10.49.201.0/24     *[BGP/170] 03:29:00, MED 0, localpref 100
                      AS path: 100 I
                    > to 192.255.198.22 via t3-0/0/3.0
10.255.255.181/32  *[BGP/170] 00:23:27, localpref 100, from 10.255.255.177
                      AS path: 100 I
                    > via t1-0/1/1.0, Push 100013, Push 100012(top)
10.255.255.182/32  *[BGP/170] 03:29:00, localpref 100
                      AS path: 100 I
                    > to 192.255.198.22 via t3-0/0/3.0
192.255.14.0/24    *[BGP/170] 03:29:00, localpref 100
                      AS path: 100 I
                    > to 192.255.198.22 via t3-0/0/3.0
                     [BGP/170] 00:23:27, localpref 100, from 10.255.255.177
                      AS path: 100 I
                    > via t1-0/1/1.0, Push 100013, Push 100012(top)
192.255.198.12/30  *[BGP/170] 00:23:27, localpref 100, from 10.255.255.177
                      AS path: I
                    > via t1-0/1/1.0, Push 100014, Push 100012(top)
192.255.198.20/30   [BGP/170] 03:29:00, localpref 100
                      AS path: 100 I
                    > to 192.255.198.22 via t3-0/0/3.0
mpls.0: 13 destinations, 13 routes (13 active, 0 holddown, 0 hidden)
bgp.l3vpn.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.255.255.177:1:10.49.100.0/24
                   *[BGP/170] 00:28:38, MED 0, localpref 100, from 10.255.255.177
                      AS path: 100 I
                    > via t1-0/1/1.0, Push 100013, Push 100012(top)
10.255.255.177:1:10.49.100.1/32
                   *[BGP/170] 00:28:38, localpref 100, from 10.255.255.177
                      AS path: 100 I
                    > via t1-0/1/1.0, Push 100013, Push 100012(top)
10.255.255.177:1:10.49.101.0/24
                   *[BGP/170] 00:28:38, MED 0, localpref 100, from 10.255.255.177
                      AS path: 100 I
                    > via t1-0/1/1.0, Push 100013, Push 100012(top)
10.255.255.177:1:10.255.255.181/32
                   *[BGP/170] 00:28:38, localpref 100, from 10.255.255.177
                      AS path: 100 I
                    > via t1-0/1/1.0, Push 100013, Push 100012(top)
10.255.255.177:1:192.255.14.0/24
                   *[BGP/170] 00:28:38, localpref 100, from 10.255.255.177
                      AS path: 100 I
                    > via t1-0/1/1.0, Push 100013, Push 100012(top)
10.255.255.177:1:192.255.198.12/30
                   *[BGP/170] 00:28:38, localpref 100, from 10.255.255.177
                      AS path: I
                    > via t1-0/1/1.0, Push 100014, Push 100012(top)

user@PE4>  show route protocol ldp
inet.0: 20 destinations, 21 routes (19 active, 0 holddown, 1 hidden)
inet.3: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.255.255.175/32  *[LDP/9] 00:29:08, metric 1
                    > via t1-0/1/1.0, Push 100011
10.255.255.176/32  *[LDP/9] 00:29:08, metric 1
                    > via t1-0/1/1.0
10.255.255.177/32  *[LDP/9] 00:29:08, metric 1
                    > via t1-0/1/1.0, Push 100012
10.255.255.179/32  *[LDP/9] 00:29:08, metric 1
                    > via t1-0/1/1.0, Push 100013
```

```
10.255.255.180/32  *[LDP/9] 00:29:08, metric 1
                    > via t1-0/1/1.0, Push 100010
vpn-customer.inet.0: 12 destinations, 14 routes (12 active, 0 holddown, 0 hidden)
mpls.0: 13 destinations, 13 routes (13 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
100014             *[LDP/9] 00:29:08, metric 1
                    > via t1-0/1/1.0, Pop
100014(S=0)        *[LDP/9] 00:29:08, metric 1
                    > via t1-0/1/1.0, Pop
100015             *[LDP/9] 00:29:08, metric 1
                    > via t1-0/1/1.0, Swap 100010
100016             *[LDP/9] 00:29:08, metric 1
                    > via t1-0/1/1.0, Swap 100011
100017             *[LDP/9] 00:29:08, metric 1
                    > via t1-0/1/1.0, Swap 100012
100018             *[LDP/9] 00:29:08, metric 1
                    > via t1-0/1/1.0, Swap 100013
bgp.l3vpn.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
```

Router CE4 Status

```
user@CE4> show route protocol bgp
inet.0: 20 destinations, 25 routes (19 active, 0 holddown, 6 hidden)
+ = Active Route, - = Last Active, * = Both
10.49.100.0/24     *[BGP/170] 00:28:00, localpref 100
                      AS path: 200 200 I
                    > to 192.255.198.21 via t3-0/0/3.0
10.49.100.1/32     *[BGP/170] 00:28:00, localpref 100
                      AS path: 200 200 I
                    > to 192.255.198.21 via t3-0/0/3.0
10.49.101.0/24     *[BGP/170] 00:28:00, localpref 100
                      AS path: 200 200 I
                    > to 192.255.198.21 via t3-0/0/3.0
10.255.255.181/32  *[BGP/170] 00:28:00, localpref 100
                      AS path: 200 200 I
                    > to 192.255.198.21 via t3-0/0/3.0
192.255.198.12/30  *[BGP/170] 00:28:00, localpref 100
                      AS path: 200 I
                    > to 192.255.198.21 via t3-0/0/3.0

user@CE4>  show bgp summary
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 0          0          0          0          0          0
Peer               AS      InPkt     OutPkt    OutQ   Flaps Last
Up/DwnState|#Active/Received/Damped...
192.255.198.21    200        426        421       0       0     3:28:20 5/10/0
             0/0/0
```

Related Topics For additional information about multiple-instance LDP and carrier-of-carriers configuration, see the following resources:

- *JUNOS VPNs Configuration Guide*
- *JUNOS MPLS Applications Configuration Guide*
- *JUNOS Policy Framework Configuration Guide*

- RFC 3017, *Carrying Label Information in BGP-4*
- RFC 3036, *LDP Specification*

Revision History

5 October 2009—10.0R1 Release. Roy Spencer.

13 June 2009—9.6R1 Release. Roy Spencer.

13 April 2009—9.5R1 Release. Roy Spencer.

15 January 2009—9.4R1 Release. Roy Spencer.

10 October 2008—9.3R1 Release. Roy Spencer.

10 July 2008—9.2R1 Release. Roy Spencer.

10 April 2008—9.1R1 Release. Roy Spencer.

1 February 2008—9.0R1 Release. Fawn Damitio.

27 March 2007—8.3R1 Release. Fawn Damitio.

12 January 2007—Added support for MX960 Ethernet Services Routers. 8.2R1 Release. Fawn Damitio.

15 September 2006—8.1R1 Release. Richard Hendricks.

29 June 2006—8.0R1 Release. Richard Hendricks.

27 March 2006—7.6R1 Release. Richard Hendricks.

9 January 2006—7.5R1 Release. Richard Hendricks.

14 September 2005—7.4R1 Release. Richard Hendricks.

13 June 2005—7.3R1 Release. Richard Hendricks.

5 April 2005—7.2R1 Release. Richard Hendricks.

2 February 2005—7.1R1 Release. Richard Hendricks.

6 October 2004—7.0R1 Release. Richard Hendricks.

6 July 2004—6.4R1 Release. Richard Hendricks.

5 April 2004—6.3R1 Release. Richard Hendricks.

22 December 2003—6.2R1 Release. Richard Hendricks.

22 September 2003—6.1R1 Release. Richard Hendricks.

30 June 2003—6.0R1 Release. Richard Hendricks.

2 April 2003—5.7R1 Release. Richard Hendricks.

27 December 2002—5.6R1 Release. Richard Hendricks.

30 September 2002—5.5R1 Release. Richard Hendricks.

19 July 2002—5.4R1 Release. Richard Hendricks.

6 May 2002—Initial document written. Richard Hendricks.

Chapter 4

MPLS LSP Link Protection and Node-Link Protection

This chapter covers the following topics:

- MPLS LSP Link Protection and Node-Link Protection Overview on page 101
- Link Protection on page 103
- Node-Link Protection on page 104
- MPLS LSP Link Protection and Node-Link Protection System Requirements on page 105
- MPLS LSP Link Protection and Node-Link Protection Terms and Acronyms on page 105
- Configuring MPLS LSP Link Protection or Node-Link Protection on page 106
- Configuring Link Protection or Node-Link Protection on the LSP on page 106
- Configuring Link Protection on the RSVP Interfaces Traversed by the LSP on page 106
- Option: Configuring Multiple Bypass LSPs, Manual Bypass LSPs, and Link Protection Priority on page 107
- Option: Adding Class of Service to a Link-Protected LSP or a Bypass LSP on page 108
- Verifying MPLS LSP Link Protection and Node Link Protection on page 109
- MPLS LSP Link Protection or Node-Link Protection Configuration Examples on page 109
- Example: Configuring MPLS LSP Link Protection on page 110
- Example: Node-Link Protection Configuration on page 129
- For More Information on page 139
- Revision History on page 139

MPLS LSP Link Protection and Node-Link Protection Overview

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 Software LSP protection method.

While fast reroute can only protect LSPs on a one-to-one basis, link protection and node-link protection can protect multiple LSPs by using only a single, logical bypass LSP. Link protection can provide robust backup support for a link, node-link protection can bypass 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.

Prior to JUNOS Release 5.4, the two mechanisms used to enable rapid MPLS LSP reroutes in Juniper Networks routers were Packet Forwarding Engine local repair and fast reroute. Packet Forwarding Engine local repair is an infrastructure-based solution and fast reroute provides a single backup LSP for every protected primary LSP. However, configuring backup LSPs on a one-to-one basis can become a scaling challenge for a growing MPLS network.

Scalable solutions for LSP redundancy include link protection and node-link protection. Both approaches are explained in RFC 4090, *Fast Reroute Extensions to RSVP-TE for LSP Tunnels*. In general, these are facility-based methods that quickly reroute traffic from multiple LSPs. They also reduce the amount of configuration necessary to implement LSP protection.

You can configure either link protection or node-link protection by itself, fast reroute by itself, or both fast reroute and one of the protection methods. Whenever one or more of these reroute options are enabled, Packet Forwarding Engine local repair is activated by default.

To enable any of Juniper Networks MPLS LSP reroute options, you must first install the LSP as a valid next hop in the main inet.0 routing table on the ingress PE router. You can accomplish this in one of several of ways:

- Enable the BGP learned routes to use the LSP.
- Set the bgp-igp or bgp-igp-both-ribs parameters at the [edit protocols mpls traffic engineering] hierarchy level.
- Configure install *prefix* active at the [edit protocols mpls lsp *lsp-name*] hierarchy level.
- Configure a static route with an indirect next hop that goes to the LSP end.
- Configure a static route with an LSP next hop.
- Configure IS-IS support for bidirectional LSPs.

To summarize, the MPLS LSP reroute options available in JUNOS Software are as follows:

- Packet Forwarding Engine local repair—This data plane method adds enhanced capabilities to the Packet Forwarding Engine subsystem and reduces the time needed for path switchover. With local repair, the Packet Forwarding Engine can correct a path failure before it receives recomputed paths from the Routing Engine. The Routing Engine pre-computes backup routes for every MPLS path and provides this information to the Packet Forwarding Engine before any failure. Packet Forwarding Engine local repair is enabled by default and requires no additional configuration.
- Fast reroute—The original control plane method for fast reroute of individual LSPs is described as "one-to-one" protection in the IETF Internet draft *Fast Reroute Extensions to RSVP-TE for LSP Tunnels*. JUNOS Software calculates LSP detours for LSPs and implements the rerouted paths as needed. You can configure the command fast-reroute at the [edit protocols mpls *lsp-name*] hierarchy level. For more information about MPLS LSP fast reroute, see the *JUNOS MPLS Applications Configuration Guide*.
- Link protection—Another control plane method discussed in this guide. In general, link protection is useful when you wish to protect LSPs after a supporting link is lost.
- Node-link protection—This is also a control plane method and is discussed in this guide. In general, link protection is useful when you wish to protect LSPs after a supporting node fails.

Link Protection

Link protection offers per-link traffic protection. It supports fast rerouting of user traffic over one mission-critical link. It does this on a per-LSP basis, much like the fast reroute option. However, it can also aggregate several protected LSPs over a single bypass LSP.

This flexible approach to single-link, rapid reroute does not require any new protocol modification beyond the RSVP-TE specification. Bypass LSPs efficiently aggregate traffic from multiple LSPs when the reroute occurs.

When link protection is enabled on a router interface and a protected LSP traverses this protected interface, JUNOS Software creates a trunk-like, bypass LSP to provide an alternate path to the RSVP neighbor. Each bypass LSP keeps track of all protected LSPs that are associated with the neighbor. In case of a neighbor failure, the protected LSPs are rerouted over the bypass LSP. Bypass LSPs use label stacking to protect user traffic.

At the interface level, the router keeps track of bypass LSP characteristics. Whenever an interface enables or disables link protection, the changes are saved at the interface level and then propagated to the RSVP neighbor. When a neighbor requires link protection, the router checks the associated interface structure to determine how to create a bypass LSP.

On a per-RSVP neighbor basis, the router keeps track of all the LSP sessions passing through a neighbor as well as the bypass LSP status. For the bypass LSP, the router

maintains information about protected neighbors. For regular LSPs, the router monitors all threads containing the LSP. When a regular LSP is lost, the bypass LSP reroutes user traffic by using information about the next hop, egress Explicit Route Object (ERO), interface, and peer address.

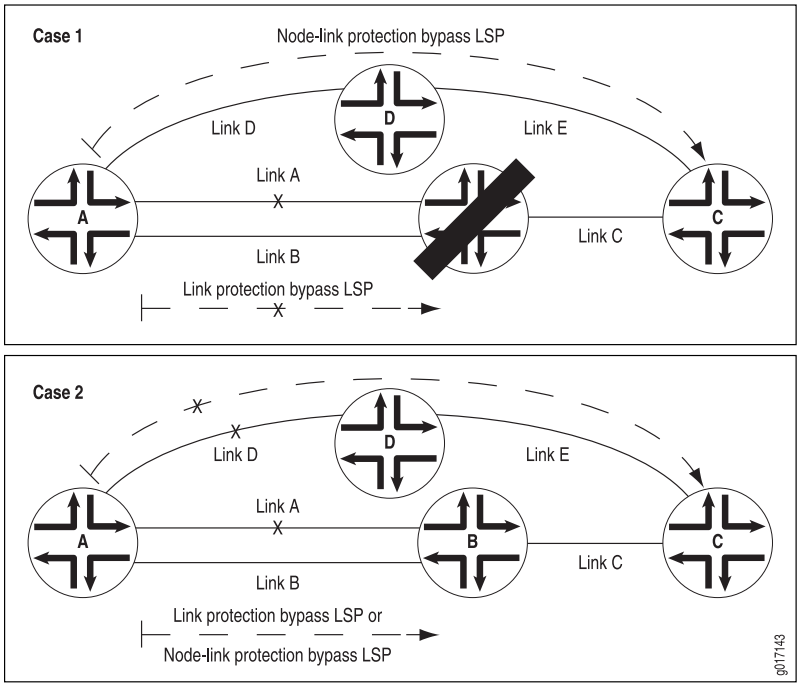
NOTE: Fast reroute, link protection, and node-link protection all rely on Constrained Shortest Path First (CSPF) to select bypass LSPs. The CSPF computation attempts to find an LSP that bypasses an affected node first, but can select an alternate link through the affected node if a node bypass LSP is not available.

Node-Link Protection

While link protection is useful for selecting an alternate path to the same router when a link fails, node-link protection establishes a bypass LSP through a different router altogether. For Case 1 in Figure 9 on page 104, link protection allows an LSP to switch to link B and immediately bypass failed link A. However, if Router B fails, link B will fail and the link-protected LSP will be lost.

With node-link protection, the backup LSP can switch to link D instead and bypass the failed links and router. Another benefit of node-link protection shown in Case 2 is that a node-link-protected LSP can act like a link-protected LSP and switch to link B if link D is unavailable.

Figure 9: Link Protection and Node-Link Protection Comparison



JUNOS Software signals bypass LSPs dynamically when a protected LSP transverses the protected link. The software determines if the protected LSP needs a node bypass or a link bypass and prepares the necessary bypass LSP automatically. The bypass LSP is torn down automatically when a protected LSP does not use the link.

Because the creation and removal of bypass LSPs is automatic, network resources can be used for other purposes when the bypass LSP is not needed. Likewise, network administrators do not need to configure bypass LSPs statically and can focus their maintenance efforts elsewhere.

MPLS LSP Link Protection and Node-Link Protection System Requirements

To implement MPLS LSP link protection or node-link protection, your system must meet these minimum requirements:

- JUNOS Release 8.2 or later for support on MX Series routers
- JUNOS Release 7.4 or later for enhanced operational commands and system log messages for link protection and node-link protection
- JUNOS Release 7.3 or later for link protection of point-to-multipoint LSPs and for class of service on link-protected LSPs and bypass LSPs
- JUNOS Release 7.1 or later for multiple bypass LSPs, manual bypass LSPs, and link protection priority
- JUNOS Release 5.4 or later for link protection
- JUNOS Release 6.0 and later for node-link protection
- Three Juniper Networks M Series, MX Series, or T Series routers

MPLS LSP Link Protection and Node-Link Protection Terms and Acronyms

B

backup LSP A redundant LSP used to reroute a single, primary LSP. Backup LSPs are found in link protection, node-link protection, and fast reroute redundancy methods.

bypass LSP A logical trunk used to reroute multiple backup LSPs over a single connection protected with link protection.

L

link protection A method of establishing bypass LSPs to provide rapid reroute capability for primary LSPs on a per-link basis. For more information about link protection, see the *JUNOS MPLS Applications Configuration Guide*.

N

node-link protection A method of establishing bypass LSPs to provide rapid reroute capability for primary LSPs on a per-node basis. If node protection is unavailable, the LSP attempts to use link protection. For more information about node-link protection, see the *JUNOS MPLS Applications Configuration Guide*.

Configuring MPLS LSP Link Protection or Node-Link Protection

To implement MPLS LSP link protection or node-link protection, perform the following:

- Configuring Link Protection or Node-Link Protection on the LSP on page 106
- Configuring Link Protection on the RSVP Interfaces Traversed by the LSP on page 106
- Option: Configuring Multiple Bypass LSPs, Manual Bypass LSPs, and Link Protection Priority on page 107
- Option: Adding Class of Service to a Link-Protected LSP or a Bypass LSP on page 108
- Verifying MPLS LSP Link Protection and Node Link Protection on page 109

Configuring Link Protection or Node-Link Protection on the LSP

You enable the level of LSP protection you want on the ingress router. Link protection redirects LSP traffic to a bypass LSP that can traverse the same router that contains the affected link, whereas node-link protection sends LSP traffic to a bypass LSP that circumvents the affected router. To enable link protection for an LSP or point-to-multipoint LSP, include the link-protection statement at the [edit protocols mpls label-switched-path *lsp-name*] hierarchy level. To enable node-link protection for an LSP, include the node-link-protection statement at the [edit protocols mpls label-switched-path *lsp-name*] hierarchy level.

```
[edit]
protocols {
    mpls {
        label-switched-path lsp-name {
            ( link-protection | node-link-protection );
        }
    }
}
```

After link protection or node-link protection is established, the LSP marks the desired link protection bit in the RSVP Session Attribute (SA) object. To disable link protection or node-link protection for an LSP, delete the link-protection or node-link-protection statements at the [edit protocols mpls label-switched-path *lsp-name*] hierarchy level. For more information about point-to-multipoint LSPs, see the *JUNOS MPLS Applications Configuration Guide*.

Configuring Link Protection on the RSVP Interfaces Traversed by the LSP

To complete your link protection or node-link protection configuration, configure RSVP interface-level link protection. Include the link-protection statement at the [edit

protocols rsvp interface *interface-name*] hierarchy level. You must configure the link-protection statement on every RSVP interface used to exit each router in the LSP or point-to-multipoint LSP path. As an option, you can configure a loose or strict path for all bypass LSPs with the path statement at the [edit protocols rsvp interface *interface-name* link-protection] hierarchy level.

```
[edit]
protocols {
   rsvp {
      interface interface-name {
         link-protection {
            path  ip-address  {
               (loose | strict);
            }
         }
      }
   }
}
```

To disable link protection on an RSVP interface, include the disable statement at the [edit protocols rsvp interface *interface-name* link-protection] hierarchy level.

Option: Configuring Multiple Bypass LSPs, Manual Bypass LSPs, and Link Protection Priority

You can configure multiple bypass LSP paths for a link-protected RSVP LSP. When you enable this option, RSVP signals multiple bypasses concurrently for a link-protected LSP. To configure, start by enabling link protection or node-link protection by including the link-protection or node-link-protection statement at the [edit protocols mpls label-switched-path *lsp-name*] hierarchy level. To configure an RSVP interface to include multiple bypasses, specify how much bandwidth the bypasses should consume by including the bandwidth statement at the [edit protocols rsvp interface *interface-name* link-protection] hierarchy level. To limit the total number of bypasses that can be created, include the max-bypasses statement at the [edit protocols rsvp interface *interface-name* link-protection] hierarchy level.

Other bypass options include limiting the maximum number of hops a bypass LSP will traverse, selectively disallowing node-link protection, setting a timer to redistribute data periodically across the bypass LSPs, requiring strict or loose paths for the bypass LSPs, and establishing the percentage of bandwidth required for the bypass LSPs. To configure, include the hop-limit, no-node-protection, optimize-timer, path, and subscription statements, respectively, at the [edit protocols rsvp interface *interface-name* link-protection] hierarchy level.

Another option is to specify bypass LSPs manually. To configure, include the to and bandwidth statements at the [edit protocols rsvp interface *interface-name* link-protection bypass *bypass-name*] hierarchy level. Optionally, you can specify the hop limit and path type for the manual bypass LSP. To configure the options, include the hop-limit and path statements at the [edit protocols rsvp interface *interface-name* link-protection bypass *bypass-name*] hierarchy level.

Link protection priority enables you to provide preferred combinations of priority and class in a traffic engineering class matrix. When selecting a bypass LSP, the

routing platform selects the bypass containing the lowest priority. To configure link protection priority for all bypass LSP paths, include the **priority** statement at the [**edit protocols rsvp interface** *interface-name* **link-protection**] hierarchy level. To configure link protection priority for a manually specified bypass LSP path, include the **priority** statement at the [**edit protocols rsvp interface** *interface-name* **link-protection bypass** *bypass-name*] hierarchy level.

```
[edit protocols]
mpls {
    label-switched-path lsp-name {
        (link-protection | node-link-protection);
    }
}
rsvp {
    interface interface-name {
        link-protection {
            bandwidth bps;
            bypass bypass-name {
                to ip-address;
                bandwidth bps;
                hop-limit maximum-hops; # The default value is 255 hops.
                path ip-address {
                    (loose | strict);
                }
                priority priority;
            }
            hop-limit maximum-hops; # The range is 2 hops to the default (255 hops).
            max-bypasses number; # The range for this statement is 1 to 99.
            no-node-protection;
            optimize-timer seconds; # The default value of 0 disables this option.
            path ip-address {
                (loose | strict);
            }
            subscription percent; # The range for this statement is 1 to 65535.
        }
    }
}
```

For more information on multiple bypass LSPs, manually configured bypass LSPs, and link protection priority, see the *JUNOS MPLS Applications Configuration Guide*.

Option: Adding Class of Service to a Link-Protected LSP or a Bypass LSP

For link-protected LSPs and bypass LSPs, you can specify a class-of-service designation to provide different levels of traffic quality. To configure class of service for link-protected LSPs, include the **class-of-service** statement at the [**edit protocols rsvp interface** *interface-name* **link-protection**] hierarchy level. To configure class of service for bypass LSPs, include the **class-of-service** statement at the [**edit protocols rsvp interface** *interface-name* **link-protection bypass** *ip-address*] hierarchy level.

For more information about class of service for link-protected LSPs and bypass LSPs, see the *JUNOS MPLS Applications Configuration Guide* and the *JUNOS Class of Service Configuration Guide*.

Verifying MPLS LSP Link Protection and Node Link Protection

Purpose In JUNOS Release 7.4 and later, you can issue enhanced operational mode commands and receive system log messages that provide more details about the operation of your link-protected or node-link-protected LSPs. The following guidelines explain the type of information available from the output of each command or message.

Action

- show rsvp session extensive—Indicates if an LSP is protected, displays which backup and bypass LSPs provide the protection, and records the history of protection-related events (such as bypass LSP creation and detailed LSP failure information).
- show rsvp interface extensive—Indicates protection status for an interface, displays the number of LSPs protected by the RSVP interface, and records the history of bypass LSPs that protect the interface.
- show rsvp session bypass extensive—Displays active RSVP reservations for bypass LSPs.
- RPD_RSVP_BACKUP_DOWN—This system log message records when a backup LSP is not created, when a backup LSP stops operating, and when a bypass LSP carrying backup LSPs goes down.

Related Topics

- For more information about the enhanced link protection and node-link protection operational commands, see the *JUNOS Routing Protocols and Policies Command Reference*.
- For more information about the system log message, see the *JUNOS System Log Messages Reference*.

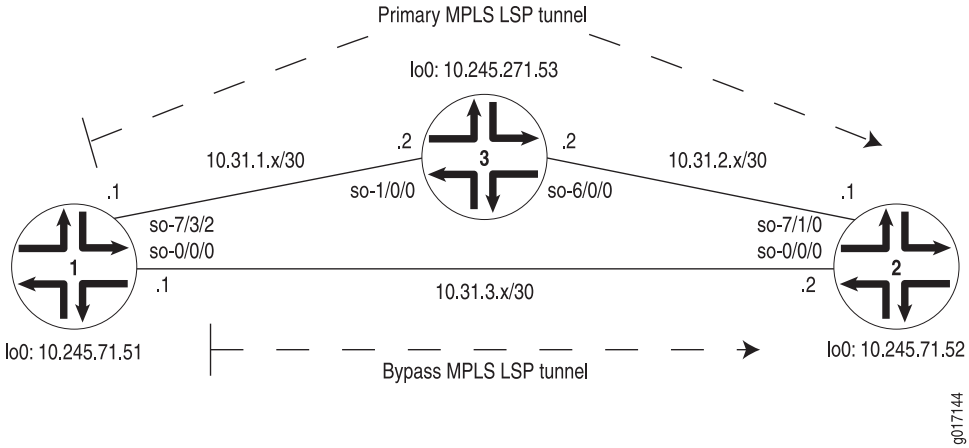
MPLS LSP Link Protection or Node-Link Protection Configuration Examples

This section contains configuration examples and commands you can issue to verify MPLS LSP link protection and node-link protection:

- Example: Configuring MPLS LSP Link Protection on page 110
- Example: Node-Link Protection Configuration on page 129

Example: Configuring MPLS LSP Link Protection

Figure 10: MPLS LSP Link Protection Topology Diagram



In Figure 10 on page 110, a primary MPLS LSP is established from Router 1 through Router 3 to destination Router 2. To implement link protection, include the link-protection statement on the primary LSP at the ingress point and on the appropriate downstream RSVP interfaces you wish to protect. In this case, the primary LSP named Protected_LSP on Router 1 requires link protection, as does the so-7/3/2 RSVP interface of Router 1 and the so-6/0/0 RSVP interface of Router 3. After link protection is enabled for the protected LSP, bypass LSPs are established automatically for the LSP-traversed interfaces of Routers 1 and 3.

On Router 1, configure an interior gateway protocol (IGP) routing protocol (in this case, IS-IS), RSVP, and MPLS on the so-0/0/0 and so-7/3/2 interfaces. Next, configure the primary LSP on Router 1 to point to the loopback address of Router 2. The primary LSP's strict path must travel though Router 3.

Enable link protection on both the LSP itself and the outgoing RSVP interface traversed by the primary LSP (in this case, the so-7/3/2 RSVP interface of Router 1). After you enable link protection, the router notices that the primary LSP is protected and prepares a bypass LSP.

Configure a static route of 10.31.5.1 in the LSP on Router 1. You can use this route for testing purposes. Also, if you want to enable Packet Forwarding Engine local repair, establish a policy that requires all traffic to use per-packet load balancing. Once this policy is configured, export it to the neighboring routers with the export statement at the [edit routing-options forwarding-table] hierarchy level.

Router 1

```
[edit]
interfaces {
   so-0/0/0 {
      unit 0 {
         family inet {
            address 10.31.3.1/30;
         }
```

```
        family iso;
        family mpls;
    }
}
so-7/3/2 {
    unit 0 {
        family inet {
            address 10.31.1.1/30;
        }
        family iso;
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.245.71.51/32;
        }
        family iso {
        }
    }
}
protocols {
    rsvp {
        interface so-7/3/2.0 {
            link-protection; # Enable link protection on the interface carrying the main
                LSP.
        }
        interface so-0/0/0.0 {
        }
        mpls {
            label-switched-path Protected_LSP {
                to 10.245.71.52;
                install 10.31.5.1/32 active; # This route is used for testing the LSPs.
                link-protection; # Enable link protection on the protected LSP.
                primary path1;
            }
            path path1 {
                10.31.1.2 strict;
            }
            interface so-0/0/0.0;
            interface so-7/3/2.0;
        }
        isis {
            level 2 wide-metrics-only;
            interface so-0/0/0.0 {
                level 1 disable;
                level 2 {
                    metric 100;
                    te-metric 100;
                }
            }
            interface so-7/3/2.0 {
                level 1 disable;
                level 2 {
                    metric 10;
```

```
                te-metric 10;
            }
        }
        interface lo0.0 {
            passive;
        }
    }
}
routing-options {
    forwarding-table { # Apply this policy to the forwarding table only
        export pplb; # if Packet Forwarding Engine local repair is needed.
    }
}
policy-options {
    policy-statement pplb {
        then {# Configure this policy only if
            load-balance per-packet; # Packet Forwarding Engine local repair is needed.
        }
    }
}
}
```

On Router 2, no link protection configuration is needed. However, you should configure MPLS, RSVP, and IS-IS to communicate with the other routers.

Router 2

```
[edit]
interfaces {
    so-0/0/0 {
        unit 0 {
            family inet {
                address 10.31.3.2/30;
            }
            family iso;
            family mpls;
        }
    }
    so-7/1/0 {
        unit 0 {
            family inet {
                address 10.31.2.1/30;
            }
            family iso;
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.245.71.53/32;
            }
            family iso {
            }
        }
    }
```

```
    protocols {
      rsvp {
        interface so-0/0/0.0;
        interface so-7/1/0.0;
      }
      mpls {
        interface so-0/0/0.0;
        interface so-7/1/0.0;
      }
      isis {
        level 2 wide-metrics-only;
        interface so-0/0/0.0 {
          level 1 disable;
          level 2 {
            metric 100;
            te-metric 100;
          }
        }
        interface so-7/1/0.0 {
          level 1 disable;
          level 2 {
            metric 10;
            te-metric 10;
          }
        }
        interface lo0.0 {
          passive;
        }
      }
    }
  }
```

On Router 3, include IS-IS, RSVP, and MPLS on the so-1/0/0 and so-6/0/0 interfaces. Enable link protection on the remaining RSVP interface traversed by the primary LSP (in this case, the so-6/0/0 RSVP interface). After you enable link protection, the router notices the primary LSP is protected and prepares a bypass LSP.

To enable Packet Forwarding Engine local repair, establish a policy that requires traffic to use per-packet load balancing. Once this policy is configured, export it to the neighboring routers.

Router 3

```
[edit]
interfaces {
  so-1/0/0 {
    unit 0 {
      family inet {
        address 10.31.1.2/30;
      }
      family iso;
      family mpls;
    }
  }
  so-6/0/0 {
    unit 0 {
      family inet {
```

```
            address 10.31.2.2/30;
        }
        family iso;
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.245.271.53/32;
        }
        family iso {
        }
    }
}
protocols {
    rsvp {
        interface so-1/0/0.0;
        interface so-6/0/0.0 { # Primary interface going to Router 2.
            link-protection;
        }
    }
    mpls {
        interface so-1/0/0.0;
        interface so-6/0/0.0;
    }
    isis {
        level 2 wide-metrics-only;
        interface so-1/0/0.0 {
            level 1 disable;
            level 2 {
                metric 10;
                te-metric 10;
            }
        }
        interface so-6/0/0.0 {
            level 1 disable;
            level 2 {
                metric 10;
                te-metric 10;
            }
        }
        interface lo0.0 {
            passive;
        }
    }
}
policy-options {
    policy-statement pplb {
        then {
            load-balance per-packet; # If Packet Forwarding Engine local repair is needed.
        }
    }
}
routing-options {
    forwarding-table {
```

```
            export pplb; # If Packet Forwarding Engine local repair is needed.
        }
    }
}
```

Verifying Your Work

To verify proper operation of MPLS LSP link protection, use the following commands:

- show mpls lsp
- show route
- show route forwarding-table
- show rsvp interface detail
- show rsvp neighbor detail
- show rsvp session detail

The following sections show the output of these commands used with the configuration example:

- Case 1: Normal Operation on page 115
- Case 2: When the Link from Router 1 to Router 3 Is Disabled on page 122
- Case 3: When the Link from Router 3 to Router 2 Is Disabled on page 125

Case 1: Normal Operation

Once link protection is enabled on the required RSVP interfaces and primary LSP, the bypass LSPs are prepared.

Router 1

```
user@Router1> show mpls lsp
Ingress LSP: 1 sessions
To              From            State Rt ActivePath       P     LSPname
10.245.71.52    10.245.71.51    Up     1 path1            *     Protected_LSP
```

This is the main LSP.

```
Total 1 displayed, Up 1, Down 0
Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
Transit LSP: 1 sessions
To              From            State Rt Style Labelin Labelout LSPname
10.245.71.52    10.245.271.53   Up     1  1 SE  100003        0
Bypass->10.31.2.1
```

This is the bypass LSP from Router 3 to Router 2.

```
Total 1 displayed, Up 1, Down 0

user@Router1>  show rsvp session detail
Ingress RSVP: 2 sessions
```

```
10.245.71.52
   From: 10.245.71.51, LSPstate: Up, ActiveRoute: 1
  LSPname: Protected_LSP
```

This is the main LSP. Notice that a backup LSP is not signaled when the main LSP is still up.

```
  Resv style: 1 SE, Label in: -, Label out: 100007
  Time left:    -,  Since: Thu Aug  8 12:13:24 2002
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 2 receiver 33 protocol 0
  Link protection desired
  Type: Link protected LSP
  PATH rcvfrom: localclient
  PATH sentto: 10.31.1.2 (so-7/3/2.0) 36 pkts
  RESV rcvfrom: 10.31.1.2  (so-7/3/2.0) 38 pkts
  Explct route: 10.31.1.2 10.31.2.1
  Record route: <self>  10.31.1.2  10.31.2.1
10.245.271.53
   From: 10.245.71.51, LSPstate: Up, ActiveRoute: 0
  LSPname: Bypass->10.31.1.2
```

This is the bypass from Router 1 to Router 2. This also appears in the show mpls lsp command output above.

```
  Resv style: 1 SE, Label in: -, Label out: 100000
  Time left:    -,  Since: Thu Aug  8 12:14:31 2002
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 51 protocol 0
  Type: Bypass LSP
  PATH rcvfrom: localclient
  PATH sentto: 10.31.3.2 (so-0/0/0.0) 32 pkts
  RESV rcvfrom: 10.31.3.2  (so-0/0/0.0) 32 pkts
  Explct route: 10.31.3.2 10.31.2.2
  Record route: <self>  10.31.3.2  10.31.2.2
Total 2 displayed, Up 2, Down 0
Egress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0
Transit RSVP: 1 sessions
10.245.71.52
  From: 10.245.271.53, LSPstate: Up, ActiveRoute: 1
  LSPname: Bypass->10.31.2.1
  Resv style: 1 SE, Label in: 100003, Label out: 0
  Time left:   52,  Since: Thu Aug  8 12:03:27 2002
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 19 protocol 0
  PATH rcvfrom: 10.31.1.2  (so-7/3/2.0) 76 pkts
  PATH sentto: 10.31.3.2 (so-0/0/0.0) 77 pkts
  RESV rcvfrom: 10.31.3.2  (so-0/0/0.0) 78 pkts
  Explct route: 10.31.3.2
  Record route: 10.31.1.2  <self>  10.31.3.2
Total 1 displayed, Up 1, Down 0

user@Router1>  show rsvp interface detail
RSVP interface: 2 active
fxp0.0  Index 1, State Dis/Up
  NoAuthentication, NoAggregate, NoReliable, NoLinkProtection
  HelloInterval 9(second)
  Address 192.168.71.52
```

```
  PacketType              Total                 Last 5 seconds
                     Sent      Received        Sent      Received
  Path                  0             0           0             0
  PathErr               0             0           0             0
  PathTear              0             0           0             0
  Resv                  0             0           0             0
  ResvErr               0             0           0             0
  ResvTear              0             0           0             0
  Hello                 0             0           0             0
  Ack                   0             0           0             0
  Srefresh              0             0           0             0
  EndtoEnd RSVP         0             0           0             0
so-0/0/0.0  Index 8, State Ena/Up
  NoAuthentication, NoAggregate, NoReliable, NoLinkProtection
  HelloInterval 20(second)
  Address 10.31.3.1, 10.245.71.51
  ActiveResv 2, PreemptionCnt 0, Update threshold 10%
  Subscription 100%, StaticBW 622.08Mbps, AvailableBW 622.08Mbps
  PacketType              Total                 Last 5 seconds
                     Sent      Received        Sent      Received
  Path                441             0           0             0
  PathErr               0             0           0             0
  PathTear              3             0           0             0
  Resv                  0           431           0             0
  ResvErr               0             0           0             0
  ResvTear              0             0           0             0
  Hello               489           487           0             0
  Ack                   0             0           0             0
  Srefresh              0             0           0             0
  EndtoEnd RSVP         0             0           0             0
so-7/3/2.0  Index 11, State Ena/Up
  NoAuthentication, NoAggregate, NoReliable,  LinkProtection
```

Link protection is enabled.

```
  HelloInterval 3(second)
  Address 10.31.1.1, 10.245.71.51
  ActiveResv 1, PreemptionCnt 0, Update threshold 10%
  Subscription 100%, StaticBW 2.48832Gbps, AvailableBW 2.48832Gbps
  PacketType              Total                 Last 5 seconds
                     Sent      Received        Sent      Received
  Path                225           138           0             0
  PathErr              12             4           0             0
  PathTear              5             3           0             0
  Resv                134           216           0             1
  ResvErr               0             0           0             0
  ResvTear              3             1           0             0
  Hello               750           799           1             1
  Ack                   0             0           0             0
  Srefresh              0             0           0             0
  EndtoEnd RSVP         0             0           0             0

user@Router1>  show rsvp neighbor detail
RSVP neighbor: 2 learned
Address: 10.31.1.2   via: so-7/3/2.0    status: Up
  Last changed time: 38:17, Idle: 5 sec, Up cnt: 1, Down cnt: 0
  Message received: 329
  Hello: sent 747, received: 747, interval: 3 sec
  Remote instance: 0x41b21a47, Local instance: 0x238fa919
  Refresh reduction:  not operational
```

```
  Link protection:   enabled            #   This should be enabled
    LSP name: Bypass->10.31.1.2
    Bypass LSP: operational,   Backup routes: 2 ,  Backup LSPs: 0
```

The number of backup routes equals 2 because the main LSP is already considered for protection.

```
    Bypass explicit route: 10.31.3.2 10.31.2.2

Address: 10.31.3.2   via: so-0/0/0.0    status: Up
  Last changed time: 17:46, Idle: 5 sec, Up cnt: 4, Down cnt: 3
  Message received: 430
  Hello: sent 506, received: 486, interval: 20 sec
  Remote instance: 0x194fa7af, Local instance: 0x507b7c2a
  Refresh reduction:  not operational
  Link protection:  disabled
    Bypass LSP: does not exist,  Backup routes: 0,  Backup LSPs: 0

user@Router1>show route 10.31.5.1 extensive

inet.0: 24 destinations, 24 routes (22 active, 0 holddown, 2 hidden)
10.31.5.1/32 (1 entry, 1 announced)
        State: <FlashAll>
TSI:
KRT in-kernel 10.31.5.1/32 -> {0.0.0.0, 0.0.0.0}
        *RSVP   Preference: 7
                 Next hop: via so-7/3/2.0 weight 1, selected
```

This is the main LSP.

```
                Label-switched-path Protected_LSP
                Label operation: Push 100007
                 Next hop: via so-0/0/0.0 weight 20001
```

This is a backup route, though the backup LSP has not been signaled yet.

```
                Label-switched-path Bypass->10.31.1.2
                Label operation: Push 100007, Push 100000(top)[0]
                State: <Active Int>
                Local AS:    69
                Age: 8:34       Metric: 20
                Task: RSVP
                Announcement bits (1): 0-KRT
                AS path: I

user@Router1>  show route forwarding-table destination 10.31.5.1 extensive

Routing table: inet [Index 0]
Internet:
    Destination:  10.31.5.1/32
  Route type: user               Route reference: 0
  Flags: sent to PFE
  Next-hop type: unilist              Index: 39       Reference: 1
   Next-hop type: Push  100007
   Next-hop interface: so-7/3/2.0       Weight: 1
```

Packet Forwarding Engine local repair is enabled (otherwise, only one entry appears for Next-hop).

```
  Next-hop type: Push 100007, Push 100000(top)[0]
  Next-hop interface: so-0/0/0.0        Weight: 20001
```

The Weight value for the backup starts at 20000 .

Router 3

```
user@Router3> show mpls lsp
Ingress LSP: 0 sessions
```

The ingress bypass LSP to Router 2 does not appear here.

```
Total 0 displayed, Up 0, Down 0
Egress LSP: 1 sessions
To              From            State Rt Style Labelin Labelout LSPname
10.245.271.53   10.245.71.51    Up     0  1 SE       3        -
Bypass->10.31.1.2
Total 1 displayed, Up 1, Down 0
Transit LSP: 1 sessions
To              From            State Rt Style Labelin Labelout LSPname
10.245.71.52    10.245.71.51    Up     1  1 SE  100000        0 Protected_LSP
Total 1 displayed, Up 1, Down 0

user@Router3>  show rsvp session detail
Ingress RSVP: 1 sessions

10.245.71.52
  From: 10.245.271.53, LSPstate: Up, ActiveRoute: 0
   LSPname: Bypass->10.31.2.1
```

This is the ingress bypass session to Router 2 from Router 1.

```
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 100004
  Resv style: 1 SE, Label in: -, Label out: 100004
  Time left:    -,  Since: Thu Aug  8 12:27:07 2002
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 16 protocol 0
  Type: Bypass LSP
  PATH rcvfrom: localclient
  PATH sentto: 10.31.1.1 (so-1/0/0.0) 3 pkts
  RESV rcvfrom: 10.31.1.1  (so-1/0/0.0) 3 pkts
  Explct route: 10.31.1.1 10.31.3.2
  Record route: <self>  10.31.1.1  10.31.3.2
Total 1 displayed, Up 1, Down 0

Egress RSVP: 1 sessions

10.245.271.53
  From: 10.245.71.51, LSPstate: Up, ActiveRoute: 0
  LSPname: Bypass->10.31.1.2
```

This is the bypass from Router 1 to Router 3, arriving by way of Router 2.

```
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: -
  Resv style: 1 SE, Label in: 3, Label out: -
  Time left:   54,  Since: Thu Aug  8 12:26:48 2002
```

```
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 52 protocol 0
  PATH rcvfrom: 10.31.2.1  (so-6/0/0.0) 5 pkts
  PATH sentto: localclient
  RESV rcvfrom: localclient
  Record route: 10.31.3.1  10.31.2.1  <self>
Total 1 displayed, Up 1, Down 0
Transit RSVP: 1 sessions
10.245.71.52
  From: 10.245.71.51, LSPstate: Up, ActiveRoute: 1
  LSPname: Protected_LSP
```

This is the main LSP.

```
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 0
  Resv style: 1 SE, Label in: 100000, Label out: 0
  Time left:   41,  Since: Thu Aug  8 12:26:39 2002
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 2 receiver 33 protocol 0
  Link protection desired
  Type: Link protected LSP
  PATH rcvfrom: 10.31.1.1  (so-1/0/0.0) 9 pkts
  PATH sentto: 10.31.2.1 (so-6/0/0.0) 11 pkts
  RESV rcvfrom: 10.31.2.1  (so-6/0/0.0) 10 pkts
  Explct route: 10.31.2.1
  Record route: 10.31.1.1  <self>  10.31.2.1
Total 1 displayed, Up 1, Down 0

user@Router3>  show rsvp neighbor detail
RSVP neighbor: 2 learned
Address: 10.31.2.1   via: so-6/0/0.0    status: Up
  Last changed time: 27, Idle: 0 sec, Up cnt: 1, Down cnt: 0
  Message received: 19
  Hello: sent 6, received: 6, interval: 9 sec
  Remote instance: 0x625d2852, Local instance: 0x327317df
  Refresh reduction:  not operational
  Link protection:  enabled
    LSP name: Bypass->10.31.2.1
    Bypass LSP: operational,   Backup routes: 1 ,  Backup LSPs: 0
```

The number of backup routes is 1.

```
    Bypass explicit route: 10.31.1.1 10.31.3.2
Address: 10.31.1.1   via: so-1/0/0.0    status: Up
  Last changed time: 41, Idle: 0 sec, Up cnt: 1, Down cnt: 0
  Message received: 15
  Hello: sent 17, received: 17, interval: 3 sec
  Remote instance: 0x2ebdcf43, Local instance: 0x643d9e23
  Refresh reduction:  not operational
  Link protection:  disabled
    Bypass LSP: does not exist,  Backup routes: 0,  Backup LSPs: 0

user@Router3>  show rsvp interface detail
RSVP interface: 2 active
fxp0.0  Index 1, State Dis/Up
  NoAuthentication, NoAggregate, NoReliable, NoLinkProtection
  HelloInterval 9(second)
  Address 192.168.6.64
```

```
  PacketType              Total                 Last 5 seconds
                    Sent      Received        Sent      Received
  Path                 0             0           0             0
  PathErr              0             0           0             0
  PathTear             0             0           0             0
  Resv                 0             0           0             0
  ResvErr              0             0           0             0
  ResvTear             0             0           0             0
  Hello                0             0           0             0
  Ack                  0             0           0             0
  Srefresh             0             0           0             0
  EndtoEnd RSVP        0             0           0             0
so-1/0/0.0  Index 6, State Ena/Up
  NoAuthentication, NoAggregate, NoReliable, NoLinkProtection
  HelloInterval 3(second)
  Address 10.31.1.2, 10.245.271.53
  ActiveResv 1, PreemptionCnt 0, Update threshold 10%
  Subscription 100%, StaticBW 2.48832Gbps, AvailableBW 2.48832Gbps
  PacketType              Total                 Last 5 seconds
                    Sent      Received        Sent      Received
  Path                 5            14           0             1
  PathErr              0             0           0             0
  PathTear             0             0           0             0
  Resv                 8             9           0             0
  ResvErr              0             0           0             0
  ResvTear             0             0           0             0
  Hello               23            25           2             2
  Ack                  0             0           0             0
  Srefresh             0             0           0             0
  EndtoEnd RSVP        0             0           0             0
so-6/0/0.0  Index 9, State Ena/Up
  NoAuthentication, NoAggregate, NoReliable,  LinkProtection
```

Link protection is enabled.

```
  HelloInterval 9(second)
  Address 10.31.2.2, 10.245.271.53
  ActiveResv 1, PreemptionCnt 0, Update threshold 10%
  Subscription 100%, StaticBW 9.95328Gbps, AvailableBW 9.95328Gbps
  PacketType              Total                 Last 5 seconds
                    Sent      Received        Sent      Received
  Path                12             8           0             0
  PathErr              0             0           0             0
  PathTear             0             1           0             0
  Resv                 9            14           0             1
  ResvErr              0             0           0             0
  ResvTear             0             0           0             0
  Hello                8             8           1             1
  Ack                  0             0           0             0
  Srefresh             0             0           0             0
  EndtoEnd RSVP        0             0           0             0

user@Router3>  show route forwarding-table family mpls
Routing table: ccc
MPLS:
Destination       Type RtRef Next hop          Type Index NhRef Netif
default           perm     0                   dscd     1     1
0                 user     0                   recv     3     3
1                 user     0                   recv     3     3
```

```
2                   user     0                    recv     3     3
100000              user     0                    ulst    77     1
```

This is the main LSP.

```
                                                 Swap       0     so-6/0/0.0
                                                 Swap  100004[0]  so-1/0/0.0
100000(S=0)         user     0                   ulst    78     1
                                                 Pop              so-6/0/0.0
                                                 Swap  100004[0]  so-1/0/0.0
100007              user     0                   ulst    71     1
```

This is the bypass LSP from Router 1 to Router 2.

```
                                                 Swap       0     so-6/0/0.0
                                                 Swap  100003[0]  so-1/0/0.0
100007(S=0)         user     0                   ulst    73     1
                                                 Pop              so-6/0/0.0
                                                 Swap  100003[0]  so-1/0/0.0
```

Case 2: When the Link from Router 1 to Router 3 Is Disabled

```
[edit]
user@Router1# set interfaces so-7/3/2 disable

[edit]
user@Router1# commit
commit complete
```

The primary interface from Router 1 to Router 3 is disabled.

Router 1

```
user@Router1> show mpls lsp
Ingress LSP: 1 sessions
To              From            State  Rt ActivePath       P     LSPname
10.245.71.52    10.245.71.51    Up      0 path1            *     Protected_LSP
```

The main LSP is up.

```
Total 1 displayed, Up 1, Down 0
Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
Transit LSP: 1 sessions
To              From           State Rt Style Labelin Labelout LSPname
10.245.71.52    10.245.271.53  Up     0  1 SE  100004        0
Bypass->10.31.2.1
```

This is the bypass LSP from Router 3 to Router 2.

```
Total 1 displayed, Up 1, Down 0

user@Router1>  show rsvp session detail
Ingress RSVP: 3 sessions
10.245.71.52
```

```
  From: 10.31.3.1, LSPstate: Up, ActiveRoute: 0
  LSPname: Protected_LSP
```

This is the newly signaled backup LSP, as indicated by the To/From field.

```
  Resv style: 1 SE, Label in: -, Label out: 100000
  Time left:    -,  Since: Thu Aug  8 12:29:16 2002
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 2 receiver 33 protocol 0
  Link protection desired
  Type: Backup LSP at Point-of-Local-Repair
  PATH rcvfrom: localclient
  PATH sentto: 10.31.1.2 (so-0/0/0.0) 4 pkts
  RESV rcvfrom: 10.31.2.2  (so-0/0/0.0) 3 pkts
  Explct route: 10.31.2.2 10.31.2.1
  Record route: <self>  10.31.2.2  10.31.2.1
10.245.71.52
  From: 10.245.71.51, LSPstate: Dn, ActiveRoute: 0
```

The original LSP is now down.

```
  LSPname: Protected_LSP
  Resv style: 0 -, Label in: -, Label out: -
  Time left:    -,  Since: Thu Aug  8 12:13:24 2002
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 2 receiver 33 protocol 0
  Link protection desired
  Type: Link protected LSP
  PATH rcvfrom: localclient
  PATH sentto: [no route]
  Explct route: 10.31.1.2 10.31.2.1
  Record route: <self>  ...incomplete

10.245.271.53
  From: 10.245.71.51, LSPstate: Up, ActiveRoute: 1
  LSPname: Bypass->10.31.1.2
```

This is the bypass LSP from Router 1 to Router 2.

```
  Resv style: 1 SE, Label in: -, Label out: 100001
  Time left:    -,  Since: Thu Aug  8 12:26:48 2002
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 52 protocol 0
  Type: Bypass LSP
  PATH rcvfrom: localclient
  PATH sentto: 10.31.3.2 (so-0/0/0.0) 13 pkts
  RESV rcvfrom: 10.31.3.2  (so-0/0/0.0) 13 pkts
  Explct route: 10.31.3.2 10.31.2.2
  Record route: <self>  10.31.3.2  10.31.2.2
Total 3 displayed, Up 2, Down 1

Egress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit RSVP: 1 sessions

10.245.71.52
```

```
  From: 10.245.271.53, LSPstate: Up, ActiveRoute: 0
  LSPname: Bypass->10.31.2.1
```

This is the bypass LSP from Router 3 to Router 2, which will fail in the next case.

```
  Resv style: 1 SE, Label in: 100004, Label out: 0
  Time left:   38,  Since: Thu Aug  8 12:27:07 2002
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 16 protocol 0
  PATH rcvfrom: 10.31.1.2  (so-7/3/2.0) 11 pkts
  PATH sentto: 10.31.3.2 (so-0/0/0.0) 12 pkts
  RESV rcvfrom: 10.31.3.2  (so-0/0/0.0) 12 pkts
  Explct route: 10.31.3.2
  Record route: 10.31.1.2  <self>  10.31.3.2
Total 1 displayed, Up 1, Down 0

user@Router1> show rsvp neighbor detail
RSVP neighbor: 2 learned
Address: 10.31.1.2   via: so-7/3/2.0    status: Down
```

The neighbor is down.

```
  Last changed time: 25, Idle: 25 sec, Up cnt: 2, Down cnt: 2
  Message received: 397
  Hello: sent 900, received: 890, interval: 3 sec
  Remote instance: 0x0, Local instance: 0x41b41b17
  Refresh reduction:  not operational
  Link protection:  enabled
    LSP name: Bypass->10.31.1.2
    Bypass LSP: operational,  Backup routes: 2,  Backup LSPs: 1
    Bypass explicit route: 10.31.3.2 10.31.2.2
Address: 10.31.3.2   via: so-0/0/0.0    status: Up
  Last changed time: 25:40, Idle: 5 sec, Up cnt: 4, Down cnt: 3
  Message received: 502
  Hello: sent 558, received: 538, interval: 20 sec
  Remote instance: 0x194fa7af, Local instance: 0x507b7c2a
  Refresh reduction:  not operational
  Link protection:  disabled
    Bypass LSP: does not exist,  Backup routes: 0,  Backup LSPs: 0
user@Router1> show route 10.31.5.1
inet.0: 23 destinations, 23 routes (22 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both
10.31.5.1/32       *[RSVP/7] 00:03:04, metric 20
                    > via so-0/0/0.0, label-switched-path Bypass->10.31.1.2
```

The route can be reached by way of the backup LSP.

```
user@Router1>  show route forwarding-table destination 10.31.5.1
Routing table: inet
Internet:
Destination        Type RtRef Next hop          Type Index NhRef Netif
10.31.5.1/32       user     0                   Push 100000, Push 100001(top)[0]

so-0/0/0.0
```

Double-stacked labels appear on the backup LSP from Router 1 to Router 2 to Router 3.

Before proceeding to Case 3, reenable the so-7/3/2 interface on Router 1.

```
[edit]
user@Router1# delete interfaces so-7/3/2 disable

[edit]
user@Router1# commit
commit complete
```

Case 3: When the Link from Router 3 to Router 2 Is Disabled

```
[edit]
user@Router3# set interfaces so-6/0/0 disable

[edit]
user@Router3# commit
commit complete
```

The primary interface from Router 3 to Router 2 is disabled.

Router 3

```
user@Router3> show rsvp session
Ingress RSVP: 2 sessions
To              From            State Rt Style Labelin Labelout LSPname
10.245.71.52    10.245.271.53   Up     1  1 SE       -   100005
Bypass->10.31.2.1
10.245.71.52    10.31.1.2       Up     0  1 SE       -        0 Protected_LSP
```

The backup is signaled from Router 3 to Router 2.

```
Total 2 displayed, Up 2, Down 0
Egress RSVP: 1 sessions
To              From            State Rt Style Labelin Labelout LSPname
10.245.271.53   10.245.71.51    Up     0  1 SE       3        -
Bypass->10.31.1.2
Total 1 displayed, Up 1, Down 0
Transit RSVP: 1 sessions
To              From             State  Rt Style Labelin Labelout LSPname
10.245.71.52    10.245.71.51     Dn      0  0  -  100000        - Protected_LSP
```

The main LSP is down.

```
Total 1 displayed, Up 0, Down 1

user@Router3> show rsvp session detail
Ingress RSVP: 2 sessions

10.245.71.52
  From: 10.245.271.53, LSPstate: Up, ActiveRoute: 1
  LSPname: Bypass->10.31.2.1
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 100005
  Resv style: 1 SE, Label in: -, Label out: 100005
  Time left:    -,  Since: Thu Aug  8 12:31:09 2002
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 17 protocol 0
```

```
  Type: Bypass LSP
  PATH rcvfrom: localclient
  PATH sentto: 10.31.1.1 (so-1/0/0.0) 6 pkts
  RESV rcvfrom: 10.31.1.1  (so-1/0/0.0) 6 pkts
  Explct route: 10.31.1.1 10.31.3.2
  Record route: <self>  10.31.1.1  10.31.3.2

10.245.71.52
  From: 10.31.1.2, LSPstate: Up, ActiveRoute: 0
  LSPname: Protected_LSP
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 0
  Resv style: 1 SE, Label in: -, Label out: 0
  Time left:    -,  Since: Thu Aug  8 12:31:59 2002
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 2 receiver 33 protocol 0
  Link protection desired
  Type: Backup LSP at Point-of-Local-Repair
```

This is the backup LSP.

```
  PATH rcvfrom: localclient
  PATH sentto: 10.31.2.1 (so-1/0/0.0) 5 pkts
  RESV rcvfrom: 10.31.3.2  (so-1/0/0.0) 2 pkts
  Explct route: 10.31.3.2
  Record route: <self>  10.31.3.2
Total 2 displayed, Up 2, Down 0

Egress RSVP: 1 sessions

10.245.271.53
  From: 10.245.71.51, LSPstate: Up, ActiveRoute: 0
  LSPname: Bypass->10.31.1.2
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: -
  Resv style: 1 SE, Label in: 3, Label out: -
  Time left:   31,  Since: Thu Aug  8 12:26:48 2002
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 52 protocol 0
  PATH rcvfrom: 10.31.2.1  (so-6/0/0.0) 23 pkts
  PATH sentto: localclient
  RESV rcvfrom: localclient
  Record route: 10.31.3.1  10.31.2.1  <self>
Total 1 displayed, Up 1, Down 0

Transit RSVP: 1 sessions

10.245.71.52
  From: 10.245.71.51, LSPstate: Dn, ActiveRoute: 0
  LSPname: Protected_LSP
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: -
  Resv style: 0 -, Label in: 100000, Label out: -
  Time left:   53,  Since: Thu Aug  8 12:26:39 2002
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 2 receiver 33 protocol 0
  Link protection desired
  Type: Link protected LSP
  PATH rcvfrom: 10.31.1.1  (so-1/0/0.0) 30 pkts
  PATH sentto: [no route]
```

```
  Explct route: 10.31.2.1
  Record route: 10.31.1.1  <self>  ...incomplete
Total 1 displayed, Up 0, Down 1

user@Router3> show route forwarding-table family mpls
Routing table: ccc
MPLS:
Destination        Type RtRef Next hop          Type Index NhRef Netif
default            perm     0                   dscd     1     1
0                  user     0                   recv     3     3
1                  user     0                   recv     3     3
2                  user     0                   recv     3     3
100000             user     0                   Swap  100005[0]  so-1/0/0.0
```

This shows label swapping for the main LSP traveling over the backup LSP through Router 2.

Router 1

```
user@Router1> show rsvp session detail
Ingress RSVP: 1 sessions
10.245.71.52
  From: 10.245.71.51, LSPstate: Up, ActiveRoute: 1
  LSPname: Protected_LSP
```

The main LSP is not affected.

```
  Resv style: 1 SE, Label in: -, Label out: 100000
  Time left:    -,  Since: Thu Aug  8 12:13:24 2002
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 2 receiver 33 protocol 0
  Link protection desired
  PATH rcvfrom: localclient
  PATH sentto: 10.31.1.2 (so-7/3/2.0) 95 pkts
  RESV rcvfrom: 10.31.1.2  (so-7/3/2.0) 87 pkts
  Explct route: 10.31.1.2 10.31.2.1
  Record route: <self>  10.31.1.2  10.31.3.2
Total 1 displayed, Up 1, Down 0

Egress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit RSVP: 1 sessions

10.245.71.52
  From: 10.245.271.53, LSPstate: Up, ActiveRoute: 0
  LSPname: Bypass->10.31.2.1
```

The bypass LSP is listed, because Router 1 does not detect the backup from Router 3 to Router 2.

```
  Resv style: 1 SE, Label in: 100005, Label out: 0
  Time left:   53,  Since: Thu Aug  8 12:31:09 2002
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 17 protocol 0
  PATH rcvfrom: 10.31.1.2  (so-7/3/2.0) 11 pkts
  PATH sentto: 10.31.3.2 (so-0/0/0.0) 11 pkts
  RESV rcvfrom: 10.31.3.2  (so-0/0/0.0) 11 pkts
  Explct route: 10.31.3.2
```

```
  Record route: 10.31.1.2  <self>  10.31.3.2
Total 1 displayed, Up 1, Down 0

user@Router1> show route forwarding-table family mpls
Routing table: ccc
MPLS:
Destination        Type RtRef Next hop          Type Index NhRef Netif
default            perm     0                   dscd     8     1
0                  user     0                   recv    10     3
1                  user     0                   recv    10     3
2                  user     0                   recv    10     3
100005             user     0                   Swap       0     so-0/0/0.0
```

Labels are swapped on the bypass LSP.

```
100005(S=0)        user     0                   Pop              so-0/0/0.0
user@Router1> show route forwarding-table destination 10.31.5.1 extensive
Routing table: inet [Index 0]
Internet:
Destination:  10.31.5.1/32
  Route type: user               Route reference: 0
  Flags: sent to PFE
  Next-hop type: Push  100000
```

There is only one entry here. The bypass to Router 3 is broken because the connection between Router 3 and Router 2 is disabled.

```
Next-hop interface: so-7/3/2.0
user@Router1> show isis adjacency
Interface             System         L State        Hold (secs) SNPA
so-0/0/0.0            Router2        2 Up                   24
so-7/3/2.0            Router3        2 Up                   23

user@Router1>  show route 10.31.5.1 extensive

inet.0: 22 destinations, 22 routes (21 active, 0 holddown, 1 hidden)
10.31.5.1/32 (1 entry, 1 announced)
        State: <FlashAll>
TSI:
KRT in-kernel 10.31.5.1/32 -> {0.0.0.0}
        *RSVP   Preference: 7
                Next hop: via so-7/3/2.0 weight 1, selected
```

Only one entry is shown.

```
                Label-switched-path Protected_LSP
                Label operation: Push 100000
                State: <Active Int>
                Local AS:    69
                Age: 2:21       Metric: 100
                Task: RSVP
                Announcement bits (1): 0-KRT
                AS path: I

user@Router1> show rsvp neighbor detail
RSVP neighbor: 2 learned
Address: 10.31.1.2   via: so-7/3/2.0    status: Up
  Last changed time: 2:58, Idle: 0 sec, Up cnt: 3, Down cnt: 2
```

```
  Message received: 433
  Hello: sent 965, received: 950, interval: 3 sec
  Remote instance: 0x74be7c42, Local instance: 0x41b41b17
  Refresh reduction:  not operational
  Link protection:  disabled
    Bypass LSP: does not exist,  Backup routes: 0,  Backup LSPs: 0
Address: 10.31.3.2   via: so-0/0/0.0    status: Up
  Last changed time: 30:54, Idle: 0 sec, Up cnt: 4, Down cnt: 3
  Message received: 533
  Hello: sent 593, received: 573, interval: 20 sec
  Remote instance: 0x194fa7af, Local instance: 0x507b7c2a
  Refresh reduction:  not operational
  Link protection:  disabled
    Bypass LSP: does not exist,  Backup routes: 0,  Backup LSPs: 0

user@Router1>  show rsvp session
Ingress RSVP: 1 sessions
To              From            State Rt Style Labelin Labelout LSPname
10.245.71.52    10.245.71.51    Up     1  1 SE       -   100000 Protected_LSP
Total 1 displayed, Up 1, Down 0
Egress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0
Transit RSVP: 1 sessions
To              From            State Rt Style Labelin Labelout LSPname
10.245.71.52    10.245.271.53   Up     0  1 SE  100005        0
Bypass->10.31.2.1
Total 1 displayed, Up 1, Down 0
```

Example: Node-Link Protection Configuration

Figure 11: Node-Link Protection Topology Diagram

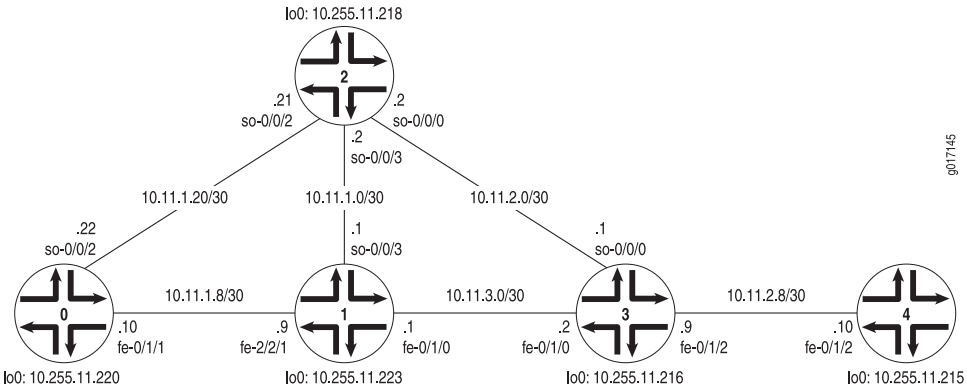


Figure 11 on page 129 shows an example of how you can implement node-link protection. An LSP is initiated on Router 0 with a strict path travelling through Router 1, Router 2, Router 3, and Router 4. You configure node-link protection within the LSP and link protection on all RSVP interfaces in the path.

On Router 0, configure an LSP to travel across routers 1, 2, 3, and 4. Include the node-link-protection statement in the LSP and configure link protection on outgoing RSVP interface fe-0/1/1. To support the LSP, configure OSPF, MPLS, and RSVP on the needed interfaces.

Router 0

```
[edit]
interfaces {
    fe-0/1/1 {
        unit 0 {
            family inet {
                address 10.11.1.10/30;
            }
            family mpls;
        }
    }
    so-0/0/2 {
        unit 0 {
            family inet {
                address 10.11.1.22/30;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.11.220/32;
            }
        }
    }
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface lo0.0 {
                passive;
            }
            interface fe-0/1/1.0;
            interface so-0/0/2.0;
        }
        traffic-engineering;
    }
    mpls {
        statistics {
            file mplsStat.log size 1m;
            interval 30;
            display-id;
        }
        traffic-engineering bgp-igp-both-ribs;
        traceoptions {
            file mpls.log size 5m world-readable;
            flag cspf;
            flag cspf-link;
            flag cspf-node;
            flag state;
            flag error;
        }
        explicit-null;
        label-switched-path test_r0_r4 {
            from 10.255.11.220;
            to 10.255.11.215;
```

```
        node-link-protection; # Apply node-link protection to the LSP.
        primary pathP;
      }
      path pathP {# Define the LSP path across routers 1, 2, 3, and 4.
        10.11.1.9 strict;
        10.11.1.2 strict;
        10.11.2.1 strict;
        10.11.2.10 strict;
      }
      interface fe-0/1/1.0;
      interface so-0/0/2.0;
    }
    rsvp {
      traceoptions {
        file rsvp.log size 3m files 12 world-readable;
        flag event;
        flag state;
        flag error;
        flag packets detail;
      }
      interface fe-0/1/1.0 {
        link-protection; # Apply link protection to RSVP interfaces in the LSP path.
      }
      interface so-0/0/2.0;
    }
  }
```

On Router 1, configure link protection on outgoing RSVP interface so-0/0/3. Configure OSPF, MPLS, and RSVP on all transit interfaces.

Router 1

```
[edit]
interfaces {
  fe-0/1/0 {
    unit 0 {
      family inet {
        address 10.11.3.1/30;
      }
      family mpls;
    }
  }
  fe-2/2/1 {
    unit 0 {
      family inet {
        address 10.11.1.9/30;
      }
      family mpls;
    }
  }
  so-0/0/3 {
    unit 0 {
      family inet {
        address 10.11.1.1/30;
      }
      family mpls;
    }
  }
```

```
    lo0 {
        unit 0 {
            family inet {
                address 10.255.11.223/32;
            }
        }
    }
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface lo0.0 {
                passive;
            }
            interface fe-0/1/0.0;
            interface fe-2/2/1.0;
            interface so-0/0/3.0;
        }
        traffic-engineering;
    }
    mpls {
        traffic-engineering bgp-igp-both-ribs;
        explicit-null;
        interface fe-0/1/0.0;
        interface fe-2/2/1.0;
        interface so-0/0/3.0;
    }
    rsvp {
        interface fe-0/1/0.0;
        interface fe-2/2/1.0;
        interface so-0/0/3.0 {
            link-protection; # Apply link protection on all RSVP interfaces in the LSP path.
        }
    }
}
```

On Router 2, configure link protection on outgoing RSVP interface so-0/0/0. Configure OSPF, MPLS, and RSVP on all transit interfaces.

Router 2

```
[edit]
interfaces {
    so-0/0/0 {
        unit 0 {
            family inet {
                address 10.11.2.2/30;
            }
            family mpls;
        }
    }
    so-0/0/2 {
        unit 0 {
            family inet {
                address 10.11.1.21/30;
            }
            family mpls;
        }
```

```
    }
    so-0/0/3 {
      unit 0 {
        family inet {
          address 10.11.1.2/30;
        }
        family mpls;
      }
    }
    lo0 {
      unit 0 {
        family inet {
          address 10.255.11.218/32;
        }
      }
    }
  }
  protocols {
    ospf {
      area 0.0.0.0 {
        interface lo0.0 {
          passive;
        }
        interface so-0/0/0.0;
        interface so-0/0/2.0;
        interface so-0/0/3.0;
      }
      traffic-engineering;
    }
    mpls {
      traffic-engineering bgp-igp-both-ribs;
      explicit-null;
      interface so-0/0/0.0;
      interface so-0/0/2.0;
      interface so-0/0/3.0;
    }
    rsvp {
      interface so-0/0/0.0 {
        link-protection; # Apply link protection to RSVP interfaces in the LSP path.
      }
      interface so-0/0/2.0;
      interface so-0/0/3.0;
    }
  }
```

On Router 3, configure link protection on outgoing RSVP interface fe-0/1/2. Configure OSPF, MPLS, and RSVP on all transit interfaces.

Router 3

```
[edit]
interfaces {
  fe-0/1/0 {
    unit 0 {
      family inet {
        address 10.11.3.2/30;
      }
      family mpls;
```

```
        }
      }
      fe-0/1/2 {
        unit 0 {
          family inet {
            address 10.11.2.9/30;
          }
          family mpls;
        }
      }
      so-0/0/0 {
        unit 0 {
          family inet {
            address 10.11.2.1/30;
          }
          family mpls;
        }
      }
      lo0 {
        unit 0 {
          family inet {
            address 10.255.11.216/32;
          }
        }
      }
    }
    protocols {
      ospf {
        area 0.0.0.0 {
          interface lo0.0 {
            passive;
          }
          interface fe-0/1/0.0;
          interface fe-0/1/2.0;
          interface so-0/0/0.0;
        }
        traffic-engineering;
      }
      mpls {
        traffic-engineering bgp-igp-both-ribs;
        explicit-null;
        interface fe-0/1/0.0;
        interface fe-0/1/2.0;
        interface so-0/0/0.0;
      }
      rsvp {
        interface fe-0/1/0.0;
        interface fe-0/1/2.0 {
          link-protection; # Apply link protection to RSVP interfaces in the LSP path.
        }
        interface so-0/0/0.0;
      }
    }
```

Because Router 4 is the endpoint of the LSP, you can configure interfaces and protocols as usual. There is no need to configure any node-link protection or link protection statements on this router.

Router 4

```
[edit]
interfaces {
    fe-0/1/2 {
        unit 0 {
            family inet {
                address 10.11.2.10/30;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.11.215/32;
            }
        }
    }
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface lo0.0 {
                passive;
            }
            interface fe-0/1/2.0;
        }
        traffic-engineering;
    }
    mpls {
        traffic-engineering bgp-igp-both-ribs;
        explicit-null;
        interface fe-0/1/2.0;
    }
    rsvp {
        interface fe-0/1/2.0;
    }
}
```

Verifying Your Work

To verify proper operation of MPLS LSP node-link protection, use the following commands:

- show mpls lsp extensive
- show route detail
- show rsvp neighbor detail
- show rsvp session detail

The following section shows the output of these commands used with the configuration example.

```
user@router0> show rsvp session detail
Ingress RSVP: 2 sessions
10.255.11.215
  From: 10.255.11.220, LSPstate: Up, ActiveRoute: 5
  LSPname: test_r0_r4, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 100128
  Resv style: 1 SE, Label in: -, Label out: 100128
  Time left:    -, Since: Thu May  8 13:36:58 2003
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 3 receiver 56517 protocol 0
  Node/Link protection desired
  Type: Node/Link protected LSP
  PATH rcvfrom: localclient
  PATH sentto: 10.11.1.9 (fe-0/1/1.0) 20 pkts
  RESV rcvfrom: 10.11.1.9 (fe-0/1/1.0) 37 pkts
  Explct route: 10.11.1.9 10.11.1.2 10.11.2.1 10.11.2.10
  Record route: <self> 10.11.1.9 10.11.1.2 10.11.2.1 10.11.2.10

10.255.11.218
  From: 10.255.11.220, LSPstate: Up, ActiveRoute: 0
  LSPname: Bypass->10.11.1.9->10.11.1.2 # 2 next hops indicate node-link
protection.
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 0
  Resv style: 1 SE, Label in: -, Label out: 0
  Time left:    -, Since: Thu May  8 13:36:58 2003
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 56521 protocol 0
  Type: Bypass LSP
    Number of data route tunnel through: 4
    Number of RSVP session tunnel through: 0
  PATH rcvfrom: localclient
  PATH sentto: 10.11.1.21 (so-0/0/2.0) 1 pkts
  RESV rcvfrom: 10.11.1.21 (so-0/0/2.0) 1 pkts
  Explct route: 10.11.1.21
  Record route: <self> 10.11.1.21
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
```

The show mpls lsp extensive command provides some useful information about link protection and node-link protection. The protection flag entry indicates a series of values. By adding the values together, you can learn the protection state of an LSP based on the total sum. Significant values for the flags include: 1 = Available (Link Protection), 2 = In Use, and 8 = Node Protection. Thus, a value of 9 means that node protection is available (1 + 8 = 9) and a value of A means that a node protected link is in use (8 + 2 = A, in hexadecimal).

```
user@router0> show mpls lsp extensive
Ingress LSP: 1 sessions
```

```
10.255.11.215
  From: 10.255.11.220, State: Up, ActiveRoute: 5, LSPname: test_r0_r4
  ActivePath: pathP (primary)
  Node/Link protection desired
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary   pathP            State: Up
    OptimizeTimer: 30
    Reoptimization in 13 second(s).
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 40)
 10.11.1.9 S 10.11.1.2 S 10.11.2.1 S 10.11.2.10 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node):
          10.11.1.9(flag=9 Label=100128) 10.11.1.2(flag=9 Label=100080)
10.11.2.1(flag=1 Label=100080) 10.11.2.10(Label=0)
   67 May  8 13:46:14 CSPF: computation result ignored[18 times]
   66 May  8 13:37:31 Record Route:  10.11.1.9(flag=9 Label=100128)
10.11.1.2(flag=9 Label=100080) 10.11.2.1(flag=1 Label=100080) 10.11.2.10(Label=0)

   65 May  8 13:37:28 CSPF: computation result ignored
   64 May  8 13:37:07 Record Route:  10.11.1.9(flag=9 Label=100128)
10.11.1.2(flag=1 Label=100080) 10.11.2.1(Label=100080) 10.11.2.10(Label=0)
   63 May  8 13:37:01 Record Route:  10.11.1.9(flag=9 Label=100128)
10.11.1.2(Label=100080) 10.11.2.1(Label=100080) 10.11.2.10(Label=0)
   62 May  8 13:37:01 Link-protection Up
   61 May  8 13:36:58 Selected as active path
   60 May  8 13:36:58 Record Route:  10.11.1.9(Label=100128)
10.11.1.2(Label=100080) 10.11.2.1(Label=100080) 10.11.2.10(Label=0)
   59 May  8 13:36:58 Up
   58 May  8 13:36:58 Originate Call
   57 May  8 13:36:58 CSPF: computation result accepted
   56 May  8 13:36:29 CSPF failed: no route toward 10.11.2.10[10 times]
   55 May  8 13:32:04 Clear Call
   54 May  8 13:31:40 Deselected as active
   53 May  8 13:31:40 Link-protection Down
   52 May  8 13:31:40 Down
   51 May  8 13:31:36 CSPF failed: no route toward 10.11.2.10[6 times]
   50 May  8 13:29:00 10.11.2.1: Session preempted
   49 May  8 13:28:42 Record Route:  10.11.1.9(flag=9 Label=100064)
10.11.1.2(flag=9 Label=100064) 10.11.2.1(Label=100064) 10.11.2.10(Label=0)
   48 May  8 13:28:40 CSPF failed: no route toward 10.11.2.10
   47 May  8 13:28:35 Link-protection Up
   46 May  8 13:28:35 Link-protection Down
   45 May  8 13:28:30 Link-protection Up
   44 May  8 13:28:30 Link-protection Down
   43 May  8 13:28:10 CSPF failed: no route toward 10.11.2.10
   42 May  8 13:27:44 Link-protection Up
   41 May  8 13:27:44 Link-protection Down
   40 May  8 13:27:42 Link-protection Up
   39 May  8 13:27:42 Record Route:  10.11.1.9(flag=9 Label=100064)
10.11.1.2(flag=9 Label=100064) 10.11.2.1(flag=1 Label=100064) 10.11.2.10(Label=0)

   38 May  8 13:27:41 CSPF failed: no route toward 10.11.2.10[2 times]
   37 May  8 13:27:39 CSPF: link down/deleted
10.11.2.9(eagle.00/10.255.11.216)->0.0.0.0(eagle.04/0.0.0.0)
   36 May  8 13:27:39 Link-protection Down
   35 May  8 13:27:39 Record Route:  10.11.1.9(Label=100064)
10.11.1.2(Label=100064) 10.11.2.1(Label=100064) 10.11.2.10(Label=0)
   34 May  8 13:27:39 CSPF failed: no route toward 10.11.2.10
   33 May  8 13:27:39 CSPF: link down/deleted
0.0.0.0(eagle.04/0.0.0.0)->0.0.0.0(papst.00/10.255.11.215)
   32 May  8 13:27:12 CSPF: computation result ignored[16 times]
```

```
   31 May  8 13:19:35 Record Route:  10.11.1.9(flag=9 Label=100064)
10.11.1.2(flag=9 Label=100048) 10.11.2.1(flag=1 Label=100048) 10.11.2.10(Label=0)

   30 May  8 13:19:22 Link-protection Up
   29 May  8 13:19:22 Record Route:  10.11.1.9(flag=9 Label=100064)
10.11.1.2(flag=9 Label=100048) 10.11.2.1(Label=100048) 10.11.2.10(Label=0)
   28 May  8 13:19:22 Up
   27 May  8 13:19:22 Link-protection Down
   26 May  8 13:19:22 CSPF: computation result accepted
   25 May  8 13:19:16 Link-protection Up
   24 May  8 13:19:16 Link-protection Down
   23 May  8 13:18:54 CSPF failed: no route toward 10.11.2.1
   22 May  8 13:18:54 CSPF: link down/deleted
0.0.0.0(eagle.04/0.0.0.0)->0.0.0.0(papst.00/10.255.11.215)
   21 May  8 13:18:53 CSPF failed: no route toward 10.11.2.1[2 times]
   20 May  8 13:18:46 CSPF: link down/deleted
10.11.2.9(eagle.00/10.255.11.216)->0.0.0.0(eagle.04/0.0.0.0)
   19 May  8 13:18:35 Record Route:  10.11.1.9(flag=9 Label=100032)
10.11.1.2(flag=9 Label=100032) 10.11.2.1(Label=100016) 10.11.2.10(Label=0)
   18 May  8 13:18:35 Record Route:  10.11.1.9(flag=9 Label=100032)
10.11.1.2(Label=100032) 10.11.2.1(Label=100016) 10.11.2.10(Label=0)
  Created: Thu May  8 13:13:28 2003
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

user@router0>  show rsvp neighbor detail
RSVP neighbor: 2 learned
Address: 10.11.1.9 via: fe-0/1/1.0 status: Up
  Last changed time: 33:02, Idle: 5 sec, Up cnt: 1, Down cnt: 0
  Message received: 130
  Hello: sent 221, received: 221, interval: 9 sec
  Remote instance: 0x66368e80, Local instance: 0x643f57b5
  Refresh reduction: incomplete
    Remote end: enabled, Ack-extension: disabled

Address: 10.11.1.21 via: so-0/0/2.0 status: Up
  Last changed time: 32:41, Idle: 10 sec, Up cnt: 1, Down cnt: 0
  Message received: 78
  Hello: sent 218, received: 218, interval: 9 sec
  Remote instance: 0x74b57f2a, Local instance: 0x66341d2f
  Refresh reduction:  operational
    Remote end: enabled, Ack-extension: enabled

user@router0>  show route 10.255.11.215 detail
inet.0: 33 destinations, 34 routes (31 active, 0 holddown, 2 hidden)
10.255.11.215/32 (2 entries, 1 announced)
        State: <FlashAll>
        *RSVP   Preference: 7
                Next hop: 10.11.1.9 via fe-0/1/1.0 weight 1, selected
                Label-switched-path test_r0_r4
                Label operation: Push 100128, selfID=RSVP-7
                 Next hop: via so-0/0/2.0 weight 20001
                 Label-switched-path Bypass->10.11.1.9->10.11.1.2
                Label operation: Push 100080, selfID=RSVP-7, parentID=RSVP-8
                State: <Active Int>
                Local AS:    69
                Age: 13:14      Metric: 40
                Task: RSVP
```

```
                Announcement bits (2): 0-KRT 3-Resolve inet.0
                AS path: I
         IS-IS  Preference: 18
                Level: 2
                Next hop: 10.11.1.9 via fe-0/1/1.0, selected
                State: <Int>
                Inactive reason: Route Preference
                Local AS:    69
                Age: 13:20      Metric: 40
                Task: IS-IS
                AS path: I
inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
10.255.11.215/32 (1 entry, 1 announced)
        State: <FlashAll>
        *RSVP   Preference: 7
                Next hop: 10.11.1.9 via fe-0/1/1.0 weight 1, selected
                Label-switched-path test_r0_r4
                Label operation: Push 100128, selfID=RSVP-7
                Next hop: via so-0/0/2.0 weight 20001
                Label-switched-path Bypass->10.11.1.9->10.11.1.2
                Label operation: Push 100080, selfID=RSVP-7, parentID=RSVP-8
                State: <Active Int>
                Local AS:    69
                Age: 13:14      Metric: 40
                Task: RSVP
                Announcement bits (1): 1-Resolve inet.0
                AS path: I
```

For More Information

For additional information about MPLS LSP link protection or node-link protection, see the following:

- *JUNOS MPLS Applications Configuration Guide*
- RFC 4090, *Fast Reroute Extensions to RSVP-TE for LSP Tunnels*

Revision History

5 October 2009—10.0R1 Release. Roy Spencer.

13 June 2009—9.6R1 Release. Roy Spencer.

13 April 2009—9.5R1 Release. Roy Spencer.

15 January 2009—9.4R1 Release. Roy Spencer.

10 October 2008—9.3R1 Release. Roy Spencer.

10 July 2008—9.2R1 Release. Roy Spencer.

10 April 2008—9.1R1 Release. Roy Spencer.

29 June 2007—8.4R1 Release. Fawn Damitio.

27 March 2007—8.3I Release. Fawn Damitio.

12 January 2007—Added support on MX960 Ethernet Services Routers. 8.2R1 Release. Fawn Damitio.

15 September 2006—8.1R1 Release. Richard Hendricks.

29 June 2006—8.0R1 Release. Richard Hendricks.

27 March 2006—7.6R1 Release. Richard Hendricks.

9 January 2006—7.5R1 Release. Richard Hendricks.

14 September 2005—Added support for enhanced operational commands and system log messages for link protection and node-link protection, 7.4R1 Release. Richard Hendricks.

13 June 2005—Added support for link protection of point-to-multipoint LSPs and for class of service on link-protected LSPs and bypass LSPs, 7.3R1 Release. Richard Hendricks.

5 April 2005—7.2R1 Release. Richard Hendricks.

2 February 2005—Added information about configuring multiple bypass LSPs, manual bypass LSPs, and link protection priority, 7.1R1 Release. Richard Hendricks.

6 October 2004—7.0R1 Release. Richard Hendricks.

6 July 2004—6.4R1 Release. Richard Hendricks.

5 April 2004—6.3R1 Release. Richard Hendricks.

22 December 2003—6.2R1 Release. Richard Hendricks.

22 September 2003—6.1R1 Release. Richard Hendricks.

30 June 2003—Added node-link protection, 6.0R1 Release. Richard Hendricks.

2 April 2003—5.7R1 Release. Richard Hendricks.

27 December 2002—5.6R1 Release. Richard Hendricks.

30 September 2002—New link protection example added for the 5.5R1 Release. Richard Hendricks.

19 July 2002—5.4R1 Release. Richard Hendricks.

15 July 2002—Document revised based on comments from Ping Pan. Richard Hendricks.

28 May 2002—Document revised and reformatted. Richard Hendricks.

18 March 2002—Initial document written. Greg Ketell.

Chapter 5

RSVP LSP Tunnels

This chapter covers the following topics:

- RSVP LSP Tunnels Overview on page 141
- RSVP LSP Tunnels System Requirements on page 142
- RSVP LSP Tunnels Terms and Acronyms on page 142
- RSVP LSP Tunneling Operation on page 143
- Configuring an RSVP LSP Tunnel on page 143
- Configuring Link Management Protocol Traffic Engineering Links on page 144
- Configuring Link Management Protocol Peers on page 144
- Configuring Peer Interfaces in OSPF and RSVP on page 145
- Establishing FA-LSP Path Information on page 145
- Defining Label-Switched Paths for the FA-LSP on page 146
- Creating End-to-End LSPs to Traverse the FA-LSP on page 146
- Option: Tearing Down RSVP LSPs Gracefully on page 146
- Example: RSVP LSP Tunnel Configuration on page 147
- For More Information on page 165
- Revision History on page 165

RSVP LSP Tunnels Overview

A Resource Reservation Protocol (RSVP) label-switched path (LSP) tunnel allows you to send RSVP LSPs inside other RSVP LSPs. This allows 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 Multiprotocol Label Switching (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 “GMPLS Overview” on page 4.)

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 Software 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 travelling to virtual peer interfaces defined in an LMP configuration.

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.

RSVP LSP Tunnels System Requirements

To implement an RSVP LSP tunnel, your system must meet these minimum requirements:

- JUNOS Release 8.2 or later for support on MX Series routing platforms
- JUNOS Release 7.4 or later for RSVP LSP tunnel support on M Series and T Series routers
- Two Juniper Networks M Series, MX Series, or T Series routers for the ingress and egress points of the forwarding adjacency, and two Juniper Networks M Series, MX Series, or T Series routers for the ingress and egress PE routers

RSVP LSP Tunnels Terms and Acronyms

F

forwarding adjacency A forwarding path for sending data between peer devices in an RSVP LSP tunnel network.

forwarding adjacency LSP (FA-LSP) A hierarchical RSVP LSP that provides a tunnel-like forwarding adjacency for other packet-based LSPs.

L

Link Management Protocol A protocol used to define forwarding adjacencies between RSVP LSP tunnel peers and to maintain and allocate resources on traffic engineering links.

T

traffic engineering link A logical connection between devices used to support RSVP LSP tunnels. traffic engineering links can have addresses or IDs and are associated with certain resources or interfaces. Each traffic engineering link represents a forwarding adjacency between a pair of devices.

RSVP LSP Tunneling Operation

An RSVP LSP tunnel requires close interaction between LMP, RSVP, and OSPF. The following sequence of events describes how this works:

1. LMP notifies RSVP and OSPF of the control peer, the control adjacency, and resources for the traffic engineering link.
2. GMPLS extracts the LSP attributes from the configuration and requests RSVP to signal one or more specific paths, 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 successfully finds a resource matching the attributes, label allocation succeeds. RSVP sends a PathMsg hop-by-hop until it reaches the target router.
4. The target router, on receiving the RSVP PathMsg, requests that LMP allocate a resource based on the signaled parameters. If label allocation succeeds, it sends back a ResvMsg.
5. If the signaling is successful, an RSVP LSP tunnel is provisioned.

Configuring an RSVP LSP Tunnel

You must complete the following tasks to implement an RSVP LSP tunnel:

- Configuring Link Management Protocol Traffic Engineering Links on page 144
- Configuring Link Management Protocol Peers on page 144
- Configuring Peer Interfaces in OSPF and RSVP on page 145
- Establishing FA-LSP Path Information on page 145
- Defining Label-Switched Paths for the FA-LSP on page 146

- Creating End-to-End LSPs to Traverse the FA-LSP on page 146
- Option: Tearing Down RSVP LSPs Gracefully on page 146

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 Link Management Protocol Peers

After you set up traffic engineering links, configure LMP network peers with the peer 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 configuration will fail to commit.

```
[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 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-name* statement configured in LMP at the [edit protocols link-management peer] 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.
            }
         }
      }
   }
}
```

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.

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.
      }
   }
}
```

Creating End-to-End LSPs to Traverse the FA-LSP

After you create the FA-LSPs and traffic engineering links, you can configure end-to-end LSPs to travel across the FA-LSPs. Here are some general guidelines for this step:

- In general, it is best to start your end-to-end LSP at least one hop before the ingress of the FA-LSP and terminate your LSP at least one hop after the egress of the FA-LSP. If you do originate both the end-to-end LSP and the FA-LSP on the same routing platform, you must disable CSPF and configure strict paths for the LSPs.
- If you configure CSPF, OSPF, and RSVP to create the FA-LSP, the LSP should automatically use the FA-LSP because of the preferred lower metric generated by the reduced number of hops.
- If you disable CSPF on the FA-LSP with the no-cspf option at the [edit protocols mpls label-switched-path] hierarchy level, you must configure the peer interface as a strict next hop in the path for your regular LSP. To manually configure an LSP to travel over the FA-LSP, include the *peer-name* statement at the [edit protocols mpls path *path-name*] hierarchy level and include the strict option.

For more information about creating RSVP LSPs, see the *JUNOS MPLS Applications Configuration Guide*.

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. To configure, include 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.

Example: RSVP LSP Tunnel Configuration

Figure 12: RSVP LSP Tunnel Topology Diagram

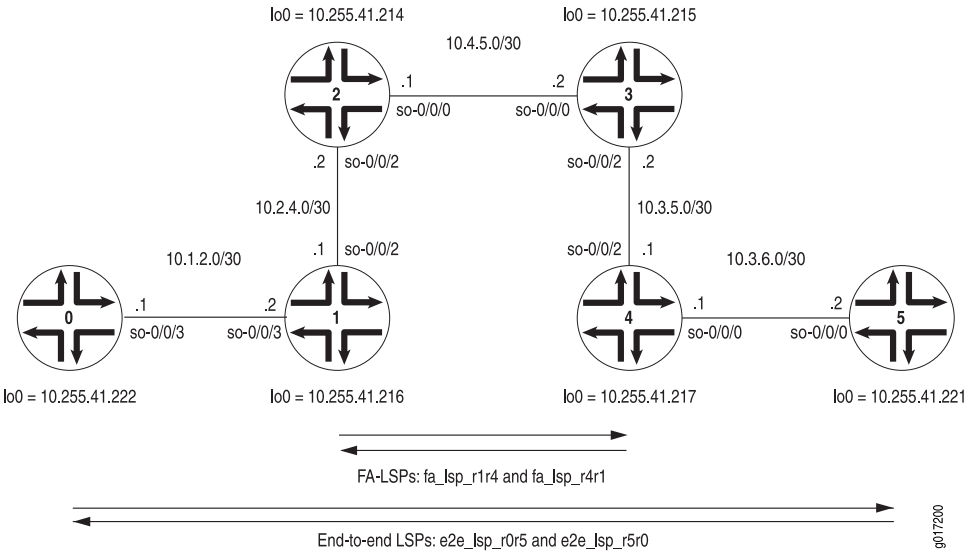


Figure 12 on page 147 shows an end-to-end RSVP LSP called e2e_lsp_r0r5 that originates on Router 0 and terminates on Router 5. In transit, this LSP traverses the FA-LSP fa_lsp_r1r4. The return path is represented by the end-to-end RSVP LSP e2e_lsp_r5r0 that travels over the FA-LSP fa_lsp_r4r1.

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 page 160
- Router 1 on page 164

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
```

For More Information

For additional information about implementing RSVP LSP tunnels, see the following:

- *JUNOS MPLS Applications Configuration Guide*
- RFC 2205, *Resource ReSerVation Protocol (RSVP)*
- Internet draft draft-ietf-mpls-bundle-04.txt, *Link Bundling in MPLS Traffic Engineering* (expires January 2003)
- Internet draft draft-ietf-ccamp-lmp-10.txt, *Link Management Protocol (LMP)* (expires April 2004)
- Internet draft draft-ietf-mpls-lsp-hierarchy-08.txt, *LSP Hierarchy with Generalized MPLS TE* (expires March 2003)

Revision History

5 October 2009—10.0R1 Release. Roy Spencer.

13 June 2009—9.6R1 Release. Roy Spencer.

13 April 2009—9.5R1 Release. Roy Spencer.

15 January 2009—9.4R1 Release. Roy Spencer.

10 October 2008—9.3R1 Release. Roy Spencer.

10 July 2008—9.2R1 Release. Roy Spencer.

10 April 2008—9.1R1 Release. Roy Spencer.

1 February 2008—9.0R1 Release. Fawn Damitio.

29 June 2007—8.4R1 Release. Fawn Damitio.

27 March 2007—8.3R1 Release. Fawn Damitio.

12 January 2007—Added support for MX960 Ethernet Services Routers. 8.2R1 Fawn Damitio.

15 September 2006—8.1R1 Release. Richard Hendricks.

29 June 2006—8.0R1 Release. Richard Hendricks.

27 March 2006—7.6R1 Release. Richard Hendricks.

9 January 2006—7.5R1 Release. Richard Hendricks.

14 September 2005—Initial document written, 7.4R1 Release. Richard Hendricks.

Chapter 6

Simplified Interinstance Route Sharing

This chapter covers the following topics:

- Simplified Interinstance Route Sharing Overview on page 167
- Simplified Interinstance Route Sharing System Requirements on page 168
- Simplified Interinstance Route Sharing Terms and Acronyms on page 169
- Simplified Interinstance Configuration on page 169
- Instance Export Using an IGP Export Policy on page 171
- Configuring Overlapping VPNs on page 171
- Example: Configuring Overlapping VPNs on page 175
- Configuring Nonforwarding Instances on page 183
- Example: Nonforwarding Instances Configuration on page 185
- For More Information on page 189
- Revision History on page 189

Simplified Interinstance Route Sharing Overview

When support for multiple virtual private network (VPN) routing and forwarding (VRF) instances was added to the JUNOS Software, the import and export of routes to other instances and routing protocols from individual instances caused an issue. Interinstance route sharing required configuration of routing table groups (sometimes referred to as routing information base [RIB] groups) for every routing instance that exported routes to other tables.

Policy-based interinstance export in JUNOS Release 5.4 and later simplifies the configuration requirements for users, maintains existing functionality, and, when possible, eliminates the use of routing table groups. This document explores in detail the streamlined configuration hierarchy that has been created for interinstance route sharing.

In JUNOS Release 5.3 and earlier, interinstance route sharing often required configuration of routing table groups by means of the rib-group statement. Although these configurations performed well, the routing table group technique had several limitations:

- Lack of intuitiveness—A routing table group is an unfamiliar configuration construct for many users.
- Complex configuration requirements—Routing table groups specify a primary import routing table that must match the routing table of the VRF instance on which they are applied. Thus, a different routing table group is defined for each of the instances that participate in interinstance route export.
- Redundancy—The information imported and exported by the routing table groups is already present in the router or can be deduced from most configurations (for example, overlapping VPNs).
- Per-protocol configuration—Routing table groups must be applied to every protocol containing routes designated for export.

There are two typical situations in which interinstance export is used:

- Overlapping VPNs—VPN configurations where more than one VRF instance lists the same community route target in a vrf-import policy. In this case, the use of routing table groups is particularly tricky. Incoming routes from other provider edge (PE) routers are automatically imported according to the community route targets, but local VRFs require additional configuration.
- Nonforwarding instances—Multilevel interior gateway protocols (IGPs) that have multiple routing instances and perform route sharing through interinstance route export. The IGP export policy contains the specific instances that are permitted to advertise routes.

These two scenarios differ in the way that policy clauses are specified (route targets in the VRF case; instances in the IGP routing instance case), but are similar in that import and export routes can be deduced by examining the policy configuration. In this guide, you can learn about new hierarchy statements that simplify interinstance route sharing, such as auto-export, instance-import, and instance-export.

Simplified Interinstance Route Sharing System Requirements

To implement simplified interinstance route sharing, your system must meet these minimum requirements:

- JUNOS Release 8.2 or later for support on MX Series routing platforms
- JUNOS Release 5.4 or later for simplified interinstance route sharing on M Series and T Series routers
- Four Juniper Networks M Series, MX Series, or T Series routers for the overlapping VPNs example: Two routers act as PE routers and two act as customer edge (CE) routers
- Six Juniper Networks M Series, MX Series, or T Series routers for the nonforwarding instance example: Two act as PE routers and four act as CE routers

Simplified Interinstance Route Sharing Terms and Acronyms

R

RIB group A routing table group. The group is a master routing table for individual routing tables and stores information about routes that are shared between instances.

V

VPN routing and forwarding (VRF) instance A private routing table created for an individual VPN customer. For more information about VRFs, see the *JUNOS VPNs Configuration Guide*.

Simplified Interinstance Configuration

By changing the configuration format of interinstance export policies, JUNOS Release 5.4 software makes it easier to share routes between VRF instances, other types of instances (such as nonforwarding instances), and IGPs.

VRF instances can share routes with the auto-export statement. When you configure auto-export, the vrf-import and vrf-export policies are compared across all VRF instances. If there is a common route target community between the instances, the routes are shared.

For VRF instances, such as overlapping VPNs, the basic hierarchy levels for auto-export are as follows:

```
[edit]
routing-instances {
   instance-name {
      routing-options {
         auto-export;
      }
   }
}
```

For nonforwarding instances, routes are imported into the instance so routing protocols can announce them. For more information, see “Configuring Nonforwarding Instances” on page 183.

A third option for interinstance export is using an Interior Gateway Protocol (IGP), such as Open Shortest Path First (OSPF) or Intermediate System-to-Intermediate System (IS-IS). An example of the hierarchy used for IGP import and export is listed below.

```
[edit]
routing-options {
   instance-import;
   instance-export;
   auto-export {
```

```
        (enable | disable);
        family inet {
            unicast {
                (enable | disable);
                rib-group;
            }
            multicast {
                (enable | disable);
                rib-group;
            }
        }
    }
}
```

When configuring interinstance route sharing, keep this information in mind:

- The instance-import and instance-export commands cannot be used with VRF instances. They are equivalent to the vrf-import and vrf-export VRF-specific commands.
- Traceoptions in the master instance apply to the routing table export task. Consequently, such traceoptions are propagated to all other instances, although they can be modified as needed.
- The auto-export statement applies to VRF and non-VRF instances.
- Use of the command instance-import automatically enables auto-export for non-VRF instances.
- Some network administrators use the instance-import functionality to create communities of interest. By setting up different VPNs and sharing routes between instances as needed, administrators can tailor services to the needs of their customers. For an example, see “Configuring Nonforwarding Instances” on page 183.

To save time when configuring interinstance parameters on multiple instances, you can configure auto-export to be the default behavior for all your routing instances by means of a configuration group.

```
[edit]
groups {
    vrf-export-on {
        routing-instances {
            routing-options {
                auto-export;
            }
        }
    }
}
```

At the desired hierarchy level, you apply the configuration group and all members of the group receive the same policy:

```
apply-groups vrf-export-on;
```

Instance Export Using an IGP Export Policy

Current configurations that use routing table groups define a policy with a from *instance* statement to select routes in an IGP export policy. However, no policy controls the export process itself. Therefore, the configuration has been simplified so that you do not need to specify additional policies to control the export process. Many current interinstance implementations use an IGP export policy model. The policy model has been extended to support both interinstance route export and IGP export.

If the [edit routing-options auto-export] hierarchy is enabled when an instance-import policy has not been defined, OSPF and IS-IS export policies are automatically examined for the presence of from *instance* statements. If these statements are present, the instance-import policy is selected.

This verification process prevents attribute changes from being applied twice. It also prevents the second policy (IGP export policy) from causing conflicting routing choices. The following is an example of a configuration using an IGP export policy:

```
[edit]
policy-options {
   policy-statement {
      red-ospf-export {
         from instance blue;
         then {
            tag 1;
            accept;
         }
      }
   }
}
routing-instances {
   red {
      routing-options {
         auto-export;
      }
      protocols ospf {
         export red-ospf-export;
      }
   }
   blue {
   }
}
```

Configuring Overlapping VPNs

Policy-based instance export automatically exports routes between VRF instances that refer to the same route target community. If this feature is enabled, a VRF-target tree is constructed by examining the vrf-import and vrf-export policies configured on the system. When an instance refers to a given target in its vrf-import policy, this instance is added to the import list of the target. Similarly, if the instance refers to a specific route target in its vrf-export policy, the instance is added to the export list

for that target. Route targets that contain a single importer that matches a single exporter, or that lack importers and exporters altogether, are ignored by the router when policies are evaluated.

The "rt-export" module tracks changes in routing tables that export a specified route target. When routing changes occur, the vpn-export policy of an instance is applied to the route. Also, if allowed, the route will be imported to all the import tables (subject to vrf-import policy) of the route targets set by the export policy.

The auto-export statement is particularly useful for configuring overlapping VPNs. The auto-export statement determines which routing tables to export routes from and import routes to by examining the existing policy configuration, which can include vrf-target configuration. (For more information on the vrf-target statement, see the *JUNOS VPNs Configuration Guide*.)

When you use the auto-export statement, the behavior varies significantly from the behavior of the rib-groups statement. With the auto-export statement, only the primary route from the originating routing table is exported. In addition, routes exported from the originating VRF to another on the same PE router honor the export policy changes to route attributes. As a result, you must add each originating route target to the exported routes when you use the auto-export statement.

The next sample configuration uses a Border Gateway Protocol (BGP) session between a PE and a CE router. It shows the configuration changes required when you use the autoexport feature. Text marked in italics indicates the rib-group statements from JUNOS Release 5.4 and earlier that can be omitted, whereas bold text highlights the new, simplified style of configuration.

```
[edit]
#  routing-options  { # Old method
   # rib-groups  { # Old method
      # vpna-vpnab  { # Old method
         # import-rib [VPN-A.inet.0 VPN-AB.inet.0]; # Old method
      }
      ## vpnab-vpna_and_vpnb { # Old method
         ## import-rib [VPN-AB.inet.0 VPN-A.inet.0 VPN-B.inet.0]; # Old method
      }
   }
}
routing-instances {
   VPN-A {
      instance-type vrf;
      interface fe-1/0/0.0;
      route-distinguisher 10.255.255.175:3;
      vrf-import vpna-import;
      vrf-export vpna-export;
      routing-options { # New method
         auto-export; # New method
      }
   }
   protocols {
      bgp {
         group vpna-site1 {
            ##family inet { # Old method
               ##unicast  { # Old method
```

```
                ##rib-group vpna-vpnab; # Old method
              }
            }
            peer-as 1;
            neighbor 192.255.197.141;
          }
        }
      }
    }
    VPN-AB {
      instance-type vrf;
      interface fe-1/1/0.0;
      route-distinguisher 10.255.255.175:9;
      vrf-import vpnab-import;
      vrf-export vpnab-export;
      routing-options { # New method
        auto-export; # New method
      }
    }
    protocols {
      bgp {
        group vpnab-site1 {
          ##family inet { # Old method
            ##unicast { # Old method
              ##rib-group vpnab-vpna_and_vpnb; # Old method
            }
          }
          peer-as 9;
          neighbor 192.255.197.178;
        }
      }
    }
```

In some overlapping VPN cases, additional configuration information is required:

- When vrf-import and vrf-export policies are configured on a per-instance basis, you must enable or disable the policies individually for unicast or multicast, especially when a multicast network layer reachability information (NLRI) is configured.
- When you use auto-export between overlapping VPNs and require a subset of the routes learned from an instance to be installed into the inet.0 or instance.inet.2 routing tables, you must install the routes with additional configuration statements.

To support scenarios in which not all the required information is present in the vrf-import and vrf-export policies, you can configure additional routing tables with a routing table group. For example, if you wish to add routes from VPN A and VPN AB to the inet.0 routing table, the following additional configuration parameters are required:

```
[edit]
routing-options {
  rib-groups {
    inet-access {
```

```
            import-rib inet.0;
        }
    }
}
routing-instances {
    VPN-A {
        routing-options {
            auto-export {
                family inet {
                    unicast {
                        rib-group inet-access;
                    }
                }
            }
        }
    }
    VPN-AB {
        routing-options {
            auto-export {
                family inet {
                    unicast {
                        rib-group inet-access;
                    }
                }
            }
        }
    }
}
```

There is a significant difference in how routing table groups are used in this case and how they are used more generally. Typically, routing table groups require the exporting routing table to be referenced as the primary import routing table in the rib-group configuration. In this case, the restriction is lifted and the routing table group functions as an additional list of tables that export routes.

NOTE: When upgrading to JUNOS Release 5.4 or later, be aware that route export behavior differs when using the auto-export command instead of rib-group export:

- When routes are exported between routing tables by using the rib-group statements, both primary routes (routes in the originating routing table) and secondary routes (routes imported from other routing tables) are exported to the remote PE routers. When the auto-export statement is used, only the primary routes from the originating routing table are exported.
- Routes exported from an originating VRF instance to another on the same PE now honor export policy changes to route attributes. When you use auto-export, you must add the originating route target to the exported routes. With rib-group statements, no additional configuration is necessary.

Example: Configuring Overlapping VPNs

Figure 13: Overlapping VPNs Topology Diagram

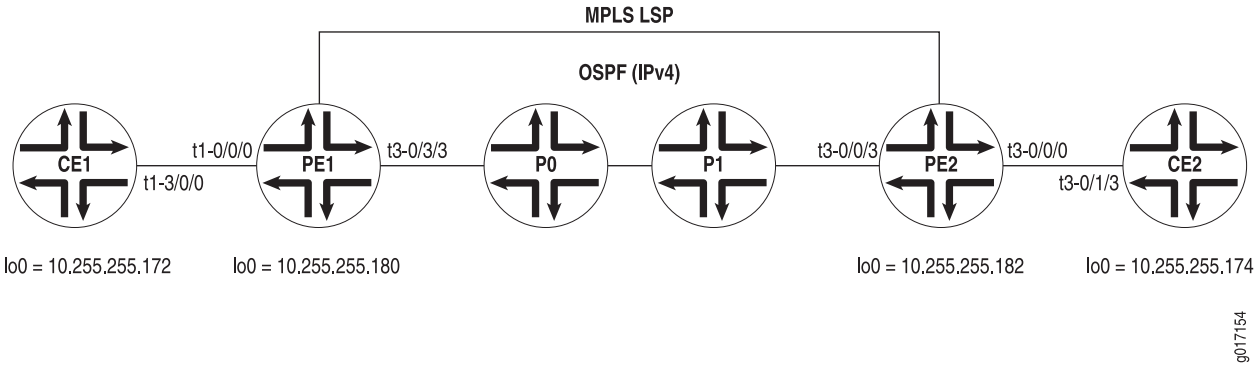


Figure 13 on page 175 shows a standard Multiprotocol Label Switching (MPLS) VPN topology. Routers PE1 and PE2 are acting as PE routers, CE1 and CE2 are CE routers, and P0 and P1 are core provider routers. You will establish three VRF instances: A, B, and AB. You will also configure auto-export as the method of sharing routing information between instances.

This example focuses on the interinstance and policy statements. As a result, some information has been omitted.

- Because PE1 uses static routing instances, the router configuration for CE1 is not included in this example.
- Most routers display a minimal configuration. Interface addresses and loopback addresses are assumed to have been enabled properly.

For more information about VPNs, see the *JUNOS VPNs Configuration Guide*.

Routers PE1 and PE2 contain the bulk of the configuration. At PE1, initiate an IBGP connection to PE2 and open a VPN connection to CE Router CE1 through three VRF instances: A, B, and AB.

The auto-export policy is applied to all instances simultaneously by means of a configuration group. Another method of enabling this option is to configure auto-export individually on each VRF instance.

Finally, the policy statements add the appropriate communities to each instance and accept traffic coming from the desired community. For example, the policy for VRF A sets community A on all outbound traffic leaving the instance, and only accepts traffic from PE2 that is tagged with community A.

Router PE1

```
[edit]
groups {
    vrf-export on {
        routing-instances {
            <*> {
                routing-options {
```

```
                auto-export;
            }
        }
    }
}
interfaces {
    t1-0/0/0
    description " to vpn02 t1-3/0/0";
    dce;
    encapsulation frame-relay;
    unit 0 {
        dlci 100;
        family inet {
            address 192.255.197.38/30;
        }
    }
    unit 1 {
        dlci 101;
        family inet {
            address 10.3.0.1/30;
        }
    }
    unit 2 {
        dlci 102;
        family inet {
            address 10.3.0.5/30;
        }
    }
}
lo0
unit 0
family inet {
    address 10.255.255.180/32;
}
protocols {
    mpls {
        interface all;
    }
    bgp {
        group pepe {
            type internal;
            neighbor 10.255.255.182 {
                family inet-vpn {
                    unicast;
                }
            }
        }
    }
    ospf {
        area 0.0.0.0 {
            interface t3-0/3/3.0;
            interface lo0.0 {
                passive;
            }
        }
```

```
    }
    ldp {
        interface all;
    }
}
policy-options {
    policy-statement A-in {
        from community A;
        then accept;
    }
    policy-statement A-out {
        then {
            community add A;
            accept;
        }
    }
    policy-statement B-in {
        from community B;
        then accept;
    }
    policy-statement B-out {
        then {
            community add B;
            accept;
        }
    }
    policy-statement AB-in {
        from community [A B];
        then accept;
    }
    policy-statement AB-out {
        then {
            community add A;
            community add B;
            accept;
        }
    }
    community A members target:69:1;
    community B members target:69:2;
}
routing-instances {
    apply-groups vrf-export-on;
    A {
        instance-type vrf;
        interface t1-0/0/0.0;
        route-distinguisher 10.255.255.180:69;
        vrf-import A-in;
        vrf-export A-out;
        routing-options {
            static {
                route 1.1.1.1/32 next-hop t1-0/0/0.0;
                route 1.1.1.2/32 next-hop t1-0/0/0.0;
            }
        }
    }
    AB {
```

```
        instance-type vrf;
        interface t1-0/0/0.2;
        route-distinguisher 10.255.255.180:69;
        vrf-import AB-in;
        vrf-export AB-out;
        routing-options {
            static {
                route 1.1.3.1/32 next-hop t1-0/0/0.2;
                route 1.1.3.2/32 next-hop t1-0/0/0.2;
            }
        }
    }
    B {
        instance-type vrf;
        interface t1-0/0/0.1;
        route-distinguisher 10.255.255.180:69;
        vrf-import B-in;
        vrf-export B-out;
        routing-options {
            static {
                route 1.1.2.1/32 next-hop t1-0/0/0.1;
                route 1.1.2.2/32 next-hop t1-0/0/0.1;
            }
        }
    }
}
```

As a provider core transit router, Router P0 only needs to provide connectivity to the PE routers. You configure OSPF, MPLS, and LDP on the interfaces pointing to both PE routers.

Router P0

```
[edit]
protocols {
    mpls {
        interface all;
    }
    ospf {
        area 0.0.0.0 {
            interface t3-0/0/3.0;
            interface t1-0/1/1.0;
        }
    }
    ldp {
        interface all;
    }
}
```

Like Router P0, Router P1 also needs to provide basic core connectivity for the PE routers. You can configure OSPF, MPLS, and LDP on the interfaces pointing toward routers P0 and PE2.

Router P1

```
[edit]
protocols {
    mpls {
        interface all;
```

```
    }
    ospf {
      area 0.0.0.0 {
        interface t1-0/1/1.0;
        interface t3-0/0/3.0;
      }
    }
    ldp {
      interface all;
    }
  }
```

At Router PE2, complete your IBGP connection to PE1 and finish the VPN connection to CE Router CE2 through VRF instance AB. The VRF import policy named AB-in is the same as the export policy used for the OSPF protocol in the AB VRF instance. The policy statements add communities A and B to all outbound routes and accept any traffic coming from these communities.

Router PE2

```
[edit]
interfaces {
  lo0
  unit 0
  family inet {
    address 10.255.255.182/32;
  }
}
protocols {
  mpls {
    interface all;
  }
  bgp {
    keep all;
    group pepe {
      type internal;
      neighbor 10.255.255.180 {
        family inet-vpn {
          unicast;
        }
      }
    }
  }
  ospf {
    area 0.0.0.0 {
      interface t3-0/0/3.0;
      interface lo0.0 {
        passive;
      }
    }
  }
  ldp {
    interface all;
  }
}
policy-options {
  policy-statement AB-in {
```

```
        from community [A B];
        then accept;
      }
      policy-statement AB-out {
        then {
          community add A;
          community add B;
          accept;
        }
      }
      community A members target:69:1;
      community B members target:69:2;
    }
    routing-instances {
      AB {
        instance-type vrf;
        interface t3-0/0/0.0;
        route-distinguisher 10.255.255.182:69;
        vrf-import AB-in;
        vrf-export AB-out;
        protocols {
          ospf {
            export AB-in;
            area 0.0.0.0 {
              interface all;
            }
          }
        }
      }
    }
```

At Router CE2, advertise the 10.255.255.174 loopback address into the VPN. Look for this route when you check the routing tables for the A, B, and AB instances on Router PE1. If the route appears in these instances, interinstance route sharing is successful.

Router CE2

```
[edit]
interfaces {
  lo0
  unit 0
  family inet {
    address 10.255.255.174/32;
  }
}
protocols {
  ospf {
    area 0.0.0.0 {
      interface t3-0/1/3.0;
      interface lo0.0;
    }
  }
}
```

Verifying Your Work

To verify that your overlapping VPN configuration is functioning properly, use the following commands:

- show route export table *table-name* (brief | detail)
- show route export instance *instance-name* (brief | detail)
- show route export vrf-target (community *community-regular-expression*) (brief | detail)

The following section shows the output of these commands as used with the configuration example.

Router PE1 Status

```
user@PE1> show route export
Table                            Export           Routes
A.inet.0                         Y                     4
AB.inet.0                        Y                     4
B.inet.0                         Y                     4

user@PE1>  show route export detail
A.inet.0                                           Routes:        4
  Flags: <vrf>
AB.inet.0                                          Routes:        4
  Flags: <vrf>
B.inet.0                                           Routes:        4
  Flags: <vrf>

user@PE1>  show route export instance detail
Instance: A                                Type: vrf
  Flags: <config> Options: <unicast multicast>
Instance: AB                               Type: vrf
  Flags: <config> Options: <unicast multicast>
Instance: B                               Type: vrf
  Flags: <config> Options: <unicast multicast>

user@PE1>  show route table A.inet.0

A.inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
1.1.1.1/32         *[Static/5] 02:08:14
                    > via t1-0/0/0.0
1.1.1.2/32         *[Static/5] 02:08:14
                    > via t1-0/0/0.0
1.1.3.1/32         *[Static/5] 02:08:14
                    > via t1-0/0/0.2
1.1.3.2/32         *[Static/5] 02:08:14
                    > via t1-0/0/0.2
10.3.0.4/30        *[Direct/0] 02:08:14
                    > via t1-0/0/0.2
10.3.0.5/32        *[Local/0] 02:08:14
                      Local via t1-0/0/0.2
10.255.255.174/32   *[BGP/170] 00:18:08, MED 2, localpref 100, from 10.255.255.182
                      AS path: I
```

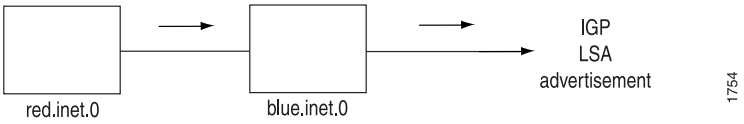
```
                    > via t3-0/3/3.0, Push 100004, Push 100017(top)
192.255.197.36/30  *[Direct/0] 02:08:14
                    > via t1-0/0/0.0
192.255.197.38/32  *[Local/0] 02:08:14
                      Local via t1-0/0/0.0
192.255.197.248/30 *[BGP/170] 00:18:18, localpref 100, from 10.255.255.182
                      AS path: I
                    > via t3-0/3/3.0, Push 100003, Push 100017(top)

user@PE1>  show route table B.inet.0

B.inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
1.1.2.1/32         *[Static/5] 02:09:28
                    > via t1-0/0/0.1
1.1.2.2/32         *[Static/5] 02:09:28
                    > via t1-0/0/0.1
1.1.3.1/32         *[Static/5] 02:09:28
                    > via t1-0/0/0.2
1.1.3.2/32         *[Static/5] 02:09:28
                    > via t1-0/0/0.2
10.3.0.0/30        *[Direct/0] 02:09:28
                    > via t1-0/0/0.1
10.3.0.1/32        *[Local/0] 02:09:28
                      Local via t1-0/0/0.1
10.3.0.4/30        *[Direct/0] 02:09:28
                    > via t1-0/0/0.2
10.3.0.5/32        *[Local/0] 02:09:28
                      Local via t1-0/0/0.2
10.255.255.174/32   *[BGP/170] 00:19:22, MED 2, localpref 100, from 10.255.255.182

                      AS path: I
                    > via t3-0/3/3.0, Push 100004, Push 100017(top)
192.255.197.248/30 *[BGP/170] 00:19:32, localpref 100, from 10.255.255.182
                      AS path: I
                    > via t3-0/3/3.0, Push 100003, Push 100017(top)

user@PE1>  show route table AB.inet.0
AB.inet.0: 14 destinations, 14 routes (14 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
1.1.1.1/32         *[Static/5] 02:09:43
                    > via t1-0/0/0.0
1.1.1.2/32         *[Static/5] 02:09:43
                    > via t1-0/0/0.0
1.1.2.1/32         *[Static/5] 02:09:43
                    > via t1-0/0/0.1
1.1.2.2/32         *[Static/5] 02:09:43
                    > via t1-0/0/0.1
1.1.3.1/32         *[Static/5] 02:09:43
                    > via t1-0/0/0.2
1.1.3.2/32         *[Static/5] 02:09:43
                    > via t1-0/0/0.2
10.3.0.0/30        *[Direct/0] 02:09:43
                    > via t1-0/0/0.1
10.3.0.1/32        *[Local/0] 02:09:43
                      Local via t1-0/0/0.1
10.3.0.4/30        *[Direct/0] 02:09:43
                    > via t1-0/0/0.2
10.3.0.5/32        *[Local/0] 02:09:43
                      Local via t1-0/0/0.2
10.255.255.174/32   *[BGP/170] 00:19:37, MED 2, localpref 100, from 10.255.255.182
```

```
                    AS path: I
                  > via t3-0/3/3.0, Push 100004, Push 100017(top)
192.255.197.36/30  *[Direct/0] 02:09:43
                    > via t1-0/0/0.0
192.255.197.38/32  *[Local/0] 02:09:43
                      Local via t1-0/0/0.0
192.255.197.248/30 *[BGP/170] 00:19:47, localpref 100, from 10.255.255.182
                      AS path: I
                    > via t3-0/3/3.0, Push 100003, Push 100017(top)

user@PE1>  show route export vrf-target detail
Target: 69:1                                inet     unicast
   Import table(s): A.inet.0 AB.inet.0
  Export table(s): A.inet.0 AB.inet.0
Target: 69:2                                inet     unicast
  Import table(s): AB.inet.0 B.inet.0
  Export table(s): AB.inet.0 B.inet.0
```

Configuring Nonforwarding Instances

In nonforwarding instances implemented in JUNOS Release 5.3 and earlier, you could configure interinstance export through use of import routing table groups. A secondary routing instance would import routes from the primary routing instance. Then, IGPs would advertise the routes received from the second instance table as shown in the example in Figure 14 on page 183.

Figure 14: Nonforwarding Instance Concept



In JUNOS Release 5.4 and later, you can use the instance-import and instance-export policy keywords to perform nonforwarding, interinstance route sharing. The keywords are assigned at the [edit routing-instances *instance-name* routing-options] hierarchy level. These statements are similar to VRF import and VRF export policies used for VRF instances.

The "rt-export" module examines the from *instance* statements present in an instance import policy to construct the list of import tables for a particular exporting instance. The following example illustrates the configuration hierarchy for this feature:

```
[edit]
policy-options {
   policy-statement {
      red-import {
         from instance blue;
         then {
            tag 1;
            accept;
         }
      }
      blue-import {
```

```
            from instance red;
            then {
               tag 2;
               accept;
            }
         }
      }
   }
   routing-instances {
      red {
         routing-options {
            instance-import red-import;
         }
      }
      blue {
         routing-options {
            instance-import blue-import;
         }
      }
   }
```

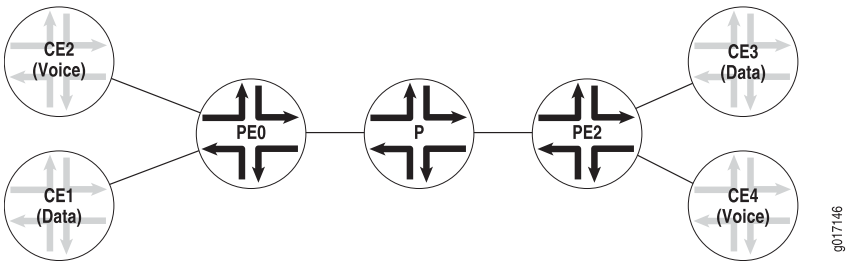
To advertise instance **blue** routes through an instance **red** IGP such as OSPF, you would add an export policy to OSPF to advertise routes from the local table.

```
[edit}
policy-options {
   policy-statement ospf-export {
      from /* some criteria */
      then accept;
   }
}
routing-instances {
   red {
      protocols ospf {
         export ospf-export;
      }
   }
}
```

When an instance import policy is configured, the policy is allowed to modify route attributes other than **next-hop**.

Example: Nonforwarding Instances Configuration

Figure 15: Nonforwarding Instances Topology Diagram



In Figure 15 on page 185, routers CE1, CE2, CE3, and CE4 are CE routers, PE0 and PE2 are PE routers, and Router P is the provider core transit router. CE1 and CE3 are part of a "community of interest" group called *data*, whereas CE2 and CE4 belong to a group called *voice*. Your goal is to connect the members of each group to each other by using a nonforwarding instance at the PE routers.

Note that routers PE0, CE1, and CE2 mirror the configurations on PE2, CE3, and CE4, respectively. Therefore, the latter routers are not shown in this example. The loopback addressing scheme for this network is shown in Table 6 on page 185.

Table 6: Nonforwarding Instances—Loopback Addresses

Router	Loopback Address
CE1	10.255.255.172
CE2	10.255.255.180
PE0	10.255.255.176
P	10.255.255.178
PE2	10.255.255.174
CE3	10.255.255.182
CE4	10.255.255.181

Routers CE1, CE2, CE3, and CE4 only need basic connectivity to their directly connected PE router. You enable OSPF on the interface that connects the CE routers to the PE routers. Since the configurations for all the CE routers are almost identical, only CE3 and CE4 are shown.

Router CE3

```
[edit]
protocols {
   ospf {
      area 0.0.0.0 {
         interface t3-0/0/0.0;
```

```
    }
  }
}
```

Router CE4

```
[edit]
protocols {
  ospf {
    area 0.0.0.0 {
      interface t3-0/0/2.0;
    }
  }
}
```

PE router configuration is next. Because the configuration for Router PE0 and Router PE2 mirror each other, only Router PE2 is displayed.

You must enable auto-export at the routing-options level for both the main configuration and the nonforwarding instances, establish policies that set tags on packets arriving from the CE routers, and accept packets into a specific instance that match the corresponding outbound tags. Specifically, you configure the router to attach a *data* tag to all packets coming from Router CE3 and a *voice* tag to all packets arriving from Router CE4. Also, forward any OSPF traffic coming from the core with a *data* tag to Router CE3, while OSPF core traffic with a *voice* tag is sent to Router CE4.

Router PE2

```
[edit]
routing-options {
  auto-export;
}
protocols {
  ospf {
    export [tag-voice tag-data];
    area 0.0.0.0 {
      interface t3-0/1/1.0;
    }
  }
}
routing-instances {
  data {
    instance-type no-forwarding;
    interface t3-0/1/3.0;
    routing-options {
      auto-export;
    }
    protocols {
      ospf {
        export import-data;
        area 0.0.0.0 {
          interface all;
        }
      }
    }
  }
  voice {
    instance-type no-forwarding;
    interface t3-0/1/0.0;
```

```
        routing-options {
            auto-export
        }
        protocols {
            ospf {
                export import-voice;
                area 0.0.0.0 {
                    interface all;
                }
            }
        }
    }
}
policy-options {
    policy-statement tag-voice {
        from instance voice;
        then {
            tag 11;
            accept;
        }
    }
    policy-statement tag-data {
        from instance data;
        then {
            tag 12;
            accept;
        }
    }
    policy-statement import-voice {
        from {
            instance master;
            protocol ospf;
            tag 11;
        }
        then accept;
    }
    policy-statement import-data {
        from {
            instance master;
            protocol ospf;
            tag 12;
        }
        then accept;
    }
}
```

On Router P, the provider core router configuration is simple. Include the interfaces that connect to the two PE routers (PE0 and PE2) in the OSPF process.

Router P

```
[edit]
protocols {
    ospf {
        area 0.0.0.0 {
            interface t1-0/1/1.0;
            interface t3-0/0/1.0;
        }
```

```
    }
  }
```

If all the configurations are correct, routers CE1 and CE3 (the routers tagged with the *data* tag) can send traffic to one another and routers CE2 and CE4 (the routers tagged with the *voice* tag) can communicate bidirectionally, but routers with different tag types cannot reach each other.

Verifying Your Work

To verify that the nonforwarding instances configuration is functioning properly, you can use the following commands:

- show ospf database
- show route detail
- ping

The following sections show the output of these commands used with the configuration example:

- Router PE2 Status on page 188
- Router CE3 Status on page 189

Router PE2 Status

```
user@PE2> show ospf database
    OSPF link state database, area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router *10.255.255.174    10.255.255.174    0x80000014   180  0x2  0x14b3  60
Router 10.255.255.176    10.255.255.176    0x80000010   592  0x2  0x14c1  60
Router 10.255.255.178    10.255.255.178    0x80000007  1074  0x2  0x9329  84
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Extern   10.255.255.172    10.255.255.176    0x8000000f   489  0x2  0xd258  36
Extern   10.255.255.180    10.255.255.176    0x8000000f   189  0x2  0x948d  36
Extern  *10.255.255.181    10.255.255.174    0x8000000f   780  0x2  0x968c  36
Extern  *10.255.255.182    10.255.255.174    0x8000000f   480  0x2  0x7aa8  36

user@PE2> show ospf database instance voice
    OSPF link state database, area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router   10.255.255.181    10.255.255.181    0x80000008  1112  0x2  0x29ac  60
Router  *192.255.197.117  192.255.197.117  0x8000000c  2681  0x2  0x5d7a  48
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Extern  *10.255.255.180    192.255.197.117  0x80000001  2681  0x2  0x5cf7  36

user@PE2> show ospf database instance data
    OSPF link state database, area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router   10.255.255.182    10.255.255.182    0x8000000b  1117  0x2  0x53d   60
Router  *192.255.197.249  192.255.197.249  0x8000000e  2686  0x2  0xbd05  48
    OSPF AS SCOPE link state database
```

```
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Extern  *10.255.255.172    192.255.197.249  0x80000002  2686  0x2  0x7d5a  36
```

Router CE3 Status

```
user@CE3> ping 10.255.255.172
PING 10.255.255.172 (10.255.255.172): 56 data bytes
64 bytes from 10.255.255.172: icmp_seq=0 ttl=252 time=2.978 ms
64 bytes from 10.255.255.172: icmp_seq=1 ttl=252 time=2.903 ms
^C
--- 10.255.255.172 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 2.903/2.941/2.978/0.037 ms

user@CE3> ping 10.255.255.180
PING 10.255.255.180 (10.255.255.180): 56 data bytes
^C
--- 10.255.255.180 ping statistics ---
2 packets transmitted, 0 packets received, 100% packet loss

user@CE3> show ospf database
    OSPF link state database, area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router  *10.255.255.182    10.255.255.182    0x8000000b  1164  0x2  0x53d   60
Router   192.255.197.249  192.255.197.249  0x8000000e  2735  0x2  0xbd05  48
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Extern   10.255.255.172    192.255.197.249  0x80000002  2735  0x2  0x7d5a  36

user@CE3> show route 10.255.255.172 detail

inet.0: 31 destinations, 32 routes (30 active, 0 holddown, 1 hidden)
10.255.255.172/32 (1 entry, 1 announced)
        *OSPF   Preference: 150
                Next hop: via t3-0/0/0.0, selected
                State: <Active Int Ext>
                Local AS:    69
                Age: 47:23      Metric: 2       Tag: 12
                Task: OSPF
                Announcement bits (1): 0-KRT
                AS path: I
```

For More Information

For additional information about interinstance route sharing, see the following resources:

- *JUNOS VPNs Configuration Guide*
- *JUNOS Routing Protocols Configuration Guide*

Revision History

5 October 2009—10.0R1 Release. Roy Spencer.

13 June 2009—9.6R1 Release. Roy Spencer.

13 April 2009—9.5R1 Release. Roy Spencer.

15 January 2009—9.4R1 Release. Roy Spencer.

10 October 2008—9.3R1 Release. Roy Spencer.

10 July 2008—9.2R1 Release. Roy Spencer.

10 April 2008—9.1R1 Release. Roy Spencer.

1 February 2008—9.0R1 Release. Fawn Damitio.

29 June 2007—8.4R1 Release. Fawn Damitio.

27 March 2007—8.3R1 Release. Fawn Damitio.

12 January 2007—Added support for MX960 Ethernet Services Routers. 8.2R1 Release. Fawn Damitio.

15 September 2006—8.1R1 Release. Richard Hendricks.

29 June 2006—8.0R1 Release. Richard Hendricks.

27 March 2006—7.6R1 Release. Richard Hendricks.

9 January 2006—7.5R1 Release. Richard Hendricks.

14 September 2005—7.4R1 Release. Richard Hendricks.

13 June 2005—7.3R1 Release. Richard Hendricks.

5 April 2005—7.2R1 Release. Richard Hendricks.

2 February 2005—7.1R1 Release. Richard Hendricks.

6 October 2004—7.0R1 Release. Richard Hendricks.

6 July 2004—6.4R1 Release. Richard Hendricks.

5 April 2004—6.3R1 Release. Richard Hendricks.

22 December 2003—6.2R1 Release. Richard Hendricks.

22 September 2003—6.1R1 Release. Richard Hendricks.

30 June 2003—6.0R1 Release. Richard Hendricks.

2 April 2003—5.7R1 Release. Richard Hendricks.

27 December 2002—5.6R1 Release. Richard Hendricks.

30 September 2002—5.5R1 Release. Richard Hendricks.

19 July 2002—5.4R1 Release. Richard Hendricks.

6 May 2002—Initial document written. Richard Hendricks.

Part 2

Routing Protocols

- Logical Systems on page 195
- OSPF Version 3 for IPv6 on page 239
- Multitopology Routing on page 271

Chapter 7

Logical Systems

This chapter covers the following topics:

- Logical Systems Overview on page 195
- Logical Systems System Requirements on page 199
- Logical Systems Terms and Acronyms on page 199
- Configuring Logical Systems on page 200
- Configuring Logical System Administrators (Master Administrator) on page 200
- Configuring Logical System Interface Properties (Master Administrator) on page 201
- Assigning Logical Interfaces to the Logical System (Master or Logical System Administrator) on page 201
- Configuring Protocols, Routing, and Policy Statements for the Logical System (Master or Logical System Administrator) on page 201
- Configuring Other Logical System Statements on page 202
- Example: Configuring Logical Systems on page 205
- For More Information on page 237
- Revision History on page 237

Logical Systems Overview

For many years, engineers have combined power supplies, routing hardware and software, forwarding hardware and software, and physical interfaces into a networking device known as a router. Networking vendors have created large routers and small routers, but all routers have been placed into service as individual devices. As a result, the router has been considered a single physical device for most of its history.

The concept of logical systems breaks with this tradition. With JUNOS Software, you can partition a single router into multiple logical devices that perform independent routing tasks. Because logical systems perform a subset of the tasks once handled by the main router, logical systems offer an effective way to maximize the use of a single routing or switching platform.

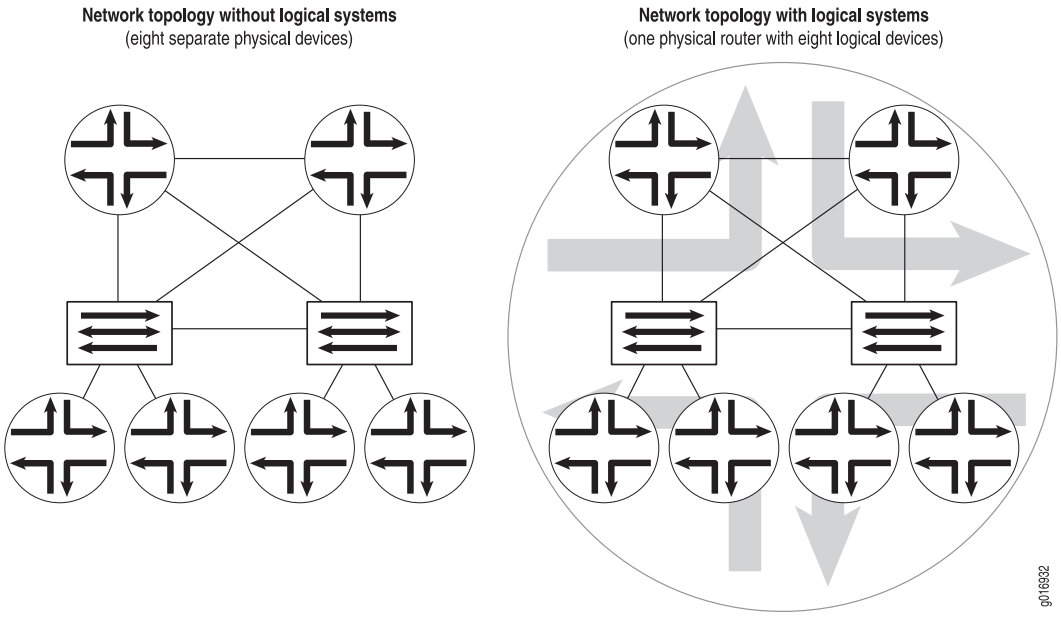
NOTE: Beginning with JUNOS Software Release 9.3, the logical router feature has been renamed logical system.

All configuration statements, operational commands, show command outputs, error messages, log messages, and SNMP MIB objects that contain the string logical-router or logical-routers have been changed to logical-system and logical-systems, respectively.

Traditionally, service provider network design requires multiple layers of switches and routers. These devices transport packet traffic between customers. As seen on the left side of Figure 16 on page 196, access devices are connected to edge devices, which are in turn connected to core devices.

However, this complexity can lead to challenges in maintenance, configuration, and operation. To reduce such complexity, Juniper Networks support logical systems. Logical systems perform a subset of the actions of the main router and have their own unique routing tables, interfaces, policies, and routing instances. As shown on the right side of Figure 16 on page 196, a set of logical systems within a single router can handle the functions previously performed by several small routers.

Figure 16: Logical Systems Concept



The following protocols and functions are supported on logical systems:

- Open Shortest Path First (OSPF), Intermediate System-to-Intermediate System (IS-IS), Routing Information Protocol (RIP), RIP next generation (RIPng), Border Gateway Protocol (BGP), Resource Reservation Protocol (RSVP), Label Distribution Protocol (LDP), static routes, and Internet Protocol version 4 (IPv4) and version 6 (IPv6) are supported at the [edit logical-systems *logical-system-name* protocols] hierarchy level.
- Multiprotocol Label Switching (MPLS) provider edge (PE) and core provider router functions, such as Layer 2 virtual private networks (VPNs), Layer 3 VPNs, circuit cross-connect (CCC), Layer 2 circuits, and virtual private LAN service (VPLS) are supported at the [edit logical-systems *logical-system-name* routing-instances] hierarchy level.
- Multicast protocols, such as Protocol Independent Multicast (PIM) and Distance Vector Multicast Routing Protocol (DVMRP), are supported at the [edit logical-systems *logical-system-name* protocols] hierarchy level. Rendezvous point (RP) and source designated router (DR) functionality for multicast protocols within a logical system are also supported.
- All policy-related statements available at the [edit policy-options] hierarchy level are supported at the [edit logical-systems policy-options] hierarchy level.
- Most routing options statements available at the [edit routing-options] hierarchy level are supported at the [edit logical-systems routing-options] hierarchy level.
- Graceful Routing Engine switchover (GRES) is supported on Juniper Networks M Series Multiservice Edge Routers, Juniper Networks MX Series Ethernet Services Routers, and Juniper Networks T Series Core Routers. For more information about GRES, see the *JUNOS High Availability Configuration Guide*.
- You can assign most interface types to a logical system, including SONET/SDH interfaces, Ethernet interfaces, Asynchronous Transfer Mode (ATM) interfaces, ATM2 intelligent queuing (IQ) interfaces, channelized IQ and Gigabit Ethernet IQ interfaces, aggregated interfaces, Link Services interfaces, and Multilink Services interfaces.
- Source class usage, destination class usage, unicast reverse-path forwarding, class of service, firewall filters, class-based forwarding, and policy-based accounting work with logical systems when you configure these features on the main router.
- Simple Network Management Protocol (SNMP) has been extended to support logical systems and routing instances. A network management system receives instance-aware information in the following format:

  logical-system-name/routing-instance@community

  As a result, a network manager can gather statistics for a specific community within a routing instance within a logical system. For more information on SNMP for logical systems, see the *JUNOS Network Management Configuration Guide*.
- SNMP support for logical systems and routing instances has been enhanced. The SNMP manager for a routing instance can now request and manage SNMP data only for that routing instance and other routing instances in the same logical system. As in previous releases, by default the SNMP manager for the default routing instance in the main router (inet.0) can access SNMP data from all routing

instances. To restrict that manager's access to the default routing instance only, include the routing-instance-access statement at the [edit snmp] hierarchy level. For more information, see the *JUNOS Network Management Configuration Guide*.

The following restrictions apply to logical systems:

- You can configure a maximum of 15 logical systems plus the master logical system on a router. When a configuration session is in use, users who are tied to the same logical system cannot commit configuration changes.
- The router has only one running configuration database, which contains configuration information for the main router and all associated logical systems. When configuring a logical system, users have their own candidate configuration database, which does not become part of the running configuration database until the user issues the commit statement.
- If a logical system experiences an interruption of its routing protocol process (rpd), the core dump output is placed in a file in the following location: /var/tmp/rpd_*logical-system-name*.core-tarball.*number*.tgz. Likewise, if you issue the restart routing command in a logical system, only the routing protocol process (rpd) for the logical system is restarted.
- If you configure trace options for a logical system, the output log file is stored in the following location: /var/log/*logical-system-name*. To monitor a log file within a logical system, issue the monitor start *logical-system-name/filename* command.
- The following Physical Interface Cards (PICs) are not supported with logical systems: Adaptive Services, ES, Monitoring Services, and Monitoring Services II.
- Generalized MPLS (GMPLS), IP Security (IPsec), point-to-multipoint label-switched paths (LSPs), port mirroring, and sampling are not supported.
- LSP ping and traceroute for autonomous system (AS) number lookup are not supported.
- Class of service (CoS) on logical tunnel (lt) or virtual loopback tunnel (vt) interfaces in a logical system is not supported.
- You cannot include the vrf-table-label statement on multiple logical systems if the core-facing interfaces are channelized or configured with multiple logical interfaces (Frame Relay DLCIs or Ethernet VLANs).
- The master administrator must configure global interface properties and physical interface properties at the [edit interfaces] hierarchy level. Logical system administrators can only configure and verify configurations for the logical systems to which they are assigned. For more information on configuring interfaces, see the *JUNOS Network Interfaces Configuration Guide*.

NOTE: A virtual router does not have the same capabilities as a logical system. A virtual router is a type of simplified routing instance that has a single routing table. A logical system is a partition of the main router and can contain multiple routing instances and routing tables. For example, a logical system can contain multiple virtual router routing instances. As a result, these two entities are not equivalent.

Logical Systems System Requirements

To implement logical systems, your system must meet these minimum requirements:

- JUNOS Release 8.5 or later for logical system administrator support.
- JUNOS Release 8.4 or later for SNMP enhancements and limits.
- JUNOS Release 8.3 or later for Bidirectional Forwarding Detection (BFD) on logical systems.
- JUNOS Release 8.2 or later for support on MX Series routers
- JUNOS Release 7.5 or later for SNMP support within a logical system
- JUNOS Release 7.4 or later for multicast protocol RP and source DR functionality within a logical system
- JUNOS Release 7.0 or later to implement a logical tunnel (lt) interface on an integrated Adaptive Services Module in an M7i router
- JUNOS Release 6.1 or later, a Tunnel Services PIC, and an Enhanced FPC on M Series or T Series routers to implement a logical tunnel (lt) interface
- JUNOS Release 6.0 or later for basic logical system functionality
- One or more M Series, MX Series, or T Series routers
- On M Series and T Series routers, a variety of PICs to assign interfaces to each logical system

Logical Systems Terms and Acronyms

L

logical system Segmentation of a system into multiple logical devices. Logical system configuration statements are found at the [edit logical-systems] hierarchy level.

logical system administrator A user account with configuration and verification privileges for only the logical systems to which that user is assigned.

M

main router The standard concept of a router. Main router configuration statements are found at the [edit] hierarchy level.

master administrator A user account with superuser configuration and verification privileges.

Configuring Logical Systems

To implement logical systems, you must configure the following:

- Configuring Logical System Administrators (Master Administrator) on page 200
- Configuring Logical System Interface Properties (Master Administrator) on page 201
- Assigning Logical Interfaces to the Logical System (Master or Logical System Administrator) on page 201
- Configuring Protocols, Routing, and Policy Statements for the Logical System (Master or Logical System Administrator) on page 201
- Configuring Other Logical System Statements on page 202

Configuring Logical System Administrators (Master Administrator)

The master administrator can assign one or more logical system administrators to each logical system. Logical system administrators are confined to the context of the logical system to which they are assigned. This means that any global configuration statements are restricted from them. This also means that command output is restricted to the context to which the logical system administrators are assigned.

To configure logical system administrators, include the **logical-system** *logical-system-name* statement at the [**edit system login class** *class-name*] hierarchy level:

```
[edit]
system {
   login {
      class admin1 {
         permissions all;
         logical-system ls1;
      }
      class admin2 {
         permissions view; # Gives users assigned to class admin2 the ability to view
         logical-system ls2; # but not to change the configuration.
      }
      user user1 {
         class admin1;
      }
      user user2 {
         class admin2;
      }
   }
}
```

Configuring Logical System Interface Properties (Master Administrator)

Before adding interfaces to a logical system, the master administrator must configure physical interface properties on the main router. Common physical interface properties include encapsulation types and interface-related options.

To configure physical interface properties, the master administrator must include the statements at the [edit interfaces *interface-name*] hierarchy level.

Assigning Logical Interfaces to the Logical System (Master or Logical System Administrator)

After the interfaces are configured, the master administrator can assign logical interfaces to a logical system. To configure, include the unit statement at the [edit logical-systems *logical-system-name* interfaces *interface-name*] hierarchy level. Once you assign logical interfaces to a logical system, they are considered part of the logical system. Any logical interface can only be assigned one logical system, including the main router.

```
[edit]
logical-systems logical-system-name {
    interfaces {
        interface-name {
            logical-interface-statements;
            unit unit-number {
                family inet {
                    address ip-address;
                }
            }
        }
    }
    interfaces {
        interface-name {
            physical-interface-statements;
        }
    }
}
```

Configuring Protocols, Routing, and Policy Statements for the Logical System (Master or Logical System Administrator)

You can configure routing protocols (such as OSPF, BGP, and MPLS), policies (such as next hop or load balancing), routing options, and routing instances for a logical system.

To configure routing protocols, include the protocols statement at the [edit logical-systems *logical-system-name*] hierarchy level. To configure policies, include the policy-options statement at the [edit logical-systems *logical-system-name*] hierarchy level. To configure routing options, include the routing-options statement at the [edit logical-systems *logical-system-name*] hierarchy level. To configure routing instances,

include the routing-instances statement at the [edit logical-systems *logical-system-name*] hierarchy level.

```
[edit]
logical-systems logical-system-name {
    protocols {
        ...
    }
    policy-options {
        ...
    }
    routing-options {
        ...
    }
    routing-instances {
        ...
    }
}
```

Configuring Other Logical System Statements

You can configure a variety of additional statements in conjunction with a logical system:

- Logical tunnel (lt) interface—You can connect different logical systems together within the same router with an lt interface. On M Series and T Series routers, you can create an lt interface if you have a Tunnel Services PIC installed on an Enhanced FPC in your routing platform. On an M7i router, logical tunnel interfaces can be created by using the integrated Adaptive Services Module. On an MX Series router, the master administrator must configure logical tunnel interfaces by including the tunnel-services statement at the [edit chassis fpc *slot-number* pic *number*] hierarchy level. For more information about configuring tunnel interfaces on MX Series routers, see the *JUNOS System Basics Configuration Guide*.

  You must treat each interface like a point-to-point connection because you can only connect one logical tunnel interface to another at any given time. Also, you must select an interface encapsulation type, specify a DLCI number or VLAN identifier, configure a corresponding protocol family, and set the logical interface unit number of the peering lt interface. To configure the interface encapsulation type, include the dlci, encapsulation, family, peer-unit,and vlan-id statements at the following hierarchy levels:

  - M Series, MX Series, or T Series router (master administrator only)—[edit interfaces lt-*fpc/pic/*0 unit *unit-number*]
  - Logical system—[edit logical-systems *logical-system-name* interfaces *lt-fpc/pic/*0 unit *unit-number*]

    ```
    [edit]
    logical-systems logical-system-name {
        interfaces {
            lt-fpc/pic/0 {
                unit unit-number {
                    encapsulation (ethernet | ethernet-ccc | ethernet-vpls | frame-relay
                        | fame-relay-ccc | vlan | vlan-ccc | vlan-vpls);
    ```

```
                peer-unit number; # The logical unit number of the peering lt
                  interface.
                dlci dlci-number;
                vlan-id vlan-number;
                family (ccc | inet | inet6 | iso | mpls | tcc);
            }
        }
    }
}
```

NOTE: When you configure IPv6 addresses on a logical tunnel interface, you must configure unique IPv6 link local addresses for any logical interfaces that peer with one another. To configure a link local address, you must be the master administrator. Include a second IPv6 address with the address statement at the [edit interfaces lt-*fpc/pic/port* unit *unit-number* family inet6] hierarchy level. Link local addresses typically begin with the numbers fe80 (such as fe80::1111:1/64).

- Dynamic Host Control Protocol (DHCP) relay (master administrator only)—In a logical system, you can configure a DHCP or BOOTP server, and allow TFTP and DNS packets to be forwarded. To configure a DHCP or BOOTP server in a logical system, include the logical-system statement at the [edit forwarding-options helpers bootp interface *interface-name* server *ip-address*] hierarchy level. To configure TFTP packet forwarding in a logical system, include the logical-system statement at the [edit forwarding-options helpers tftp interface *interface-name* server *ip-address*] hierarchy level. To configure DNS packet forwarding in a logical system, include the logical-system statement at the [edit forwarding-options helpers domain interface *interface-name* server *ip-address*] hierarchy level. For more information about DHCP relay, BOOTP, TFTP, or DNS, see the *JUNOS Policy Framework Configuration Guide*.
- Filter-based forwarding (master administrator only)—You can configure filter-based forwarding for a logical system or a routing instance within a logical system. To configure filter-based forwarding for a logical system, include the logical-system statement at the [edit firewall filter *filter-name* term *term-name* then] hierarchy level. To configure filter-based forwarding for a routing instance within a logical system, include the routing-instance option at the [edit firewall filter *filter-name* term *term-name* then logical-system *logical-system-name*] hierarchy level. For more information, see the *JUNOS Policy Framework Configuration Guide*.
- Bidirectional forwarding—You can configure Bidirectional Forwarding Detection Protocol (BFD) for a logical system or a routing instance within a logical system. To configure BFD for a logical system, include the bfd-liveness-detection statement at the [edit logical-systems *logical-system-name* protocols] hierarchy level. To configure BFD for a routing instance within a logical system, include the bfd-liveness-detection statement at the [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols] hierarchy level. This feature is supported for the following protocols: RIP, BGP, OSPF, and IS-IS. For more information, see the *JUNOS Routing Protocols Configuration Guide*.
- You can place yourself into the context of a specific logical system. To configure a logical system context, issue the set cli logical-system *logical-system-name* command.

When you enter logical system context mode and enter an operational mode command, the output of the command displays information related to the logical system only. For example, when you issue the show route command, the output shows only the routes that are assigned to the logical system.

```
user@P0> set cli logical-system ls1
Logical system: ls1
user@P0:ls1># Note that the user is now restricted to a logical system context.
```

To clear the logical system context and return to a full router (master router) context, issue the clear cli logical-system command.

```
user@P0:ls1> clear cli logical-system
Cleared default logical system
user@P0># Note that the user can now view the entire router again.
```

To achieve the same effect when using a JUNOScript client application, include the < set-logical-system > tag:

```
<rpc>
<set-logical-system>
<logical-system>ls1</logical-system>
</set-logical-system>
</rpc>
```

For more information about JUNOScript, see the *JUNOScript API Guide*.

- Enhanced SNMP support for logical systems (master administrator only)—By default, the SNMP manager can access SNMP data from all routing instances. To restrict the SNMP manager to data from the default routing instance only, include the routing-instance-access statement at the [edit snmp] hierarchy level. For more information about configuring SNMP, see the *JUNOS Network Management Configuration Guide*.
- SNMP community strings for routing instances and logical systems—To specify a routing instance when you add a client to an SNMP community, include the routing-instance *routing-instance-name* statement at the [edit snmp community *community-name*] hierarchy level. To specify a routing instance that is defined within a logical system, include the routing-instance routing-instance-name statement at the [edit snmp community community-name logical-system logical-system-name] hierarchy level.
- SNMPv3 trap targets for routing instances and logical systems—To specify a routing instance when you add a client to an SNMPv3 trap target, include the routing-instance *routing-instance-name* statement at the [edit snmp v3 target-address *target-address*] hierarchy level. To specify a logical system when you add a client in an SNMPv3 trap target, include the logical-system *logical-system-name* statement at the [edit snmp v3 target-address *target-address*] hierarchy level. For more information about configuring SNMP, see the *JUNOS Network Management Configuration Guide*.
- SNMP trap packets for routing instances—To specify a routing instance for SNMP trap packets sent by the router, include the routing-instance *routing-instance-name* statement at the [edit snmp trap-options] hierarchy level. To specify a routing instance that is defined within a logical system, include the routing-instance *routing-instance-name* statement at the [edit snmp trap-options logical-system

logical-system-name] hierarchy level. For more information about configuring SNMP, see the *JUNOS Network Management Configuration Guide*.

- SNMP trap groups for routing instances and logical systems—To specify a routing instance of an SNMP trap group, include the routing-instance *routing-instance-name* statement at the [edit snmp trap-group *trap-group-name*] hierarchy level. To specify a logical system for an SNMP trap group, include the logical-system *logical-system-name* statement at the [edit snmp trap-group *trap-group-name*] hierarchy level. For more information about configuring SNMP, see the *JUNOS Network Management Configuration Guide*.

In addition, you can configure only Frame Relay interface encapsulation on a logical tunnel interface when it is configured with an IPv6 address.

Example: Configuring Logical Systems

Figure 17: Logical System Topology Diagram

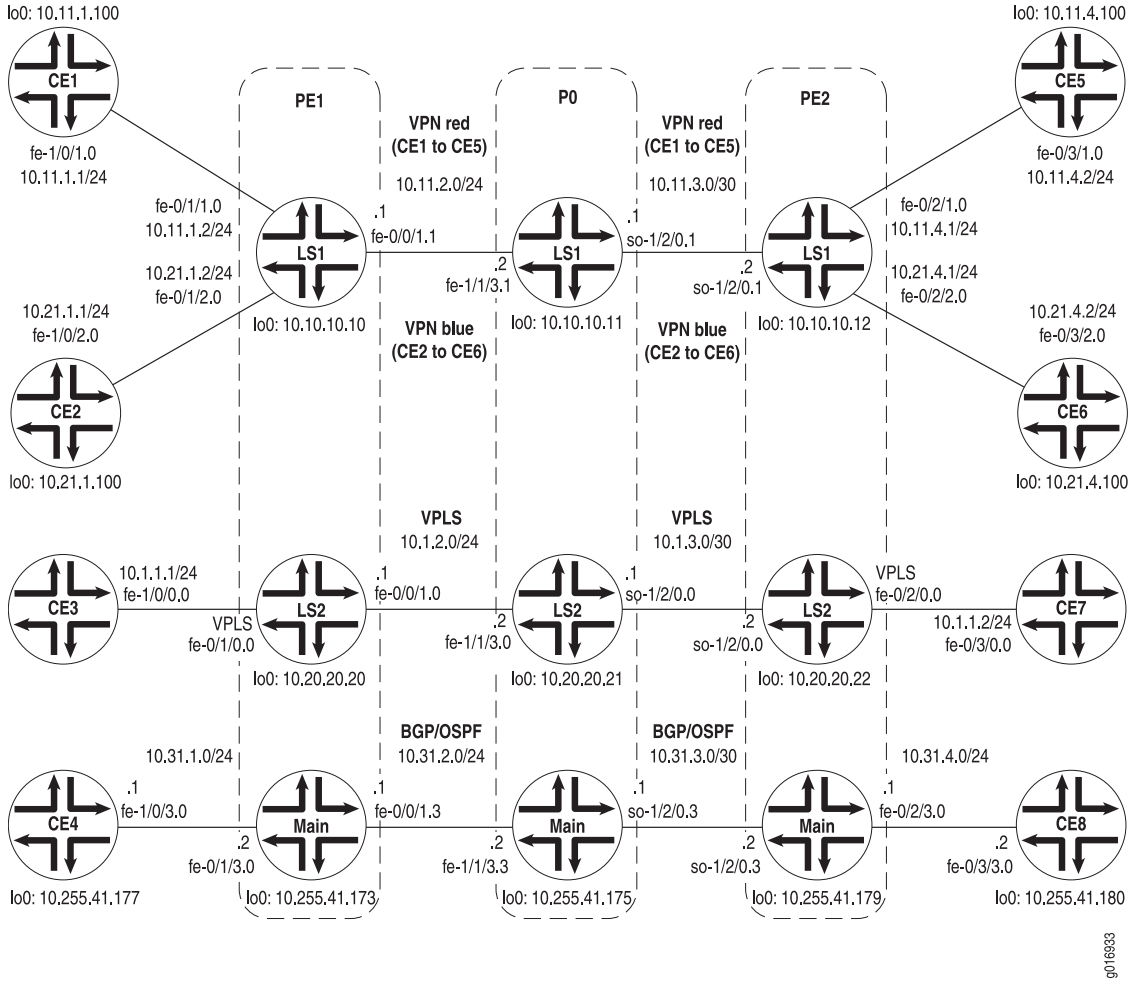


Figure 17 on page 205 shows four pairs of customer edge (CE) routers that are connected across an MPLS backbone. Routers CE1 and CE5 are part of the red VPN, routers CE2 and CE6 are in the blue VPN, routers CE3 and CE7 belong to a VPLS domain, and routers CE4 and CE8 are connected with standard protocols. Two logical systems are configured on provider edge (PE) routers PE1 and PE2 and provider core Router P0. Each of these three routers has two logical systems: LS1 and LS2. To illustrate the concept of a logical system, both VPNs are part of logical system LS1, the VPLS instance belongs to LS2, and the remaining routers use the main router portion of routers PE1, P0, and PE2.

On Router CE1, configure OSPF to connect to the red VPN in logical system LS1 on Router PE1:

Router CE1

```
[edit]
interfaces {
    fe-1/0/1 {
        vlan-tagging;
        unit 0 {
            description "routing-instance red CE";
            vlan-id 101;
            family inet {
                address 10.11.1.1/24;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.11.1.100/32;
            }
        }
    }
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface fe-1/0/1.0;
            interface lo0.0;
        }
    }
}
```

On Router CE2, configure BGP to connect to the blue VPN in logical system LS1 on Router PE1:

Router CE2

```
[edit]
interfaces {
    fe-1/0/2 {
        vlan-tagging;
        unit 0 {
            description "routing-instance blue CE";
            vlan-id 102;
            family inet {
                address 10.21.1.1/24;
            }
        }
```

```
    }
    lo0 {
      unit 0 {
        family inet {
          address 10.21.1.100/32;
        }
      }
    }
  }
  routing-options {
    autonomous-system 200;
  }
  protocols {
    bgp {
      export export_loopback;
      group to_PE {
        type external;
        local-address 10.21.1.1;
        peer-as 100;
        neighbor 10.21.1.2;
      }
    }
  }
  policy-options {
    policy-statement export_loopback {
      from {
        route-filter 10.21.1.100/32 exact;
      }
      then accept;
    }
  }
```

On Router CE3, configure the Fast Ethernet interface in VLAN 600 to connect with the VPLS routing instance in logical system LS2 on Router PE1:

Router CE3

```
[edit]
interfaces {
  fe-1/0/0 {
    vlan-tagging;
    unit 0 {
      description "vpls interface";
      vlan-id 600;
      family inet {
        address 10.1.1.1/24;
      }
    }
  }
}
```

On Router CE4, configure the Fast Ethernet interface to connect with the main router at Router PE1:

Router CE4

```
[edit]
interfaces {
  fe-1/0/3 {
    vlan-tagging;
```

```
        unit 0 {
            description "main router interface";
            vlan-id 103;
            family inet {
                address 10.31.1.1/24;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.41.177/32;
            }
        }
    }
}
```

On Router PE1, create two VPN routing and forwarding (VRF) routing instances in logical system LS1: red and blue. Configure the CE-facing logical interfaces so that traffic from Router CE1 is placed in the red VPN and traffic from Router CE2 is placed in the blue VPN. Next, create a logical interface at fe-0/0/1.1 to connect to logical system LS1 on Router P0.

Also on Router PE1, create a VPLS routing instance in logical system LS2. Configure a logical interface so that traffic from Router CE3 is sent into the VPLS domain and connects to logical system LS2 on Router P0.

Create a logical system administrator for LS1. The logical system administrator can be responsible for the maintenance of this logical system.

Finally, configure a logical interface to interconnect Router CE4 with the main router portion of Router P0.

Router PE1

```
[edit]
logical-systems {
    ls1 { # The configuration for the first logical system begins here.
        interfaces {
            fe-0/0/1 {
                unit 1 {  # This is the core-facing interface for logical system LS1.
                    description "ls1 interface";
                    vlan-id 101;
                    family inet {
                        address 10.11.2.1/24;
                    }
                    family iso;
                    family mpls;
                }
            }
            fe-0/1/1 {
                unit 0 {   # This logical interface connects to Router CE1.
                    description "routing-instance red interface";
                    vlan-id 101;
                    family inet {
                        address 10.11.1.2/24;
                    }
```

```
    }
  }
  fe-0/1/2 {
    unit 0 {   # This logical interface connects to Router CE2.
      description "routing-instance blue interface";
      vlan-id 102;
      family inet {
        address 10.21.1.2/24;
      }
    }
  }
  lo0 {
    unit 1 {
      description "ls1 loopback";
      family inet {
        address 10.10.10.10/32;
      }
      family iso {
        address 47.1111.1111.1111.1111.00;
      }
    }
  }
}
protocols { # You configure RSVP, MPLS, IS-IS, and BGP for logical system LS1.
  rsvp {
    interface all;
  }
  mpls  {
    label-switched-path to_10.10.10.12 {
      to 10.10.10.12;
    }
    interface all;
  }
  bgp {
    group to_other_PE {
      type internal;
      local-address 10.10.10.10;
      family inet-vpn {
        any;
      }
      neighbor 10.10.10.12;
    }
  }
  isis {
    interface all;
  }
}
policy-options {
  policy-statement from_bgp_to_ospf {
    then accept;
  }
}
routing-instances {
  blue {
    instance-type vrf; # You configure instance blue within logical system LS1.
    interface fe-0/1/2.0;
```

```
            route-distinguisher 10.10.10.10:200;
            vrf-target target:20:20;
            protocols {
               bgp {   #BGP connects the blue instance with Router CE2.
                  group to_CE {
                     type external;
                     local-address 10.21.1.2;
                     peer-as 200;
                     neighbor 10.21.1.1;
                  }
               }
            }
         }
         red {
            instance-type vrf; # You configure instance red within logical system LS1.
            interface fe-0/1/1.0;
            route-distinguisher 10.10.10.10:100;
            vrf-target target:10:10;
            protocols {
               ospf {#OSPF connects the red instance with Router CE1.
                  export from_bgp_to_ospf;
                  area 0.0.0.0 {
                     interface all;
                  }
               }
            }
         }
      }
      routing-options {
         autonomous-system 100;
      }
   }
   ls2 {  # The configuration for the second logical system begins here.
      interfaces {
         fe-0/0/1 {
            unit 0 {  # This is the core-facing interface for logical system LS2.
               description "ls2 interface";
               vlan-id 100;
               family inet {
                  address 10.1.2.1/24;
               }
               family iso;
               family mpls;
            }
         }
         fe-0/1/0 {
            unit 0 {  # This logical interface connects to Router CE3.
               description "vpls interface";
               encapsulation vlan-vpls;
               vlan-id 600;
               family vpls;
            }
         }
         lo0 {
            unit 2 {
               description "ls2 loopback";
```

```
            family inet {
               address 10.20.20.20/32;
            }
            family iso {
               address 47.2222.2222.2222.2222.00;
            }
         }
      }
      protocols { # You configure RSVP, MPLS, IS-IS, and BGP for logical system LS2.
         rsvp {
            interface all;
         }
         mpls {
            label-switched-path to_10.20.20.22 {
               to 10.20.20.22;
            }
            interface all;
         }
         bgp {
            group to_VPLS_PE {
               type internal;
               local-address 10.20.20.20;
               family l2vpn {
                  signaling;
               }
               neighbor 10.20.20.22;
            }
         }
         isis {
            interface fe-0/0/1.0;
            interface lo0.2;
         }
      }
      routing-instances {
         new {
            instance-type vpls; # You configure VPLS within logical system LS2.
            interface fe-0/1/0.0;
            route-distinguisher 10.20.20.20:100;
            vrf-target target:30:30;
            protocols {
               vpls {
                  site-range 10;
                  site newPE {
                     site-identifier 1;
                  }
               }
            }
         }
      }
      routing-options {
         autonomous-system 400;
      }
   }
}
interfaces {
```

```
    fe-0/0/1 {
        vlan-tagging;
        unit 3 { # This is the core-facing interface for the main router of PE1.
            description "main router to P0";
            vlan-id 103;
            family inet {
                address 10.31.2.1/24;
            }
            family iso;
            family mpls;
        }
    }
    fe-0/1/3 {
        vlan-tagging;
        unit 0 { # This logical interface in the main router of PE1 connects to CE4.
            description "main router to CE4";
            vlan-id 103;
            family inet {
                address 10.31.1.2/24;
            }
        }
    }
    fe-0/1/0 {  # You must always configure physical interface statements for
        vlan-tagging;  # logical system interfaces at the [edit interfaces] hierarchy level.
        encapsulation vlan-vpls;
    }
    fe-0/1/1 {
        vlan-tagging;
    }
    fe-0/1/2 {
        vlan-tagging;
    }
    lo0 {
        unit 0 {
            description "main router loopback";
            family inet {
                address 10.255.41.173/32;
            }
        }
    }
}
routing-options {
    static {
        route 10.255.41.177/32 next-hop 10.31.1.1;
    }
    autonomous-system 500;
}
protocols {
    bgp {  # The main router uses BGP as the exterior gateway protocol.
        group to_main_ls {
            type internal;
            local-address 10.255.41.173;
            export export_address;
            neighbor 10.255.41.179;
            neighbor 10.255.41.175;
        }
```

```
    }
    ospf {  # The main router uses OSPF as the interior gateway protocol.
      area 0.0.0.0 {
        interface lo0.0;
        interface fe-0/0/1.3;
      }
    }
  }
  policy-options {
    policy-statement export_address {
      from {
        route-filter 10.255.41.177/32 exact;
      }
      then accept;
    }
  }
  system {
    login {
      class ls1–admin {
        permissions all;
        logical-system ls1;
      }
      user ls1–admin {
        class ls1–admin;
        authentication plain-text password;
        New password: password
        Retype new password: password
      }
    }
  }
```

On Router P0, configure logical systems LS1, LS2, and the main router. For the logical system, you must configure physical interface properties at the main router [edit interfaces] hierarchy level and assign the logical interfaces to the logical systems. Next, you must configure protocols (such as RSVP, MPLS, BGP, and IS-IS), routing options, and policy options for the logical systems. Last, configure the same logical system administrator for LS1 that you configured on Router PE1. Configure this same logical system administrator for LS2 to have permission to view the LS2 configuration, but not change the configuration for LS2.

In this example, logical system LS1 transports traffic for the red VPN that exists between routers CE1 and CE5. LS1 also connects the blue VPN that exists between routers CE2 and CE6. Logical system LS2 transports VPLS traffic between routers CE3 and CE7.

For the main router on Router P0, you can configure the router as usual. In this example, the main router transports traffic between routers CE4 and CE8. As a result, configure the interfaces and routing protocols (OSPF, BGP) to connect to the main router portion of routers PE1 and PE2.

Router P0

```
[edit]
logical-systems {
  ls1 { # The configuration for the first logical system begins here.
    interfaces {
      fe-1/1/3 {
```

```
            unit 1 {  # This logical interface connects to LS1 on Router PE1.
                description "ls1 interface";
                vlan-id 101;
                family inet {
                    address 10.11.2.2/24;
                }
                family iso;
                family mpls;
            }
        }
        so-1/2/0 {
            unit 1 { # This logical interface connects to LS1 on Router PE2.
                description "ls1 interface";
                dlci 101;
                family inet {
                    address 10.11.3.1/24;
                }
                family iso;
                family mpls;
            }
        }
        lo0 {
            unit 1 {
                description "ls1 loopback";
                family inet {
                    address 10.10.10.11/32;
                }
                family iso {
                    address 47.1111.1111.1111.1112.00;
                }
            }
        }
    }
    protocols { # You configure RSVP, MPLS, and IS-IS for logical system LS1.
        rsvp {
            interface all;
        }
        mpls {
            interface all;
        }
        isis {
            interface all;
        }
    }
}
ls2 {  # The configuration for the second logical system begins here.
    interfaces {
        fe-1/1/3 {
            unit 0 {  # This logical interface connects to LS2 on Router PE1.
                description "ls2 interface";
                vlan-id 100;
                family inet {
                    address 10.1.2.2/24;
                }
                family iso;
                family mpls;
```

```
            }
          }
          so-1/2/0 {
            unit 0 { # This logical interface connects to LS2 on Router PE2.
              description "ls2 interface";
              dlci 100;
              family inet {
                address 10.1.3.1/24;
              }
              family iso;
              family mpls;
            }
          }
          lo0 {
            unit 2 {
              description "ls2 loopback";
              family inet {
                address 10.20.20.21/32;
              }
              family iso {
                address 47.2222.2222.2222.2223.00;
              }
            }
          }
        }
        protocols { # You configure RSVP, MPLS, and IS-IS for logical system LS2.
          rsvp  {
            interface all;
          }
          mpls {
            interface all;
          }
          isis {
            interface fe-1/1/3.0;
            interface so-1/2/0.0;
            interface lo0.2;
          }
        }
      }
    }
    interfaces {
      fe-1/1/3 {
        vlan-tagging;
        unit 3 { # This logical interface connects to the main router on Router PE1.
          description "main router interface";
          vlan-id 103;
          family inet {
            address 10.31.2.2/24;
          }
          family iso;
          family mpls;
        }
      }
      so-1/2/0 {
        dce;  # You must configure all physical interface statements for logical
        encapsulation frame-relay; # routers at the [edit interfaces] hierarchy level.
```

```
        unit 3 { # This logical interface connects to the main router on Router PE2.
            description "main router interface";
            dlci 103;
            family inet {
                address 10.31.3.1/24;
            }
            family iso;
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            description "main router loopback";
            family inet {
                address 10.255.41.175/32;
            }
        }
    }
}
routing-options {
    autonomous-system 500;
}
protocols { # You configure BGP and OSPF for the main router.
    bgp {
        group to_main_ls {
            type internal;
            local-address 10.255.41.175
            neighbor 10.255.41.179;
            neighbor 10.255.41.173;
        }
    }
    ospf {
        area 0.0.0.0 {
            interface lo0.0;
            interface fe-1/1/3.3;
            interface so-1/2/0.3;
        }
    }
}
system {
    login {
        class ls1–admin {
            permissions all;
            logical-system ls1;
        }
        class ls1–onlooker {
            permissions view;
            logical-system ls2;
        }
        user ls1–admin {
            class ls1–admin;
        }
    }
}
```

On Router PE2, create two VRF routing instances in logical system LS1: red and blue. Configure the CE-facing logical interfaces so that traffic from Router CE5 is placed in the red VPN and traffic from Router CE6 is placed in the blue VPN. Next, create one logical interface on so-1/2/0.1 to connect to logical system LS1 on Router P0.

Also on Router PE2, create a VPLS routing instance in logical system LS2. Configure a logical interface so that traffic from Router CE7 is sent into the VPLS domain and connects to logical system LS2 on Router P0.

Configure a logical interface to interconnect Router CE8 with the main router portion of Router P0.

Finally, you can optionally create a logical system administrator that has configuration privileges for LS1 and viewing privileges for LS2.

Router PE2

```
[edit]
logical-systems {
   ls1 { # The configuration for the first logical system begins here.
      interfaces {
         fe-0/2/0 {
            unit 1 { # This logical interface connects to Router CE5.
               description "routing-instance red interface";
               vlan-id 101;
               family inet {
                  address 10.11.4.1/24;
               }
            }
            unit 2 { # This logical interface connects to Router CE6.
               description "routing-instance blue interface";
               vlan-id 102;
               family inet {
                  address 10.21.4.1/24;
               }
            }
         }
         so-1/2/0 {
            unit 1 {# This is the core-facing interface for logical system LS1.
               description "ls1 interface";
               dlci 101;
               family inet {
                  address 10.11.3.2/24;
               }
               family iso;
               family mpls;
            }
         }
         lo0 {
            unit 1 {
               description "ls1 loopback";
               family inet {
                  address 10.10.10.12/32;
               }
               family iso {
                  address 47.1111.1111.1111.1113.00;
               }
```

```
        }
    }
}
protocols {
    rsvp {# You configure RSVP, MPLS, IS-IS, and BGP for logical system LS1.
        interface all;
    }
    mpls {
        label-switched-path to_10.10.10.10 {
            to 10.10.10.10;
        }
        interface all;
    }
    bgp {
        group to_other_PE {
            type internal;
            local-address 10.10.10.12;
            family inet {
                any;
            }
            family inet-vpn {
                any;
            }
            neighbor 10.10.10.10;
        }
    }
    isis {
        interface all;
    }
}
policy-options {
    policy-statement from_bgp_to_ospf {
        then accept;
    }
}
routing-instances {
    blue {
        instance-type vrf; # You configure instance blue within logical system LS1.
        interface fe-0/2/2.0;
        route-distinguisher 10.10.10.12:200;
        vrf-target target:20:20;
        protocols {
            bgp { # BGP connects the blue instance with Router CE6.
                group to_CE {
                    local-address 10.21.4.1;
                    peer-as 300;
                    neighbor 10.21.4.2;
                }
            }
        }
    }
    red {
        instance-type vrf; # You configure instance red within logical system LS1.
        interface fe-0/2/1.0;
        route-distinguisher 10.10.10.12:100;
        vrf-target target:10:10;
```

```
        protocols {
            ospf { # OSPF connects the red instance with Router CE5.
                export from_bgp_to_ospf;
                area 0.0.0.0 {
                    interface all;
                }
            }
        }
    }
    routing-options {
        autonomous-system 100;
    }
}
logical-systems {
    ls2 { # The configuration for the second logical system begins here.
        interfaces {
            fe-0/2/0 {
                unit 0 { # This logical interface connects to Router CE7.
                    description "vpls interface";
                    encapsulation vlan-vpls;
                    vlan-id 600;
                    family vpls;
                }
            }
            so-1/2/0 {
                unit 0 { # This is the core-facing interface for logical system LS2.
                    description "ls2 interface";
                    dlci 100;
                    family inet {
                        address 10.1.3.2/24;
                    }
                    family iso;
                    family mpls;
                }
            }
            lo0 {
                unit 2 {
                    description "ls2 loopback";
                    family inet {
                        address 10.20.20.22/32;
                    }
                    family iso {
                        address 47.2222.2222.2222.2224.00;
                    }
                }
            }
        }
        protocols { # You configure RSVP, MPLS, IS-IS, and BGP for logical system LS2.
            rsvp {
                interface all;
            }
            mpls {
                label-switched-path to_10.20.20.20 {
                    to 10.20.20.20;
                }
```

```
            interface all;
        }
        bgp {
            group to_VPLS_PE {
                type internal;
                local-address 10.20.20.22;
                family l2vpn {
                    signaling;
                }
                neighbor 10.20.20.20;
            }
        }
        isis {
            interface so-1/2/0.0;
            interface lo0.2;
        }
    }
    routing-instances {
        new {
            instance-type vpls; # You configure VPLS within logical system LS2.
            interface fe-0/2/0.0;
            route-distinguisher 10.20.20.22:100;
            vrf-target target:30:30;
            protocols {
                vpls {
                    site-range 10;
                    site newPE {
                        site-identifier 2;
                    }
                }
            }
        }
    }
    routing-options {
        autonomous-system 400;
    }
}
interfaces {
    fe-0/2/0 {   # You must always configure physical interface statements for the
        vlan-tagging; # logical system interfaces at the [edit interfaces] hierarchy level.
        encapsulation vlan-vpls;
    }
    fe-0/2/1 {
        vlan-tagging;
    }
    fe-0/2/2 {
        vlan-tagging;
    }
    fe-0/2/3 {
        vlan-tagging;
        unit 0 { # This logical interface in the main router of PE2 connects to CE8.
            description "main router to CE8";
            vlan-id 103;
            family inet {
                address 10.31.4.1/24;
            }
```

```
        }
      }
      so-1/2/0 {
        encapsulation frame-relay;
        unit 3 { # This is the core-facing interface for the main router of PE2.
          description "main router to P0";
          dlci 103;
          family inet {
            address 10.31.3.2/24;
          }
          family iso;
          family mpls;
        }
      }
      lo0 {
        unit 0 {
          description "main router loopback";
          family inet {
            address 10.155.41.179/32;
          }
        }
      }
    }
    routing-options {
      static {
        route 10.255.41.180/32 next-hop 10.31.4.2;
      }
      autonomous-system 500;
    }
    protocols {
      bgp {# The main router uses BGP as the exterior gateway protocol.
        group to_main_ls {
          type internal;
          local-address 10.255.41.179;
          export export_address;
          neighbor 10.255.41.173;
          neighbor 10.255.41.175;
        }
      }
      ospf {# The main router uses OSPF as the interior gateway protocol.
        area 0.0.0.0 {
          interface so-1/2/0.3;
          interface fe-0/2/3.0;
          interface lo0.0;
        }
      }
    }
    policy-options {
      policy-statement export_address {
        from {
          route-filter 10.255.41.180/32 exact;
        }
        then accept;
      }
    }
}
```

```
system {
   login {
      class ls1–admin {
         permissions all;
         logical-system ls1;
      }
      class ls1–onlooker {
         permissions view;
         logical-system ls2;
      }
      user ls1–admin {
         class ls1–admin;
      }
   }
}
```

On Router CE5, configure OSPF to connect to the red VPN in logical system LS1 on Router PE2:

Router CE5

```
[edit]
interfaces {
   fe-0/3/1 {
      vlan-tagging;
      unit 0 {
         description "routing-instance red CE";
         vlan-id 101;
         family inet {
            address 10.11.4.2/24;
         }
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.11.4.100/32;
         }
      }
   }
}
protocols {
   ospf {
      area 0.0.0.0 {
         interface fe-0/3/1.0;
         interface lo0.0;
      }
   }
}
system {
   login {
      class ls1–admin {
         permissions all;
         logical-system ls1;
      }
      class ls1–onlooker {
         permissions view;
         logical-system ls2;
```

```
        }
        user ls1–admin {
            class ls1–admin;
        }
    }
}
```

On Router CE6, configure BGP to connect to the blue VPN in logical system LS1 on Router PE2:

Router CE6

```
[edit]
interfaces {
    fe-0/3/2 {
        vlan-tagging;
        unit 0 {
            description "routing-instance blue CE";
            vlan-id 102;
            family inet {
                address 10.21.4.2/24;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.21.4.100/32;
            }
        }
    }
}
routing-options {
    autonomous-system 300;
}
protocols {
    bgp {
        export export_loopback;
        group to_PE {
            type external;
            local-address 10.21.4.2;
            peer-as 100;
            neighbor 10.21.4.1;
        }
    }
}
policy-options {
    policy-statement export_loopback {
        from {
            route-filter 10.21.4.100/32 exact;
        }
        then accept;
    }
}
```

On Router CE7, configure the Fast Ethernet interface in VLAN 600 to connect with the VPLS routing instance in logical system LS2 on Router PE2:

Router CE7

```
[edit]
interfaces {
    fe-0/3/0 {
        vlan-tagging;
        unit 0 {
            description "vpls interface";
            vlan-id 600;
            family inet {
                address 10.1.1.2/24;
            }
        }
    }
}
```

On Router CE8, configure the Fast Ethernet interface to connect with the main router at Router PE2:

Router CE8

```
[edit]
interfaces {
    fe-0/3/3 {
        vlan-tagging;
        unit 0 {
            description "main router interface";
            vlan-id 103;
            family inet {
                address 10.31.4.2/24;
            }
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.255.41.180/32;
        }
    }
}
```

Verifying Your Work

To verify the proper operation of logical systems, the master administrator can use the following commands:

- show bgp summary (logical-system *logical-system-name*)
- show isis adjacency (logical-system *logical-system-name*)
- show mpls lsp (logical-system *logical-system-name*)
- show ospf neighbor (logical-system *logical-system-name*)
- show route (logical-system *logical-system-name*)
- show route protocol protocol (logical-system *logical-system-name*)

- show rsvp session (logical-system *logical-system-name*)
- logical-system (*logical-system-name*)

The following sections show the output of commands used with the configuration example:

- Router CE1 Status on page 225
- Router CE2 Status on page 226
- Router CE3 Status on page 226
- Router PE1 Status: Main Router on page 226
- Router PE1 Status: LS1 on page 227
- Router PE1 Status: LS2 on page 230
- Router P0 Status: Main Router on page 230
- Router P0 Status: LS1 on page 231
- Router P0 Status: LS2 on page 231
- Router PE2 Status: Main Router on page 231
- Router PE2 Status: LS1 on page 233
- Router PE2 Status: LS2 on page 234
- Router CE5 Status on page 235
- Router CE6 Status on page 235
- Router CE7 Status on page 236
- Logical System Administrator Verification Output on page 236
- Verifying Routing Instance Connectivity on page 236

Router CE1 Status

```
user@CE1> show route table
inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.11.1.0/24       *[Direct/0] 00:20:20
                    > via fe-1/0/1.0
10.11.1.1/32       *[Local/0] 00:20:24
                      Local via fe-1/0/1.0
10.11.1.100/32     *[Direct/0] 00:21:53
                    > via lo0.0
10.11.4.0/24       *[OSPF/150] 00:18:30, metric 0, tag 3489661028
                    > to 10.11.1.2 via fe-1/0/1.0
10.11.4.100/32     *[OSPF/10] 00:18:30, metric 2
                    > to 10.11.1.2 via fe-1/0/1.0
224.0.0.5/32       *[OSPF/10] 00:21:58, metric 1
                      MultiRecv
```

Router CE2 Status

```
user@CE2> show route table
inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.21.1.0/24       *[Direct/0] 00:20:30
                    > via fe-1/0/2.0
10.21.1.1/32       *[Local/0] 00:20:34
                      Local via fe-1/0/2.0
10.21.1.100/32     *[Direct/0] 00:22:03
                    > via lo0.0
10.21.4.0/24       *[BGP/170] 00:18:43, localpref 100
                      AS path: 100 I
                    > to 10.21.1.2 via fe-1/0/2.0
10.21.4.100/32     *[BGP/170] 00:18:43, localpref 100
                      AS path: 100 300 I
                    > to 10.21.1.2 via fe-1/0/2.0
```

Router CE3 Status

```
user@CE3> show route table
inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.1.0/24       *[Direct/0] 00:20:13
                    > via fe-1/0/0.0
10.1.1.1/32       *[Local/0] 00:20:17
                      Local via fe-1/0/0.0
```

Router PE1 Status: Main Router

```
user@PE1> show bgp summary

Groups: 1 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 1          0          0          0          0          0
Peer               AS      InPkt     OutPkt    OutQ   Flaps Last Up/DwnState|#Active/Received/Damped...
10.255.41.175     500          5          8       0       0        2:31 0/0/0                0/0/0
10.255.41.179     500          6          9       0       0        2:35 0/1/0                0/0/0
```

```
user@PE1>  show route protocol bgp
inet.0: 20 destinations, 21 routes (20 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.255.41.180/32    [BGP/170] 00:02:48, localpref 100, from 10.255.41.179
                      AS path: I
                    > to 10.31.2.2 via fe-0/0/1.3
iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

user@PE1>  show ospf neighbor
  Address         Interface             State      ID              Pri  Dead
```

```
10.31.2.2        fe-0/0/1.3             Full      10.255.41.175    128   32

user@PE1>  show isis adjacency
IS-IS instance is not running
```

Router PE1 Status: LS1

The master administrator can issue the following command to view the output for a specific logical system.

```
user@PE1> show bgp summary logical-system ls1

Groups: 2 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l3vpn.0            4          4          0          0          0          0
bgp.l3vpn.2            0          0          0          0          0          0
Peer               AS      InPkt     OutPkt    OutQ   Flaps Last Up/DwnState|#Active/Received/Damped...
10.10.10.12       100         13         14       0       0        2:50 Establ
  bgp.l3vpn.0: 4/4/0
  bgp.l3vpn.2: 0/0/0
  blue.inet.0: 2/2/0
  red.inet.0: 2/2/0
10.21.1.1         200         13         14       0       0        4:33 Establ
  blue.inet.0: 1/1/0
```

The logical system administrator for LS1 will see the same output when the following command is issued.

```
ls1-admin@PE1:ls1> show bgp summary

Groups: 2 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l3vpn.0            4          4          0          0          0          0
bgp.l3vpn.2            0          0          0          0          0          0
Peer               AS      InPkt     OutPkt    OutQ   Flaps Last Up/DwnState|#Active/Received/Damped...
10.10.10.12       100         13         14       0       0        2:50 Establ
  bgp.l3vpn.0: 4/4/0
  bgp.l3vpn.2: 0/0/0
  blue.inet.0: 2/2/0
  red.inet.0: 2/2/0
10.21.1.1         200         13         14       0       0        4:33 Establ
  blue.inet.0: 1/1/0
```

Red VPN The master administrator can issue the following command to view the output for a specific logical system.

```
user@PE1> show route logical-system ls1 table red
red.inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.11.1.0/24       *[Direct/0] 00:04:51
                    > via fe-0/1/1.0
10.11.1.2/32       *[Local/0] 00:05:45
                      Local via fe-0/1/1.0
10.11.1.100/32     *[OSPF/10] 00:04:02, metric 1
```

```
                    > to 10.11.1.1 via fe-0/1/1.0
10.11.4.0/24        *[BGP/170] 00:03:05, localpref 100, from 10.10.10.12
                      AS path: I
                    > to 10.11.2.2 via fe-0/0/1.1, label-switched-path
to_10.10.10.12
10.11.4.100/32      *[BGP/170] 00:03:05, MED 1, localpref 100, from 10.10.10.12
                      AS path: I
                    > to 10.11.2.2 via fe-0/0/1.1, label-switched-path
to_10.10.10.12
224.0.0.5/32        *[OSPF/10] 00:07:02, metric 1
                      MultiRecv
```

The logical system administrator for LS1 will see the same output when the following command is issued.

```
ls1-admin@PE1:ls1> show route table red
red.inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.11.1.0/24        *[Direct/0] 00:04:51
                    > via fe-0/1/1.0
10.11.1.2/32        *[Local/0] 00:05:45
                      Local via fe-0/1/1.0
10.11.1.100/32      *[OSPF/10] 00:04:02, metric 1
                    > to 10.11.1.1 via fe-0/1/1.0
10.11.4.0/24        *[BGP/170] 00:03:05, localpref 100, from 10.10.10.12
                      AS path: I
                    > to 10.11.2.2 via fe-0/0/1.1, label-switched-path
to_10.10.10.12
10.11.4.100/32      *[BGP/170] 00:03:05, MED 1, localpref 100, from 10.10.10.12
                      AS path: I
                    > to 10.11.2.2 via fe-0/0/1.1, label-switched-path
to_10.10.10.12
224.0.0.5/32        *[OSPF/10] 00:07:02, metric 1
                      MultiRecv
```

Blue VPN

The master administrator can issue the following command to view the output for a specific logical system.

```
user@PE1> show route logical-system ls1 table blue
blue.inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.21.1.0/24        *[Direct/0] 00:05:29
                    > via fe-0/1/2.0
10.21.1.2/32        *[Local/0] 00:06:23
                      Local via fe-0/1/2.0
10.21.1.100/32      *[BGP/170] 00:05:26, localpref 100
                      AS path: 200 I
                    > to 10.21.1.1 via fe-0/1/2.0
10.21.4.0/24        *[BGP/170] 00:03:43, localpref 100, from 10.10.10.12
                      AS path: I
                    > to 10.11.2.2 via fe-0/0/1.1, label-switched-path
to_10.10.10.12
10.21.4.100/32      *[BGP/170] 00:03:43, localpref 100, from 10.10.10.12
                      AS path: 300 I
                    > to 10.11.2.2 via fe-0/0/1.1, label-switched-path
to_10.10.10.12
```

```
user@PE1>  show route logical-system ls1 table inet.0
inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.10.10/32     *[Direct/0] 00:08:05
                    > via lo0.1
10.10.10.11/32     *[IS-IS/15] 00:05:07, metric 10
                    > to 10.11.2.2 via fe-0/0/1.1
10.10.10.12/32     *[IS-IS/15] 00:04:58, metric 20
                    > to 10.11.2.2 via fe-0/0/1.1
10.11.2.0/24       *[Direct/0] 00:05:38
                    > via fe-0/0/1.1
10.11.2.1/32       *[Local/0] 00:06:51
                      Local via fe-0/0/1.1
10.11.3.0/24       *[IS-IS/15] 00:05:07, metric 20
                    > to 10.11.2.2 via fe-0/0/1.1

user@PE1>  ping logical-system ls1 routing-instance red 10.11.4.100
PING 10.11.4.100 (10.11.4.100): 56 data bytes
64 bytes from 10.11.4.100: icmp_seq=0 ttl=251 time=1.055 ms
^C
--- 10.11.4.100 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.055/1.055/1.055/0.000 ms
```

The logical system administrator for LS1 sees the same output when issuing the following command.

```
ls1-admin@PE1:ls1> show route table blue
blue.inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.21.1.0/24       *[Direct/0] 00:05:29
                    > via fe-0/1/2.0
10.21.1.2/32       *[Local/0] 00:06:23
                      Local via fe-0/1/2.0
10.21.1.100/32     *[BGP/170] 00:05:26, localpref 100
                      AS path: 200 I
                    > to 10.21.1.1 via fe-0/1/2.0
10.21.4.0/24       *[BGP/170] 00:03:43, localpref 100, from 10.10.10.12
                      AS path: I
                    > to 10.11.2.2 via fe-0/0/1.1, label-switched-path
to_10.10.10.12
10.21.4.100/32     *[BGP/170] 00:03:43, localpref 100, from 10.10.10.12
                      AS path: 300 I
                    > to 10.11.2.2 via fe-0/0/1.1, label-switched-path
to_10.10.10.12

ls1-admin@PE1:ls1>  show route table inet.0
inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.10.10/32     *[Direct/0] 00:08:05
                    > via lo0.1
10.10.10.11/32     *[IS-IS/15] 00:05:07, metric 10
                    > to 10.11.2.2 via fe-0/0/1.1
10.10.10.12/32     *[IS-IS/15] 00:04:58, metric 20
                    > to 10.11.2.2 via fe-0/0/1.1
10.11.2.0/24       *[Direct/0] 00:05:38
                    > via fe-0/0/1.1
```

```
10.11.2.1/32       *[Local/0] 00:06:51
                      Local via fe-0/0/1.1
10.11.3.0/24       *[IS-IS/15] 00:05:07, metric 20
                    > to 10.11.2.2 via fe-0/0/1.1

ls1-admin@PE1:ls1>  ping routing-instance red 10.11.4.100
PING 10.11.4.100 (10.11.4.100): 56 data bytes
64 bytes from 10.11.4.100: icmp_seq=0 ttl=251 time=1.055 ms
^C
--- 10.11.4.100 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.055/1.055/1.055/0.000 ms
```

Router PE1 Status: LS2

```
user@PE1> show vpls connections logical-system ls2
Layer-2 VPN Connections:

Legend for connection status (St)
OR -- out of range              WE -- intf encaps != instance encaps
EI -- encapsulation invalid     Dn -- down
EM -- encapsulation mismatch    VC-Dn -- Virtual circuit down
CM -- control-word mismatch     -> -- only outbound conn is up
CN -- circuit not provisioned   <- -- only inbound conn is up
OL -- no outgoing label         Up -- operational
NC -- intf encaps not CCC/TCC   XX -- unknown
NP -- intf h/w not present

Legend for interface status
Up -- operational
Dn -- down

Instance: new
Local site: newPE (1)
    connection-site           Type  St     Time last up          # Up trans
    2                         rmt   Up     Jul 16 14:05:25 2003           1
      Local interface: vt-1/2/0.49152, Status: Up, Encapsulation: VPLS
      Remote PE: 10.20.20.22, Negotiated control-word: No
      Incoming label: 800001, Outgoing label: 800000
```

Router P0 Status: Main Router

```
user@P0> show interfaces terse lo0
Interface               Admin Link Proto Local                 Remote
lo0                     up    up
lo0.0                   up    up   inet  10.255.41.175       --> 0/0
                                        127.0.0.1           --> 0/0
                                   iso
47.0005.80ff.f800.0000.0108.0003.0102.5501.4175.00
                                   inet6 fe80::2a0:a5ff:fe12:2b09
                                         feee::10:255:14:175
lo0.1                   up    up   inet  10.10.10.11         --> 0/0
                                   iso   47.1111.1111.1111.1112.00
lo0.2                   up    up   inet  10.20.20.21         --> 0/0
                                   iso   47.2222.2222.2222.2223.00
lo0.16383               up    up   inet
user@P0> show ospf neighbor
```

```
  Address         Interface             State      ID              Pri  Dead
10.31.2.1        fe-1/1/3.3            Full       10.255.41.173   128   34
10.31.3.2        so-1/2/0.3            Full       10.255.41.179   128   37
```

Router P0 Status: LS1

```
user@P0> show isis adjacency logical-system ls1
Interface             System         L State       Hold (secs) SNPA
fe-1/1/3.1            PE1         2 Up                   21  0:90:69:9:4:1
fe-1/1/3.1            PE1         1 Up                   24  0:90:69:9:4:1
so-1/2/0.1            PE2          3 Up                   25

user@P0>  show bgp summary logical-system ls1
BGP is not running

user@P0>  show route protocol isis logical-system ls1
inet.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.10.10.10/32     *[IS-IS/15] 00:09:15, metric 10
                    > to 10.11.2.1 via fe-1/1/3.1
10.10.10.12/32     *[IS-IS/15] 00:09:39, metric 10
                    > to 10.11.3.2 via so-1/2/0.1

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

mpls.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
```

Router P0 Status: LS2

```
user@P0> show bgp summary logical-system ls2
BGP is not running

user@P0>  show isis adjacency logical-system ls2
Interface             System         L State       Hold (secs) SNPA
fe-1/1/3.0            PE1            2 Up                   24  0:90:69:9:4:1
fe-1/1/3.0            PE1            1 Up                   23  0:90:69:9:4:1
so-1/2/0.0            PE2            3 Up                   24

user@P0>  show route protocol isis logical-system ls2

inet.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.20.20.20/32     *[IS-IS/15] 00:09:44, metric 10
                    > to 10.1.2.1 via fe-1/1/3.0
10.20.20.22/32     *[IS-IS/15] 00:09:45, metric 10
                    > to 10.1.3.2 via so-1/2/0.0

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

mpls.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
```

Router PE2 Status: Main Router

```
user@PE2> show ospf neighbor
```

```
                         Address         Interface             State      ID              Pri  Dead
                       10.31.4.2         fe-0/2/3.0            Full       10.255.41.180   128   38
                       10.31.3.1         so-1/2/0.3            Full       10.255.41.175   128   36

user@PE2>  show interfaces terse lo0
Interface              Admin Link Proto Local                 Remote
lo0                    up    up
lo0.0                  up    up   inet  10.255.41.179       --> 0/0
                                       127.0.0.1           --> 0/0
                                  iso   47.0005.80ff.f800.0000.0108.0003.0102.5501.4179.00
                                  inet6 fe80::2a0:a5ff:fe12:29ff
                                        feee::10:255:14:179
lo0.1                  up    up   inet  10.10.10.12         --> 0/0
                                  iso   47.1111.1111.1111.1113.00
lo0.2                  up    up   inet  10.20.20.22         --> 0/0
                                  iso   47.2222.2222.2222.2224.00
lo0.16383              up    up   inet

user@PE2>  show bgp summary
Groups: 1 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 1          1          0          0          0          0
Peer              AS      InPkt     OutPkt    OutQ   Flaps Last Up/DwnState|#Active/Received/Damped...
10.255.41.175     500         24         27       0       0       11:46 0/0/0                0/0/0
10.255.41.173     500         25         25       0       0       11:11 1/1/0                0/0/0


                       user@PE2>  show route protocol ospf

                       inet.0: 20 destinations, 22 routes (19 active, 0 holddown, 1 hidden)
                       + = Active Route, - = Last Active, * = Both

                       10.255.41.175/32   *[OSPF/10] 00:00:20, metric 1
                                           > via so-1/2/0.3
                       10.255.41.180/32    [OSPF/10] 00:00:20, metric 1
                                           > to 10.31.4.2 via fe-0/2/3.0
                       10.255.41.173/32    *[OSPF/10] 00:00:20, metric 2
                                           > via so-1/2/0.3
                       10.31.2.0/24       *[OSPF/10] 00:00:20, metric 2
                                           > via so-1/2/0.3
                       10.31.3.0/24        [OSPF/10] 00:00:20, metric 1
                                           > via so-1/2/0.3
                       224.0.0.5/32       *[OSPF/10] 00:13:46, metric 1
                                             MultiRecv

                       iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

                       inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

                       user@PE2>  show route protocol bgp

                       inet.0: 20 destinations, 22 routes (19 active, 0 holddown, 1 hidden)
                       + = Active Route, - = Last Active, * = Both

                       10.255.41.177/32   *[BGP/170] 00:11:23, localpref 100, from 10.255.41.173
                                             AS path: I
                                           > via so-1/2/0.3

                       iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
```

```
inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
```

Router PE2 Status: LS1

```
user@PE2> show bgp summary logical-system ls1

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
inet.2                 0          0          0          0          0          0
bgp.l3vpn.0            4          4          0          0          0          0
bgp.l3vpn.2            0          0          0          0          0          0
Peer               AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Damped...
10.10.10.10       100         29         31       0       0       11:25 Establ
  bgp.l3vpn.0: 4/4/0
  bgp.l3vpn.2: 0/0/0
  blue.inet.0: 2/2/0
  red.inet.0: 2/2/0
10.21.4.2         300         27         28       0       0       11:40 Establ
  blue.inet.0: 1/1/0
```

Red VPN

```
user@PE2> show route logical-system ls1 table red
red.inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.11.1.0/24       *[BGP/170] 00:12:02, localpref 100, from 10.10.10.10
                      AS path: I
                    > via so-1/2/0.1, label-switched-path to_10.10.10.10
10.11.1.100/32     *[BGP/170] 00:12:02, MED 1, localpref 100, from 10.10.10.10
                      AS path: I
                    > via so-1/2/0.1, label-switched-path to_10.10.10.10
10.11.4.0/24       *[Direct/0] 00:13:22
                    > via fe-0/2/1.0
10.11.4.1/32       *[Local/0] 00:13:29
                      Local via fe-0/2/1.0
10.11.4.100/32     *[OSPF/10] 00:12:35, metric 1
                    > to 10.11.4.2 via fe-0/2/1.0
224.0.0.5/32       *[OSPF/10] 00:15:02, metric 1
                      MultiRecv
```

Blue VPN

```
user@PE2> show route logical-system ls1 table blue
blue.inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.21.1.0/24       *[BGP/170] 00:13:12, localpref 100, from 10.10.10.10
                      AS path: I
                    > via so-1/2/0.1, label-switched-path to_10.10.10.10
10.21.1.100/32     *[BGP/170] 00:13:12, localpref 100, from 10.10.10.10
                      AS path: 200 I
                    > via so-1/2/0.1, label-switched-path to_10.10.10.10
10.21.4.0/24       *[Direct/0] 00:14:32
                    > via fe-0/2/2.0
10.21.4.1/32       *[Local/0] 00:14:39
                      Local via fe-0/2/2.0
10.21.4.100/32     *[BGP/170] 00:13:27, localpref 100
                      AS path: 300 I
                    > to 10.21.4.2 via fe-0/2/2.0
```

```
user@PE2>  show mpls lsp logical-system ls1
Ingress LSP: 1 sessions
To              From            State Rt ActivePath       P     LSPname
10.10.10.10     10.10.10.12     Up     0                  *     to_10.10.10.10
Total 1 displayed, Up 1, Down 0
Egress LSP: 1 sessions
To              From            State Rt Style Labelin Labelout LSPname
10.10.10.12     10.10.10.10     Up     0 1 FF       3        - to_10.10.10.12
Total 1 displayed, Up 1, Down 0
Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

user@PE2>  show rsvp session logical-system ls1
Ingress RSVP: 1 sessions
To              From            State Rt Style Labelin Labelout LSPname
10.10.10.10     10.10.10.12     Up     0 1 FF       -   100000 to_10.10.10.10
Total 1 displayed, Up 1, Down 0
Egress RSVP: 1 sessions
To              From            State Rt Style Labelin Labelout LSPname
10.10.10.12     10.10.10.10     Up     0 1 FF       3        - to_10.10.10.12
Total 1 displayed, Up 1, Down 0
Transit RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Router PE2 Status: LS2

```
user@PE2> show vpls connections logical-system ls2
Layer-2 VPN Connections:

Legend for connection status (St)
OR -- out of range              WE -- intf encaps != instance encaps
EI -- encapsulation invalid     Dn -- down
EM -- encapsulation mismatch    VC-Dn -- Virtual circuit down
CM -- control-word mismatch     -> -- only outbound conn is up
CN -- circuit not provisioned   <- -- only inbound  conn is up
OL -- no outgoing label         Up -- operational
NC -- intf encaps not CCC/TCC   XX -- unknown
NP -- intf h/w not present

Legend for interface status
Up -- operational
Dn -- down

Instance: new
Local site: newPE (2)
    connection-site           Type  St     Time last up          # Up trans
    1                         rmt   Up     Jul 16 14:05:25 2003           1
      Local interface: vt-1/1/0.40960, Status: Up, Encapsulation: VPLS
      Remote PE: 10.20.20.20, Negotiated control-word: No
      Incoming label: 800000, Outgoing label: 800001
```

```
user@PE2>  show bgp summary logical-system ls2
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l2vpn.0            1          1          0          0          0          0
Peer               AS      InPkt     OutPkt    OutQ   Flaps Last Up/DwnState|#Active/Received/Damped...
10.20.20.20       400         29         31       0       0       13:29 Establ
```

```
bgp.l2vpn.0: 1/1/0
new.l2vpn.0: 1/1/0
```

```
user@PE2>  show mpls lsp logical-system ls2
Ingress LSP: 1 sessions
To              From            State Rt ActivePath       P     LSPname
10.20.20.20     10.20.20.22     Up     0                  *     to_10.20.20.20
Total 1 displayed, Up 1, Down 0

Egress LSP: 1 sessions
To              From            State Rt Style Labelin Labelout LSPname
10.20.20.22     10.20.20.20     Up     0 1 FF       3        - to_10.20.20.22
Total 1 displayed, Up 1, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

user@PE2>  show rsvp session logical-system ls2
Ingress RSVP: 1 sessions
To              From            State Rt Style Labelin Labelout LSPname
10.20.20.20     10.20.20.22     Up     0 1 FF       -   100016 to_10.20.20.20
Total 1 displayed, Up 1, Down 0

Egress RSVP: 1 sessions
To              From            State Rt Style Labelin Labelout LSPname
10.20.20.22     10.20.20.20     Up     0 1 FF       3        - to_10.20.20.22
Total 1 displayed, Up 1, Down 0

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

Router CE5 Status

```
user@CE5> show route table
inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.11.1.0/24       *[OSPF/150] 00:19:47, metric 0, tag 3489661028
                    > to 10.11.4.1 via fe-0/3/1.0
10.11.1.100/32     *[OSPF/10] 00:19:47, metric 2
                    > to 10.11.4.1 via fe-0/3/1.0
10.11.4.0/24       *[Direct/0] 00:21:12
                    > via fe-0/3/1.0
10.11.4.2/32       *[Local/0] 00:21:24
                      Local via fe-0/3/1.0
10.11.4.100/32     *[Direct/0] 00:22:37
                    > via lo0.0
224.0.0.5/32       *[OSPF/10] 00:22:44, metric 1
                      MultiRecv
```

Router CE6 Status

```
user@CE6> show route table
inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.21.1.0/24       *[BGP/170] 00:19:53, localpref 100
```

```
                      AS path: 100 I
                    > to 10.21.4.1 via fe-0/3/2.0
10.21.1.100/32     *[BGP/170] 00:19:53, localpref 100
                      AS path: 100 200 I
                    > to 10.21.4.1 via fe-0/3/2.0
10.21.4.0/24       *[Direct/0] 00:21:16
                    > via fe-0/3/2.0
10.21.4.2/32       *[Local/0] 00:21:28
                      Local via fe-0/3/2.0
10.21.4.100/32     *[Direct/0] 00:22:41
                    > via lo0.0
```

Router CE7 Status

```
user@CE7> show route table

inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.1.0/24       *[Direct/0] 00:21:03
                    > via fe-0/3/0.0
10.1.1.2/32       *[Local/0] 00:21:15
                      Local via fe-0/3/0.0
```

Logical System Administrator Verification Output

Because logical system administrators only have access to the configuration information of the logical system(s) to which they are assigned, the verification output will be limited to these logical systems as well. The following outputs show what the logical system administrator "ls1–admin" in this chapter's example configuration would see.

Verifying Routing Instance Connectivity

To verify that each pair of CE routers has end-to-end connectivity, issue the ping command on routers CE1, CE2, and CE3:

CE1 to CE5 (Red VPN)

```
user@CE1> ping 10.11.4.100
PING 10.11.4.100 (10.11.4.100): 56 data bytes
64 bytes from 10.11.4.100: icmp_seq=0 ttl=252 time=1.216 ms
64 bytes from 10.11.4.100: icmp_seq=1 ttl=252 time=1.052 ms
^C
--- 10.11.4.100 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.052/1.134/1.216/0.082 ms
```

CE2 to CE6 (Blue VPN)

```
user@CE2> ping 10.21.4.100
PING 10.21.4.100 (10.21.4.100): 56 data bytes
64 bytes from 10.21.4.100: icmp_seq=0 ttl=252 time=1.205 ms
64 bytes from 10.21.4.100: icmp_seq=1 ttl=252 time=1.021 ms
^C
--- 10.21.4.100 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.021/1.113/1.205/0.092 ms
```

CE3 to CE7 (VPLS)

```
user@CE3> ping 10.1.1.2
PING 10.1.1.2 (10.1.1.2): 56 data bytes
64 bytes from 10.1.1.2: icmp_seq=0 ttl=255 time=1.186 ms
64 bytes from 10.1.1.2: icmp_seq=1 ttl=255 time=1.091 ms
64 bytes from 10.1.1.2: icmp_seq=2 ttl=255 time=1.081 ms
^C
--- 10.1.1.2 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.081/1.119/1.186/0.047 ms
```

For More Information

Because the concepts that constitute logical systems cut across the entire JUNOS Software documentation set, you will find the following manuals to be useful references:

- For additional information about logical system administration, see the *JUNOS System Basics Configuration Guide*.
- For additional information about routing protocols, see the *JUNOS Routing Protocols Configuration Guide*.
- For additional information about policies, see the *JUNOS Policy Framework Configuration Guide*.
- For additional information about interface configuration, see the *JUNOS Network Interfaces Configuration Guide*.
- For additional information about MPLS and related protocols, see the *JUNOS MPLS Applications Configuration Guide*.
- For additional information about VPN protocols, see the *JUNOS VPNs Configuration Guide*.
- For additional information about multicast protocols, see the *JUNOS Multicast Protocols Configuration Guide*.
- For additional information about operational mode commands and output, see the *JUNOS Interfaces Command Reference*, the *JUNOS Routing Protocols and Policies Command Reference*, and the *JUNOS System Basics and Services Command Reference*.

Revision History

5 October 2009—10.0R1 Release. Roy Spencer.

13 June 2009—9.6R1 Release. Roy Spencer.

13 April 2009—9.5R1 Release. Roy Spencer.

15 January 2009—9.4R1 Release. Roy Spencer.

10 October 2008—Added support for logical systems to replace logical routers. Roy Spencer.

10 July 2008—9.2R1 Release. Roy Spencer.

10 April 2008—9.1R1 Release. Roy Spencer.

1 February 2008—9.0R1 Release. Fawn Damitio.

5 October 2007—Added support for logical router administrators and information about enhanced SNMP features. 8.5R1 Release. Fawn Damitio.

29 June 2007—Added support for SNMP usage limits. 8.4R1 Release. Fawn Damitio.

27 March 2007—8.3R1 Release. Fawn Damitio.

12 January 2007—Added support for MX960 Ethernet Services Routers. 8.2R1 Release. Fawn Damitio.

15 September 2006—8.1R1 Release. Richard Hendricks.

29 June 2006—8.0R1 Release. Richard Hendricks.

27 March 2006—7.6R1 Release. Richard Hendricks.

9 January 2006—Added SNMP support within a logical router, 7.5R1 Release. Richard Hendricks.

14 September 2005—Added multicast protocol RP and source DR functionality within a logical router, 7.4R1 Release. Richard Hendricks.

13 June 2005—7.3R1 Release. Richard Hendricks.

5 April 2005—7.2R1 Release. Richard Hendricks.

2 February 2005—7.1R1 Release. Richard Hendricks.

6 October 2004—Added support for implementing logical tunnel (lt) interfaces on an integrated Adaptive Services Module in an M7i router, 7.0R1 Release. Richard Hendricks.

6 July 2004—6.4R1 Release. Richard Hendricks.

5 April 2004—Revised lt interface families and encapsulation types and added support for graceful Routing Engine switchover, 6.3R1 Release. Richard Hendricks.

22 December 2003—Added new protocol family and encapsulation types for logical tunnel interfaces, 6.2R1 Release. Richard Hendricks.

22 September 2003—Added a new example, the logical tunnel (lt) interface type, and interface hierarchy changes. 6.1R1 Release. Richard Hendricks.

30 June 2003—Initial document written, Release 6.0R1. Richard Hendricks.

Chapter 8

OSPF Version 3 for IPv6

This chapter covers the following topics:

- OSPF Version 3 for IPv6 Overview on page 239
- OSPF Version 3 for IPv6 System Requirements on page 241
- OSPF Version 3 for IPv6 Terms and Acronyms on page 241
- Configuring OSPFv3 for IPv6 on page 241
- Configuring OSPFv3 as the Routing Protocol on page 242
- Configuring Interfaces in OSPFv3 Areas on page 242
- Configuring Virtual Links for OSPFv3 on page 242
- Example: Configuring OSPFv3 for IPv6 on page 243
- For More Information on page 267
- Revision History on page 267

OSPF Version 3 for IPv6 Overview

OSPF version 2, introduced as RFC 2328 in 1998, has been one of the most widely deployed interior gateway protocols (IGPs) for intradomain routing. The protocol is extended in version 3 (RFC 2740) to support OSPF in IPv6 networks. Most of the functionality of OSPFv2 carries over into OSPFv3, but there are some significant changes to explore.

OSPFv3 adds support for IPv6 in the Open Shortest Path First (OSPF) routing protocol, as detailed in RFC 2740. Most configuration and operational commands function essentially the same as in OSPFv2:

- All OSPFv3 operational and configuration commands include the identifier **ospf3** in place of the familiar **ospf** option. For example, **show ospf database** in OSPFv2 becomes **show ospf3 database** in OSPFv3.
- OSPFv3 Router IDs, Area IDs, and LSA link-state IDs remain at the OSPFv2 IPv4 size of 32 bits.
- All the optional capabilities in OSPFv2 for IPv4, such as not-so-stubby areas (NSSA), are supported in OSPFv3 for IPv6.

However, there are many significant changes to note about OSPFv3 for IPv6:

- Router link-state advertisements (LSAs) and Network LSAs no longer carry prefix information. In OSPFv3, these LSAs only carry topology information.

NOTE: Because addressing information in the LSA header, Router LSA, and Network LSA (Type 2) has been removed, the OSPFv3 protocol is designed to be network protocol independent.

- New and modified LSAs have been created to handle the flow of IPv6 addresses and prefixes in an OSPFv3 network. As a result, some show command output appears in a different format for OSPFv3. The LSAs that have been modified are:
  - Interarea-Prefix LSA—This replaces the Network Summary or Type 3 LSA.
  - Interarea Router LSA—This replaces the Autonomous System Boundary Router (ASBR) Summary or Type 4 LSA.

  New LSAs introduced in OSPFv3 are:
  - Link LSA—This LSA has local scope and does not extend beyond the link it is associated with. The purpose of a link LSA is to provide the router's IPv6 link-local address to neighbors, inform other routers of the associated IPv6 prefixes available on the link, and provide information to the Network LSA. On all OSPF interfaces except virtual links, OSPF packets are sent using the interface's link-local address as the source address.

NOTE: A link-local address is an IPv6 address that starts with the first 10 bits set to 1111111010. This is often displayed in hexadecimal as fe80.

Juniper Networks M Series Multiservice Edge Routers, Juniper Networks MX Series Ethernet Services Routers, and Juniper Networks T Series Core Routers automatically generate link-local addresses when IPv6 is enabled. The routing platform selects one interface MAC address (derived from the available interfaces) and appends this to the fe80 prefix with some additional bit stuffing. For more information about link-local addresses, see RFC 2373.

  - Intra-Area-Prefix LSA—This carries all IPv6 prefix information to all OSPFv3 routers within an area (this information in IPv4 is carried by the Router and Network LSAs).
- OSPFv3 now runs on a per-link basis, instead of on a per-IP-subnet basis.
- IPv6 link-local addresses are used for OSPFv3 neighbor exchanges (except over virtual links).
- The flooding scope for LSAs has been generalized into three categories for OSPFv3:
  - Link-local scope—The OSPFv3 packet is flooded to the members of a link.

  - Area scope—The OSPFv3 packet is flooded to all members of an OSPFv3 area.
  - AS scope—The OSPFv3 packet is flooded to all members of an AS.
- Authentication has been removed from the OSPFv3 protocol itself and relies on the authentication header (AH) and Encapsulating Security Payload (ESP) portions of the IP Security (IPsec) protocol for all authentication tasks in IPv6. For more information about configuring IPsec, see the *JUNOS IPsec Feature Guide*.
- Label-switched paths (LSPs) and traffic engineering are not supported in OSPFv3.
- Neighboring routers are always identified by the 32-bit router ID in OSPFv3.

OSPF Version 3 for IPv6 System Requirements

To implement OSPFv3 for IPv6, your system must meet these minimum requirements:

- JUNOS Release 8.2 or later for MX Series routing platforms
- JUNOS Release 7.2 or later for J Series Services Routers
- JUNOS Release 5.5 or later for M Series and T Series routers
- Two J Series, M Series, MX Series, or T Series routers

OSPF Version 3 for IPv6 Terms and Acronyms

L

link-state advertisement (LSA) A multi-tiered message format for OSPFv2 and OSPFv3 that carries information about the OSPF network to OSPF-enabled routers. The collection of LSAs forms the link-state database used by the routers to select optimum paths. Different LSA levels limit the scope of OSPF protocol message delivery to links, areas, or autonomous systems (ASs).

O

Open Shortest Path First (OSPF) A link-state IGP that makes routing decisions based on the shortest-path-first (SPF) algorithm (also referred to as the *Dijkstra algorithm*).

OSPFv3 The IPv6-enabled version of the OSPF protocol.

Configuring OSPFv3 for IPv6

To implement OSPFv3 for IPv6, you must configure the following:

- Configuring OSPFv3 as the Routing Protocol on page 242
- Configuring Interfaces in OSPFv3 Areas on page 242

■ Configuring Virtual Links for OSPFv3 on page 242

Configuring OSPFv3 as the Routing Protocol

You enable OSPFv3 almost the same way you enable OSPFv2. The only difference is that you use the ospf3 statement in place of ospf at the [edit protocols] hierarchy level.

```
[edit]
protocols {
   ospf3 {
      ...
   }
}
```

Configuring Interfaces in OSPFv3 Areas

To place selected interfaces in an OSPFv3 area, use the interface statement at the [edit protocols ospf3 area *area-number*] hierarchy level.

```
[edit]
protocols {
   ospf3 {
      area 0 {
         interface at-0/0/0.0;
         interface fe-1/1/1;
      }
   }
}
```

Configuring Virtual Links for OSPFv3

Virtual links can connect discontiguous sections of the OSPF backbone Area 0 or extend backbone access to areas not directly adjacent to Area 0 (a requirement of the OSPF protocol). To configure a virtual link, configure the virtual-link statement at the [edit protocols ospf3 area 0] hierarchy level. In the statement, specify the router ID of your neighbor (often the loopback interface IP address) and the OSPFv3 area that the virtual link travels across to reach Area 0.

```
[edit]
protocols {
   ospf3 {
      area 0.0.0.0 {
         virtual-link neighbor-id neighbor-router-id transit-area area;
      }
   }
}
```

Example: Configuring OSPFv3 for IPv6

Figure 18: OSPFv3 for IPv6 Topology Diagram

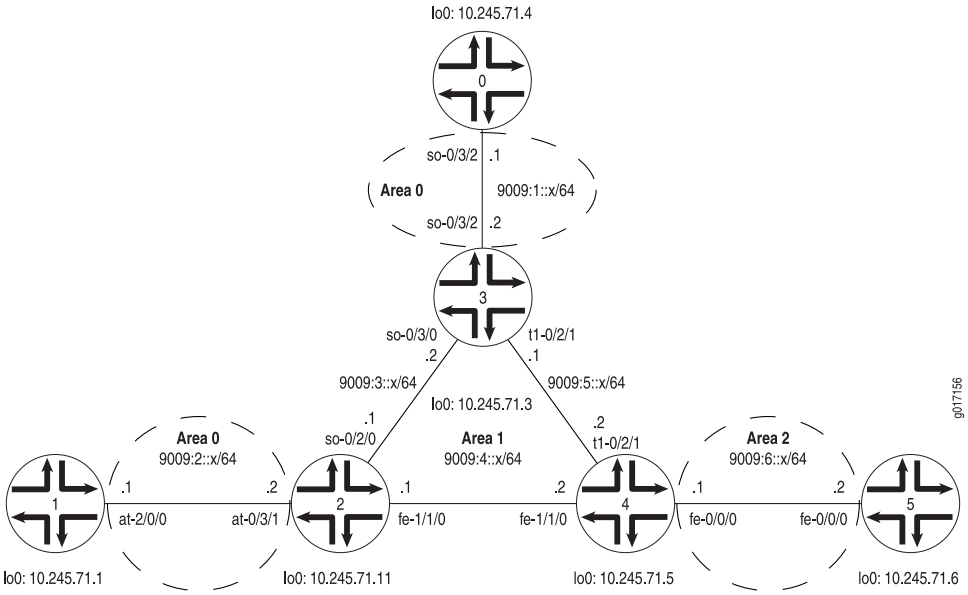


Figure 18 on page 243 shows an OSPFv3 topology. Routers 0, 1, 2, and 3 are connected to the OSPFv3 backbone Area 0; Routers 2, 3, and 4 connect to each other across Area 1; and Area 2 is located between Router 4 and Router 5. Because Router 5 does not have a direct adjacency to Area 0, a virtual link is required across Area 1 between Router 3 and Router 4. Similarly, because Routers 0 and 1 have two separate Area 0 backbone sections, you need to configure a second virtual link across Area 1 between Routers 2 and 3.

On Router 0, add the so-0/3/2 interface into Area 0 of the OSPFv3 process.

Router 0

```
[edit]
interfaces {
   so-0/3/2 {
      unit 0 {
         family inet {
            address 10.19.1.1/24;
         }
         family inet6 {
            address 9009:1::1/64;
         }
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.245.71.4/32;
         }
         family inet6 {
```

```
            address feee::10:255:71:4/128;
          }
        }
      }
    }
    protocols {
      ospf3 {
        area 0.0.0.0 {
          interface so-0/3/2.0;
          interface lo0.0 {
            passive;
          }
        }
      }
    }
```

On Router 1, add the at-2/0/0 interface into Area 0 of the OSPFv3 process:

Router 1

```
[edit]
interfaces {
  at-2/0/0 {
    atm-options {
      vpi 0;
    }
    unit 0 {
      vci 0.77;
      family inet {
        address 10.19.2.1/24;
      }
      family inet6 {
        address 9009:2::1/64;
      }
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.245.71.1/32;
      }
      family inet6 {
        address feee::10:255:71:1/128;
      }
    }
  }
}
protocols {
  ospf3 {
    area 0.0.0.0 {
      interface at-2/0/0.0;
      interface lo0.0 {
        passive;
      }
    }
  }
}
```

On Router 2, add the interfaces connected to Routers 1, 3, and 4 into the OSPFv3 process. You must also complete the virtual link to Router 3 through Area 1 so that Router 1 can access the discontiguous portion of the OSPF backbone found on Router 0.

Router 2

```
[edit]
interfaces {
    so-0/2/0 {
        unit 0 {
            family inet {
                address 10.19.3.1/24;
            }
            family inet6 {
                address 9009:3::1/64;
            }
        }
    }
    at-0/3/1 {
        atm-options {
            vpi 0 {
                maximum-vcs 1200;
            }
        }
        unit 0 {
            vci 0.77;
            family inet {
                address 10.19.2.2/24;
            }
            family inet6 {
                address 9009:2::2/64;
            }
        }
    }
    fe-1/1/0 {
        unit 0 {
            family inet {
                address 10.19.4.1/24;
            }
            family inet6 {
                address 9009:4::1/64;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.245.71.11/32;
            }
            family inet6 {
                address feee::10:255:71:11/128;
            }
        }
    }
}
protocols {
    ospf3 {
```

```
    area 0.0.0.0 {
      virtual-link neighbor-id 10.245.71.3 transit-area 0.0.0.1;
      interface at-0/3/1.0;
    }
    area 0.0.0.1 {
      interface so-0/2/0.0 {
        metric 1;
      }
      interface fe-1/1/0.0 {
        metric 10;
      }
      interface lo0.0 {
        passive;
      }
    }
  }
}
```

For the OSPFv3 process on Router 3, configure the interfaces connected to Router 2 and Router 4 into Area 1 and the interface connected to Router 0 into Area 0. You must also configure two virtual links through Area 1—one connecting to Router 2 and the second connecting to Router 4. The virtual links allow Router 5 to access the OSPF backbone, and connect the discontiguous sections of Area 0 located at Router 0 and Router 1.

Router 3

```
[edit]
interfaces {
  t1-0/2/1 {
    unit 0 {
      family inet {
        address 10.19.5.1/24;
      }
      family inet6 {
        address 9009:5::1/64;
      }
    }
  }
  so-0/3/0 {
    unit 0 {
      family inet {
        address 10.19.3.2/24;
      }
      family inet6 {
        address 9009:3::2/64;
      }
    }
  }
  so-0/3/2 {
    unit 0 {
      family inet {
        address 10.19.1.2/24;
      }
      family inet6 {
        address 9009:1::2/64;
      }
    }
```

```
    }
    lo0 {
      unit 0 {
        family inet {
          address 10.245.71.3/32;
        }
        family inet6 {
          address feee::10:255:71:3/128;
        }
      }
    }
  }
  protocols {
    ospf3 {
      area 0.0.0.0 {
        virtual-link neighbor-id 10.245.71.11 transit-area 0.0.0.1;
        virtual-link neighbor-id 10.245.71.5 transit-area 0.0.0.1;
        interface so-0/3/2.0;
      }
      area 0.0.0.1 {
        interface so-0/3/0.0 {
          metric 1;
        }
        interface t1-0/2/1.0 {
          metric 1;
        }
        interface lo0.0 {
          passive;
        }
      }
    }
  }
```

On Router 4, add the connected interfaces into the OSPFv3 process. You must also complete the virtual link to Router 3 through Area 1 so that Router 5 can access the OSPF backbone.

Router 4

```
[edit]
interfaces {
  fe-0/0/0 {
    unit 0 {
      family inet {
        address 10.19.6.1/24;
      }
      family inet6 {
        address 9009:6::1/64;
      }
    }
  }
  t1-0/2/1 {
    unit 0 {
      family inet {
        address 10.19.5.2/24;
      }
      family inet6 {
        address 9009:5::2/64;
```

```
            }
          }
        }
        fe-1/1/0 {
          unit 0 {
            family inet {
              address 10.19.4.2/24;
            }
            family inet6 {
              address 9009:4::2/64;
            }
          }
        }
        lo0 {
          unit 0 {
            family inet {
              address 10.245.71.5/32;
            }
            family inet6 {
              address feee::10:255:71:5/128;
            }
          }
        }
      }
      protocols {
        ospf3 {
          area 0.0.0.1 {
            interface fe-1/1/0.0 {
              metric 10;
            }
            interface t1-0/2/1.0 {
              metric 1;
            }
            interface lo0.0 {
              passive;
            }
          }
          area 0.0.0.0 {
            virtual-link neighbor-id 10.245.71.3 transit-area 0.0.0.1;
          }
          area 0.0.0.2 {
            interface fe-0/0/0.0;
          }
        }
      }
```

On Router 5, add the **fe-0/0/0** interface into the OSPFv3 process to complete this example:

Router 5

```
[edit]
interfaces {
  fe-0/0/0 {
    unit 0 {
      family inet {
        address 10.19.6.2/24;
      }
```

```
            family inet6 {
               address 9009:6::2/64;
            }
         }
      }
      lo0 {
         unit 0 {
            family inet {
               address 10.245.71.6/32;
            }
            family inet6 {
               address feee::10:255:71:6/128;
            }
         }
      }
   }
   protocols {
      ospf3 {
         area 0.0.0.2 {
            interface fe-0/0/0.0;
            interface lo0.0 {
               passive;
            }
         }
      }
   }
```

Verifying Your Work

To verify proper operation of OSPFv3 for IPv6, use the following commands:

- show ospf3 interface
- show ospf3 neighbor
- show ospf3 database
- show ospf3 route
- show interfaces terse (to see the IPv6 link local address assigned to the lo0 interface)

NOTE: To view prefix information, you must use the extensive option with the show ospf3 database command.

The following sections show the output of these commands used with the configuration example.

NOTE: In the below sample output, the stars indicate the "best" routes. These routes are the routes that are installed in the routing table.

- Router 0 Status on page 250
- Router 1 Status on page 253
- Router 2 Status on page 255
- Router 3 Status on page 258
- Router 4 Status on page 262
- Router 5 Status on page 265

Router 0 Status

```
user@router0> show ospf3 database
    OSPF link state database, area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Router      0.0.0.1          10.245.71.1       0x80000005   764  0x89ce  40
Router      0.0.0.1          10.245.71.3       0x80000006  1360  0x2357  72
Router     *0.0.0.1          10.245.71.4       0x80000004   758  0xc09c  40
Router      0.0.0.1          10.245.71.5       0x80000003  1891  0xf774  40
Router      0.0.0.1          10.245.71.11      0x80000005  1393  0x7f6b  56
InterArPfx  0.0.0.1          10.245.71.3       0x80000003   758  0x9f52  36
InterArPfx  0.0.0.2          10.245.71.3       0x80000003   616  0xb13d  36
InterArPfx  0.0.0.3          10.245.71.3       0x80000003   473  0x1da2  36
InterArPfx  0.0.0.4          10.245.71.3       0x80000003   458  0x99f0  44
InterArPfx  0.0.0.5          10.245.71.3       0x80000004  1058  0xbf22  36
InterArPfx  0.0.0.6          10.245.71.3       0x80000002  1958  0x5c67  36
InterArPfx  0.0.0.7          10.245.71.3       0x80000002  1816  0xf088  44
InterArPfx  0.0.0.8          10.245.71.3       0x80000002  1673  0xd3d6  36
InterArPfx  0.0.0.9          10.245.71.3       0x80000002  1658  0xa3df  44
InterArPfx  0.0.0.1          10.245.71.5       0x80000004   479  0xd50f  36
InterArPfx  0.0.0.2          10.245.71.5       0x80000003   310  0xa547  36
InterArPfx  0.0.0.3          10.245.71.5       0x80000003   913  0x1cbb  36
InterArPfx  0.0.0.5          10.245.71.5       0x80000003   163  0xddcd  36
InterArPfx  0.0.0.6          10.245.71.5       0x80000003    13  0xadd6  44
InterArPfx  0.0.0.7          10.245.71.5       0x80000002  2633  0x5f8a  36
InterArPfx  0.0.0.8          10.245.71.5       0x80000002  2488  0x427c  36
InterArPfx  0.0.0.9          10.245.71.5       0x80000002  2338  0xdcda  36
InterArPfx  0.0.0.10         10.245.71.5       0x80000002  2188  0x5929  44
InterArPfx  0.0.0.11         10.245.71.5       0x80000002  2038  0xc2af  44
InterArPfx  0.0.0.12         10.245.71.5       0x80000002   763  0x664   44
InterArPfx  0.0.0.1          10.245.71.11      0x80000003   463  0x6f7a  36
InterArPfx  0.0.0.2          10.245.71.11      0x80000003   328  0xa935  36
InterArPfx  0.0.0.3          10.245.71.11      0x80000003   193  0x427c  36
InterArPfx  0.0.0.4          10.245.71.11      0x80000003   163  0xd69d  44
InterArPfx  0.0.0.5          10.245.71.11      0x80000002  1993  0x6b78  36
InterArPfx  0.0.0.6          10.245.71.11      0x80000002  1963  0xd6dd  36
InterArPfx  0.0.0.7          10.245.71.11      0x80000002  1828  0x532c  44
InterArPfx  0.0.0.8          10.245.71.11      0x80000002  1663  0xa9f7  36
InterArPfx  0.0.0.9          10.245.71.11      0x80000002  1528  0x7901  44
InterArRtr  0.0.0.1          10.245.71.5       0x80000002   620  0xc69c  32
IntraArPfx  0.0.0.1          10.245.71.1       0x80000005   464  0x3f8   76
IntraArPfx  0.0.0.1          10.245.71.3       0x80000005  1509  0x5cc1  64
IntraArPfx *0.0.0.1          10.245.71.4       0x80000004   458  0xba44  64
```

```
IntraArPfx  0.0.0.1          10.245.71.11     0x80000003  1693  0xd835  64
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Extern     *0.0.0.1          10.245.71.4      0x80000003  1058  0x8449  36
Extern      0.0.0.1          10.245.71.6      0x80000003  1064  0xdc9e  36
    OSPF Link-Local link state database, interface so-0/3/2.0
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Link        0.0.0.6          10.245.71.3      0x80000004   158  0xae30  56
Link       *0.0.0.2          10.245.71.4      0x80000004   158  0x9e80  56

user@router0>  show ospf3 interface
Interface              State     Area            DR-ID           BDR-ID
Nbrs
lo0.0                  DRother  0.0.0.0         0.0.0.0         0.0.0.0
0
so-0/3/2.0             PtToPt   0.0.0.0         0.0.0.0         0.0.0.0
1

user@router0>  show ospf3 neighbor
ID               Interface              State     Pri   Dead
10.245.71.3      so-0/3/2.0             Full      128   34
  Neighbor-address fe80::201:afff:fe00:86ca

user@router0>  show ospf3 route
Prefix                                  Path   Route       NH   Metric
                                        type   type        type
10.245.71.1                             Intra  Router      IP   25
  NH-interface so-0/3/2.0
10.245.71.3                             Intra  Area BR     IP   12
  NH-interface so-0/3/2.0
10.245.71.5                             Intra  Area BR     IP   13
  NH-interface so-0/3/2.0
10.245.71.6                             Inter  AS BR       IP   33
  NH-interface so-0/3/2.0
10.245.71.11                            Intra  Area BR     IP   13
  NH-interface so-0/3/2.0
9009:1::/64                             Intra  Network     IP   12
  NH-interface so-0/3/2.0
9009:1::2/128                           Intra  Network     IP   12
  NH-interface so-0/3/2.0
9009:2::/64                             Intra  Network     IP   25
  NH-interface so-0/3/2.0
9009:2::2/128                           Intra  Network     IP   13
  NH-interface so-0/3/2.0
9009:3::/64                             Inter  Network     IP   13
  NH-interface so-0/3/2.0
9009:4::/64                             Inter  Network     IP   23
  NH-interface so-0/3/2.0
9009:5::/64                             Inter  Network     IP   13
  NH-interface so-0/3/2.0
9009:6::/64                             Inter  Network     IP   33
  NH-interface so-0/3/2.0
9009:110::/64                           Intra  Network     IP   27
  NH-interface so-0/3/2.0
9009:120::/64                           Inter  Network     IP   25
  NH-interface so-0/3/2.0
9009:130::/64                           Inter  Network     IP   15
  NH-interface so-0/3/2.0
9009:140::/64                           Inter  Network     IP   16
  NH-interface so-0/3/2.0
9009:150::/64                           Ext2   Network     IP   0
```

```
  NH-interface so-0/3/2.0
feee::10:255:71:1/128                       Intra  Network     IP   25
  NH-interface so-0/3/2.0
feee::10:255:71:3/128                       Inter  Network     IP   12
  NH-interface so-0/3/2.0
feee::10:255:71:4/128                       Intra  Network     IP   0
  NH-interface lo0.0
feee::10:255:71:5/128                       Inter  Network     IP   13
  NH-interface so-0/3/2.0
feee::10:255:71:6/128                       Inter  Network     IP   33
  NH-interface so-0/3/2.0
feee::10:255:71:11/128                      Inter  Network     IP   13
  NH-interface so-0/3/2.0

user@router0>  show interfaces terse
Interface       Admin Link Proto Local                 Remote
...
so-0/3/2        up    up
so-0/3/2.0      up    up   inet  10.19.1.1/24
                           inet6 9009:1::1/64
                                 fe80::201:afff:fe03:6fa1/64
...
lo0             up    up
lo0.0           up    up   inet  10.245.71.4         --> 0/0
                                 127.0.0.1           --> 0/0
                            inet6 fe80::201:afff:fe03:6fa1
                                 feee::10:255:71:4
...
```

To provide a comparison between OSPFv3 show commands and legacy OSPFv2 show commands, the following is some sample output of the OSPFv2 connection between Routers 0 and 3:

```
user@router0> show ospf interface
Interface             State     Area            DR ID           BDR ID
Nbrs
lo0.0                 DRother   0.0.0.0         0.0.0.0         0.0.0.0
0
lo0.0                 DRother   0.0.0.0         0.0.0.0         0.0.0.0
0
so-0/3/2.0            PtToPt    0.0.0.0         0.0.0.0         0.0.0.0
1

user@router0>  show ospf neighbor
  Address         Interface             State      ID              Pri  Dead
10.19.1.2        so-0/3/2.0            Full      10.245.71.3      128   35

user@router0>  show ospf database
    OSPF link state database, area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router   10.245.71.3      10.245.71.3      0x80000002   636  0x2  0x5c45  60
Router  *10.245.71.4      10.245.71.4      0x80000002   640  0x2  0x267a  60

user@router0>  show ospf route
Prefix               Path   Route       NH   Metric  NextHop       Nexthop

                     Type   Type        Type         Interface     addr/label
10.245.71.3          Intra  Router      IP   1       so-0/3/2.0
10.19.1.0/24         Intra  Network     IP   1       so-0/3/2.0
```

```
10.245.71.3/32       Intra  Network     IP   1       so-0/3/2.0
10.245.71.4/32       Intra  Network     IP   0       lo0.0
```

Router 1 Status

```
user@router1> show ospf3 interface
Interface             State     Area            DR-ID           BDR-ID
Nbrs
at-2/0/0.0            PtToPt    0.0.0.0         0.0.0.0         0.0.0.0
1
ge-1/1/0.0            DRother   0.0.0.0         0.0.0.0         0.0.0.0
0
lo0.0                 DRother   0.0.0.0         0.0.0.0         0.0.0.0
0

user@router1>  show ospf3 neighbor
ID               Interface              State     Pri   Dead
10.245.71.11     at-2/0/0.0             Full      128   36
  Neighbor-address fe80::2a0:a5ff:fe3d:56

user@router1>  show ospf3 database
    OSPF link state database, area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Router     *0.0.0.1          10.245.71.1       0x80000005   574  0x89ce  40
Router      0.0.0.1          10.245.71.3       0x80000006  1174  0x2357  72
Router      0.0.0.1          10.245.71.4       0x80000004   574  0xc09c  40
Router      0.0.0.1          10.245.71.5       0x80000003  1706  0xf774  40
Router      0.0.0.1          10.245.71.11      0x80000005  1205  0x7f6b  56
InterArPfx  0.0.0.1          10.245.71.3       0x80000003   572  0x9f52  36
InterArPfx  0.0.0.2          10.245.71.3       0x80000003   430  0xb13d  36
InterArPfx  0.0.0.3          10.245.71.3       0x80000003   288  0x1da2  36
InterArPfx  0.0.0.4          10.245.71.3       0x80000003   273  0x99f0  44
InterArPfx  0.0.0.5          10.245.71.3       0x80000004   873  0xbf22  36
InterArPfx  0.0.0.6          10.245.71.3       0x80000002  1773  0x5c67  36
InterArPfx  0.0.0.7          10.245.71.3       0x80000002  1630  0xf088  44
InterArPfx  0.0.0.8          10.245.71.3       0x80000002  1488  0xd3d6  36
InterArPfx  0.0.0.9          10.245.71.3       0x80000002  1473  0xa3df  44
InterArPfx  0.0.0.1          10.245.71.5       0x80000004   293  0xd50f  36
InterArPfx  0.0.0.2          10.245.71.5       0x80000003   124  0xa547  36
InterArPfx  0.0.0.3          10.245.71.5       0x80000003   727  0x1cbb  36
InterArPfx  0.0.0.5          10.245.71.5       0x80000002  2695  0xdfcc  36
InterArPfx  0.0.0.6          10.245.71.5       0x80000002  2601  0xafd5  44
InterArPfx  0.0.0.7          10.245.71.5       0x80000002  2448  0x5f8a  36
InterArPfx  0.0.0.8          10.245.71.5       0x80000002  2302  0x427c  36
InterArPfx  0.0.0.9          10.245.71.5       0x80000002  2152  0xdcda  36
InterArPfx  0.0.0.10         10.245.71.5       0x80000002  2002  0x5929  44
InterArPfx  0.0.0.11         10.245.71.5       0x80000002  1852  0xc2af  44
InterArPfx  0.0.0.12         10.245.71.5       0x80000002   577  0x664   44
InterArPfx  0.0.0.1          10.245.71.11      0x80000003   275  0x6f7a  36
InterArPfx  0.0.0.2          10.245.71.11      0x80000003   140  0xa935  36
InterArPfx  0.0.0.3          10.245.71.11      0x80000003     5  0x427c  36
InterArPfx  0.0.0.4          10.245.71.11      0x80000002  2105  0xd89c  44
InterArPfx  0.0.0.5          10.245.71.11      0x80000002  1805  0x6b78  36
InterArPfx  0.0.0.6          10.245.71.11      0x80000002  1775  0xd6dd  36
InterArPfx  0.0.0.7          10.245.71.11      0x80000002  1640  0x532c  44
InterArPfx  0.0.0.8          10.245.71.11      0x80000002  1475  0xa9f7  36
InterArPfx  0.0.0.9          10.245.71.11      0x80000002  1340  0x7901  44
InterArRtr  0.0.0.1          10.245.71.5       0x80000002   434  0xc69c  32
IntraArPfx *0.0.0.1          10.245.71.1       0x80000005   274  0x3f8   76
```

```
IntraArPfx  0.0.0.1          10.245.71.3      0x80000005  1323  0x5cc1  64
IntraArPfx  0.0.0.1          10.245.71.4      0x80000004   275  0xba44  64
IntraArPfx  0.0.0.1          10.245.71.11     0x80000003  1505  0xd835  64
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Extern      0.0.0.1          10.245.71.4      0x80000003   874  0x8449  36
Extern      0.0.0.1          10.245.71.6      0x80000003   877  0xdc9e  36
    OSPF Link-Local link state database, interface at-2/0/0.0
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Link       *0.0.0.3          10.245.71.1      0x80000004   874  0x296b  56
Link        0.0.0.6          10.245.71.11     0x80000003   605  0xaf4f  56

user@router1>  show ospf3 route
Prefix                                  Path   Route       NH   Metric
                                        type   type        type
10.245.71.3                             Intra  Area BR     IP   13
  NH-interface at-2/0/0.0
10.245.71.4                             Intra  AS BR       IP   25
  NH-interface at-2/0/0.0
10.245.71.5                             Intra  Area BR     IP   14
  NH-interface at-2/0/0.0
10.245.71.6                             Inter  AS BR       IP   34
  NH-interface at-2/0/0.0
10.245.71.11                            Intra  Area BR     IP   12
  NH-interface at-2/0/0.0
9009:1::/64                             Intra  Network     IP   25
  NH-interface at-2/0/0.0
9009:1::2/128                           Intra  Network     IP   13
  NH-interface at-2/0/0.0
9009:2::/64                             Intra  Network     IP   12
  NH-interface at-2/0/0.0
9009:2::2/128                           Intra  Network     IP   12
  NH-interface at-2/0/0.0
9009:3::/64                             Inter  Network     IP   13
  NH-interface at-2/0/0.0
9009:4::/64                             Inter  Network     IP   22
  NH-interface at-2/0/0.0
9009:5::/64                             Inter  Network     IP   14
  NH-interface at-2/0/0.0
9009:6::/64                             Inter  Network     IP   34
  NH-interface at-2/0/0.0
9009:100::/64                           Ext2   Network     IP   0
  NH-interface at-2/0/0.0
9009:110::/64                           Intra  Network     IP   2
  NH-interface ge-1/1/0.0
9009:120::/64                           Inter  Network     IP   24
  NH-interface at-2/0/0.0
9009:130::/64                           Inter  Network     IP   16
  NH-interface at-2/0/0.0
9009:140::/64                           Inter  Network     IP   17
  NH-interface at-2/0/0.0
9009:150::/64                           Ext2   Network     IP   0
  NH-interface at-2/0/0.0
feee::10:255:71:1/128                   Intra  Network     IP   0
  NH-interface lo0.0
feee::10:255:71:3/128                   Inter  Network     IP   13
  NH-interface at-2/0/0.0
feee::10:255:71:4/128                   Intra  Network     IP   25
  NH-interface at-2/0/0.0
feee::10:255:71:5/128                   Inter  Network     IP   14
  NH-interface at-2/0/0.0
```

```
feee::10:255:71:6/128                       Inter  Network     IP   34
  NH-interface at-2/0/0.0
feee::10:255:71:11/128                      Inter  Network     IP   12
  NH-interface at-2/0/0.0

user@router1>  show interfaces terse
Interface       Admin Link Proto Local                 Remote
...
at-2/0/0        up    up
at-2/0/0.0      up    up   inet  10.19.2.1/24
                           inet6 9009:2::1/64
                                 fe80::2a0:a5ff:fe3d:dbf/64
...
lo0             up    up
lo0.0           up    up   inet  10.245.71.1         --> 0/0
                                 127.0.0.1           --> 0/0
                            inet6 fe80::2a0:a5ff:fe3d:dbf
                                 feee::10:255:71:1
...
```

Router 2 Status

```
user@router2> show ospf3 interface
Interface             State     Area            DR-ID           BDR-ID
Nbrs
at-0/3/1.0            PtToPt   0.0.0.0         0.0.0.0         0.0.0.0
1
vl -10.245.71.3        PtToPt   0.0.0.0         0.0.0.0         0.0.0.0
 1
at-0/3/0.0            PtToPt   0.0.0.1         0.0.0.0         0.0.0.0
0
fe-1/1/0.0            DR       0.0.0.1         10.245.71.11    10.245.71.5
1
lo0.0                 DRother  0.0.0.1         0.0.0.0         0.0.0.0
0
so-0/2/0.0            PtToPt   0.0.0.1         0.0.0.0         0.0.0.0
1

user@router2>  show ospf3 neighbor
ID              Interface             State     Pri   Dead
10.245.71.1     at-0/3/1.0            Full      128   36
  Neighbor-address fe80::2a0:a5ff:fe3d:dbf
10.245.71.3       vl -10.245.71.3         Full        0   33
  Neighbor-address 9009:3::2
10.245.71.5     fe-1/1/0.0            Full      128   36
  Neighbor-address fe80::290:69ff:fe98:909d
10.245.71.3     so-0/2/0.0            Full      128   33
  Neighbor-address fe80::201:afff:fe00:86ca

user@router2>  show ospf3 database
    OSPF link state database, area 0.0.0.0
 Type       ID              Adv Rtr           Seq      Age  Cksum  Len
Router      0.0.0.1         10.245.71.1      0x80000005   277  0x89ce  40
Router      0.0.0.1         10.245.71.3      0x80000006   875  0x2357  72
Router      0.0.0.1         10.245.71.4      0x80000004   275  0xc09c  40
Router      0.0.0.1         10.245.71.5      0x80000003  1407  0xf774  40
Router     *0.0.0.1         10.245.71.11     0x80000005   906  0x7f6b  56
InterArPfx  0.0.0.1         10.245.71.3      0x80000003   273  0x9f52  36
InterArPfx  0.0.0.2         10.245.71.3      0x80000003   131  0xb13d  36
```

```
InterArPfx  0.0.0.3          10.245.71.3      0x80000002  2225  0x1fa1  36
InterArPfx  0.0.0.4          10.245.71.3      0x80000002  2076  0x9bef  44
InterArPfx  0.0.0.5          10.245.71.3      0x80000004   574  0xbf22  36
InterArPfx  0.0.0.6          10.245.71.3      0x80000002  1474  0x5c67  36
InterArPfx  0.0.0.7          10.245.71.3      0x80000002  1331  0xf088  44
InterArPfx  0.0.0.8          10.245.71.3      0x80000002  1189  0xd3d6  36
InterArPfx  0.0.0.9          10.245.71.3      0x80000002  1174  0xa3df  44
InterArPfx  0.0.0.1          10.245.71.5      0x80000003  2923  0xd70e  36
InterArPfx  0.0.0.2          10.245.71.5      0x80000002  2537  0xa746  36
InterArPfx  0.0.0.3          10.245.71.5      0x80000003   428  0x1cbb  36
InterArPfx  0.0.0.5          10.245.71.5      0x80000002  2396  0xdfcc  36
InterArPfx  0.0.0.6          10.245.71.5      0x80000002  2302  0xafd5  44
InterArPfx  0.0.0.7          10.245.71.5      0x80000002  2149  0x5f8a  36
InterArPfx  0.0.0.8          10.245.71.5      0x80000002  2003  0x427c  36
InterArPfx  0.0.0.9          10.245.71.5      0x80000002  1853  0xdcda  36
InterArPfx  0.0.0.10         10.245.71.5      0x80000002  1703  0x5929  44
InterArPfx  0.0.0.11         10.245.71.5      0x80000002  1553  0xc2af  44
InterArPfx  0.0.0.12         10.245.71.5      0x80000002   278  0x664   44
InterArPfx *0.0.0.1          10.245.71.11     0x80000002  2108  0x7179  36
InterArPfx *0.0.0.2          10.245.71.11     0x80000002  2076  0xab34  36
InterArPfx *0.0.0.3          10.245.71.11     0x80000002  1941  0x447b  36
InterArPfx *0.0.0.4          10.245.71.11     0x80000002  1806  0xd89c  44
InterArPfx *0.0.0.5          10.245.71.11     0x80000002  1506  0x6b78  36
InterArPfx *0.0.0.6          10.245.71.11     0x80000002  1476  0xd6dd  36
InterArPfx *0.0.0.7          10.245.71.11     0x80000002  1341  0x532c  44
InterArPfx *0.0.0.8          10.245.71.11     0x80000002  1176  0xa9f7  36
InterArPfx *0.0.0.9          10.245.71.11     0x80000002  1041  0x7901  44
InterArRtr  0.0.0.1          10.245.71.5      0x80000002   135  0xc69c  32
IntraArPfx  0.0.0.1          10.245.71.1      0x80000004   877  0x5f7   76
IntraArPfx  0.0.0.1          10.245.71.3      0x80000005  1024  0x5cc1  64
IntraArPfx  0.0.0.1          10.245.71.4      0x80000003  1176  0xbc43  64
IntraArPfx *0.0.0.1          10.245.71.11     0x80000003  1206  0xd835  64
    OSPF link state database, area 0.0.0.1
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Router      0.0.0.1          10.245.71.3      0x80000006   574  0xad3f  56
Router      0.0.0.1          10.245.71.5      0x80000006   577  0xde02  56
Router     *0.0.0.1          10.245.71.11     0x80000007   576  0x8853  56
Network    *0.0.0.4          10.245.71.11     0x80000003   606  0xfd16  32
InterArPfx  0.0.0.1          10.245.71.3      0x80000002  1774  0xc722  36
InterArPfx  0.0.0.2          10.245.71.3      0x80000002  1624  0x7b2f  44
InterArPfx  0.0.0.3          10.245.71.3      0x80000002   874  0x877   44
InterArPfx  0.0.0.1          10.245.71.5      0x80000003   352  0x30a9  36
InterArPfx  0.0.0.3          10.245.71.5      0x80000002   205  0x6013  44
InterArPfx *0.0.0.1          10.245.71.11     0x80000003   141  0xa33c  36
InterArPfx *0.0.0.2          10.245.71.11     0x80000003     6  0x5749  44
InterArPfx *0.0.0.3          10.245.71.11     0x80000002  1776  0x6f5e  36
InterArPfx *0.0.0.4          10.245.71.11     0x80000002  1641  0x7ff9  44
InterArRtr  0.0.0.1          10.245.71.3      0x80000002   724  0x6609  32
InterArRtr  0.0.0.1          10.245.71.5      0x80000002    64  0xc69c  32
IntraArPfx  0.0.0.1          10.245.71.3      0x80000004   424  0x4a98  88
IntraArPfx  0.0.0.1          10.245.71.5      0x80000004   502  0x3691  76
IntraArPfx *0.0.0.1          10.245.71.11     0x80000005   441  0x2c5   76
IntraArPfx *0.0.0.5          10.245.71.11     0x80000003   741  0xfa59  44
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Extern      0.0.0.1          10.245.71.4      0x80000003   575  0x8449  36
Extern      0.0.0.1          10.245.71.6      0x80000003   578  0xdc9e  36
    OSPF Link-Local link state database, interface at-0/3/1.0
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Link        0.0.0.3          10.245.71.1      0x80000004   577  0x296b  56
Link       *0.0.0.6          10.245.71.11     0x80000003   306  0xaf4f  56
```

```
    OSPF Link-Local link state database, interface fe-1/1/0.0
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Link        0.0.0.5          10.245.71.5      0x80000003   727  0x40dc  56
Link       *0.0.0.4          10.245.71.11     0x80000004   876  0x73ab  56
    OSPF Link-Local link state database, interface so-0/2/0.0
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Link        0.0.0.4          10.245.71.3      0x80000003  2074  0x9d6   56
Link       *0.0.0.3          10.245.71.11     0x80000004   276  0xed12  56

user@router2>  show ospf3 route
Prefix                                     Path   Route       NH   Metric
                                           type   type        type
10.245.71.1                                Intra  Router      IP   12
  NH-interface at-0/3/1.0
10.245.71.3                                Intra  Area BR     IP   1
  NH-interface so-0/2/0.0
10.245.71.4                                Intra  AS BR       IP   13
  NH-interface so-0/2/0.0
10.245.71.5                                Intra  Area BR     IP   2
  NH-interface so-0/2/0.0
10.245.71.6                                Inter  AS BR       IP   22
  NH-interface so-0/2/0.0
10.245.71.11;0.0.0.4                       Intra  Transit     IP   10
  NH-interface fe-1/1/0.0
9009:1::/64                                Intra  Network     IP   13
  NH-interface so-0/2/0.0
9009:1::2/128                              Intra  Network     IP   1
  NH-interface so-0/2/0.0
9009:2::/64                                Intra  Network     IP   12
  NH-interface at-0/3/1.0
9009:2::2/128                              Intra  Network     IP   0
  NH-interface at-0/3/1.0
9009:3::/64                                Intra  Network     IP   1
  NH-interface so-0/2/0.0
9009:4::/64                                Intra  Network     IP   10
  NH-interface fe-1/1/0.0
9009:5::/64                                Intra  Network     IP   2
  NH-interface so-0/2/0.0
9009:6::/64                                Inter  Network     IP   22
  NH-interface so-0/2/0.0
9009:100::/64                              Ext2   Network     IP   0
  NH-interface so-0/2/0.0
9009:110::/64                              Intra  Network     IP   14
  NH-interface at-0/3/1.0
9009:120::/64                              Intra  Network     IP   12
  NH-interface at-0/3/0.0
9009:130::/64                              Intra  Network     IP   4
  NH-interface so-0/2/0.0
9009:140::/64                              Intra  Network     IP   5
  NH-interface so-0/2/0.0
9009:150::/64                              Ext2   Network     IP   0
  NH-interface so-0/2/0.0
feee::10:255:71:1/128                      Intra  Network     IP   12
  NH-interface at-0/3/1.0
feee::10:255:71:3/128                      Intra  Network     IP   1
  NH-interface so-0/2/0.0
feee::10:255:71:4/128                      Intra  Network     IP   13
  NH-interface so-0/2/0.0
feee::10:255:71:5/128                      Intra  Network     IP   2
  NH-interface so-0/2/0.0
feee::10:255:71:6/128                      Inter  Network     IP   22
```

```
  NH-interface so-0/2/0.0
feee::10:255:71:11/128                      Intra  Network     IP   0
  NH-interface lo0.0

user@router2>  show interfaces terse
Interface       Admin Link Proto Local                 Remote
...
so-0/2/0        up    up
so-0/2/0.0      up    up   inet  10.19.3.1/24
                           inet6 9009:3::1/64
                                 fe80::2a0:a5ff:fe3d:56/64
...
at-0/3/1        up    up
at-0/3/1.0      up    up   inet  10.19.2.2/24
                           inet6 9009:2::2/64
                                 fe80::2a0:a5ff:fe3d:56/64
...
fe-1/1/0        up    up
fe-1/1/0.0      up    up   inet  10.19.4.1/24
                           inet6 9009:4::1/64
                                 fe80::290:69ff:fea0:809d/64
...
lo0             up    up
lo0.0           up    up   inet  10.245.71.11        --> 0/0
                                 127.0.0.1           --> 0/0
                           inet6 fe80::2a0:a5ff:fe3d:56
                                 feee::10:255:71:11
...
```

Router 3 Status

```
user@router3> show ospf3 interface
Interface             State     Area            DR-ID           BDR-ID
Nbrs
so-0/3/2.0            PtToPt    0.0.0.0         0.0.0.0         0.0.0.0
1
vl-10.245.71.11       PtToPt    0.0.0.0         0.0.0.0         0.0.0.0
1
vl-10.245.71.5        PtToPt    0.0.0.0         0.0.0.0         0.0.0.0
1
at-1/2/0.0            PtToPt    0.0.0.1         0.0.0.0         0.0.0.0
0
lo0.0                 DRother   0.0.0.1         0.0.0.0         0.0.0.0
0
so-0/3/0.0            PtToPt    0.0.0.1         0.0.0.0         0.0.0.0
1
t1-0/2/1.0            PtToPt    0.0.0.1         0.0.0.0         0.0.0.0
1

user@router3> show ospf3 neighbor
ID              Interface             State     Pri   Dead
10.245.71.4     so-0/3/2.0            Full      128   38
  Neighbor-address fe80::201:afff:fe03:6fa1
10.245.71.11    vl-10.245.71.11       Full        0   36
  Neighbor-address 9009:3::1
10.245.71.5     vl-10.245.71.5        Full        0   35
  Neighbor-address 9009:5::2
10.245.71.11    so-0/3/0.0            Full      128   37
  Neighbor-address fe80::2a0:a5ff:fe3d:56
```

```
10.245.71.5      t1-0/2/1.0             Full      128   39
  Neighbor-address fe80::2a0:a5ff:fe3d:b63

user@router3>  show ospf3 database
    OSPF link state database, area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Router      0.0.0.1          10.245.71.1       0x80000005    94  0x89ce  40
Router     *0.0.0.1          10.245.71.3       0x80000006   690  0x2357  72
Router      0.0.0.1          10.245.71.4       0x80000004    90  0xc09c  40
Router      0.0.0.1          10.245.71.5       0x80000003  1222  0xf774  40
Router      0.0.0.1          10.245.71.11      0x80000005   723  0x7f6b  56
InterArPfx *0.0.0.1          10.245.71.3       0x80000003    88  0x9f52  36
InterArPfx *0.0.0.2          10.245.71.3       0x80000002  2188  0xb33c  36
InterArPfx *0.0.0.3          10.245.71.3       0x80000002  2040  0x1fa1  36
InterArPfx *0.0.0.4          10.245.71.3       0x80000002  1891  0x9bef  44
InterArPfx *0.0.0.5          10.245.71.3       0x80000004   388  0xbf22  36
InterArPfx *0.0.0.6          10.245.71.3       0x80000002  1288  0x5c67  36
InterArPfx *0.0.0.7          10.245.71.3       0x80000002  1146  0xf088  44
InterArPfx *0.0.0.8          10.245.71.3       0x80000002  1003  0xd3d6  36
InterArPfx *0.0.0.9          10.245.71.3       0x80000002   988  0xa3df  44
InterArPfx  0.0.0.1          10.245.71.5       0x80000003  2738  0xd70e  36
InterArPfx  0.0.0.2          10.245.71.5       0x80000002  2352  0xa746  36
InterArPfx  0.0.0.3          10.245.71.5       0x80000003   243  0x1cbb  36
InterArPfx  0.0.0.5          10.245.71.5       0x80000002  2211  0xdfcc  36
InterArPfx  0.0.0.6          10.245.71.5       0x80000002  2117  0xafd5  44
InterArPfx  0.0.0.7          10.245.71.5       0x80000002  1964  0x5f8a  36
InterArPfx  0.0.0.8          10.245.71.5       0x80000002  1818  0x427c  36
InterArPfx  0.0.0.9          10.245.71.5       0x80000002  1668  0xdcda  36
InterArPfx  0.0.0.10         10.245.71.5       0x80000002  1518  0x5929  44
InterArPfx  0.0.0.11         10.245.71.5       0x80000002  1368  0xc2af  44
InterArPfx  0.0.0.12         10.245.71.5       0x80000002    93  0x664   44
InterArPfx  0.0.0.1          10.245.71.11      0x80000002  1925  0x7179  36
InterArPfx  0.0.0.2          10.245.71.11      0x80000002  1893  0xab34  36
InterArPfx  0.0.0.3          10.245.71.11      0x80000002  1758  0x447b  36
InterArPfx  0.0.0.4          10.245.71.11      0x80000002  1623  0xd89c  44
InterArPfx  0.0.0.5          10.245.71.11      0x80000002  1323  0x6b78  36
InterArPfx  0.0.0.6          10.245.71.11      0x80000002  1293  0xd6dd  36
InterArPfx  0.0.0.7          10.245.71.11      0x80000002  1158  0x532c  44
InterArPfx  0.0.0.8          10.245.71.11      0x80000002   993  0xa9f7  36
InterArPfx  0.0.0.9          10.245.71.11      0x80000002   858  0x7901  44
InterArRtr  0.0.0.1          10.245.71.5       0x80000001  2743  0xc89b  32
IntraArPfx  0.0.0.1          10.245.71.1       0x80000004   694  0x5f7   76
IntraArPfx *0.0.0.1          10.245.71.3       0x80000005   839  0x5cc1  64
IntraArPfx  0.0.0.1          10.245.71.4       0x80000003   990  0xbc43  64
IntraArPfx  0.0.0.1          10.245.71.11      0x80000003  1023  0xd835  64
    OSPF link state database, area 0.0.0.1
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Router     *0.0.0.1          10.245.71.3       0x80000006   389  0xad3f  56
Router      0.0.0.1          10.245.71.5       0x80000006   393  0xde02  56
Router      0.0.0.1          10.245.71.11      0x80000007   393  0x8853  56
Network     0.0.0.4          10.245.71.11      0x80000003   423  0xfd16  32
InterArPfx *0.0.0.1          10.245.71.3       0x80000002  1588  0xc722  36
InterArPfx *0.0.0.2          10.245.71.3       0x80000002  1438  0x7b2f  44
InterArPfx *0.0.0.3          10.245.71.3       0x80000002   688  0x877   44
InterArPfx  0.0.0.1          10.245.71.5       0x80000003   168  0x30a9  36
InterArPfx  0.0.0.3          10.245.71.5       0x80000002    21  0x6013  44
InterArPfx  0.0.0.1          10.245.71.11      0x80000002  2193  0xa53b  36
InterArPfx  0.0.0.2          10.245.71.11      0x80000002  2059  0x5948  44
InterArPfx  0.0.0.3          10.245.71.11      0x80000002  1593  0x6f5e  36
InterArPfx  0.0.0.4          10.245.71.11      0x80000002  1458  0x7ff9  44
InterArRtr *0.0.0.1          10.245.71.3       0x80000002   538  0x6609  32
```

```
InterArRtr  0.0.0.1          10.245.71.5      0x80000001  2743  0xc89b  32
IntraArPfx *0.0.0.1          10.245.71.3      0x80000004   238  0x4a98  88
IntraArPfx  0.0.0.1          10.245.71.5      0x80000004   318  0x3691  76
IntraArPfx  0.0.0.1          10.245.71.11     0x80000005   258  0x2c5   76
IntraArPfx  0.0.0.5          10.245.71.11     0x80000003   558  0xfa59  44
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Extern      0.0.0.1          10.245.71.4      0x80000003   390  0x8449  36
Extern      0.0.0.1          10.245.71.6      0x80000003   394  0xdc9e  36
    OSPF Link-Local link state database, interface so-0/3/0.0
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Link       *0.0.0.4          10.245.71.3      0x80000003  1888  0x9d6   56
Link        0.0.0.3          10.245.71.11     0x80000004    93  0xed12  56
    OSPF Link-Local link state database, interface so-0/3/2.0
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Link       *0.0.0.6          10.245.71.3      0x80000003  1589  0xb02f  56
Link        0.0.0.2          10.245.71.4      0x80000003   690  0xa07f  56
    OSPF Link-Local link state database, interface t1-0/2/1.0
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Link       *0.0.0.5          10.245.71.3      0x80000003  1738  0x4399  56
Link        0.0.0.3          10.245.71.5      0x80000002  2423  0x618c  56

user@router3>  show ospf3 route
Prefix                                  Path   Route       NH   Metric
                                        type   type        type
10.245.71.1                             Intra  Router      IP   13
  NH-interface (null), NH-addr feee::10:255:71:11
10.245.71.4                             Intra  AS BR       IP   12
  NH-interface so-0/3/2.0
10.245.71.5                             Intra  Area BR     IP   1
  NH-interface t1-0/2/1.0
10.245.71.6                             Inter  AS BR       IP   21
  NH-interface t1-0/2/1.0
10.245.71.11                            Intra  Area BR     IP   1
  NH-interface so-0/3/0.0
10.245.71.11;0.0.0.4                    Intra  Transit     IP   11
  NH-interface so-0/3/0.0
  NH-interface t1-0/2/1.0
9009:1::/64                             Intra  Network     IP   12
  NH-interface so-0/3/2.0
9009:1::2/128                           Intra  Network     IP   0
  NH-interface so-0/3/2.0
9009:2::/64                             Intra  Network     IP   13
  NH-interface so-0/3/0.0
9009:2::2/128                           Intra  Network     IP   1
  NH-interface so-0/3/0.0
9009:3::/64                             Intra  Network     IP   1
  NH-interface so-0/3/0.0
9009:4::/64                             Intra  Network     IP   11
  NH-interface so-0/3/0.0
  NH-interface t1-0/2/1.0
9009:5::/64                             Intra  Network     IP   1
  NH-interface t1-0/2/1.0
9009:6::/64                             Inter  Network     IP   21
  NH-interface t1-0/2/1.0
9009:100::/64                           Ext2   Network     IP   0
  NH-interface so-0/3/2.0
9009:110::/64                           Intra  Network     IP   15
  NH-interface so-0/3/0.0
9009:120::/64                           Intra  Network     IP   13
  NH-interface so-0/3/0.0
```

```
9009:130::/64                              Intra  Network     IP   3
  NH-interface at-1/2/0.0
9009:140::/64                              Intra  Network     IP   4
  NH-interface t1-0/2/1.0
9009:150::/64                              Ext2   Network     IP   0
  NH-interface t1-0/2/1.0
feee::10:255:71:1/128                      Intra  Network     IP   13
  NH-interface so-0/3/0.0
feee::10:255:71:3/128                      Intra  Network     IP   0
  NH-interface lo0.0
feee::10:255:71:4/128                      Intra  Network     IP   12
  NH-interface so-0/3/2.0
feee::10:255:71:5/128                      Intra  Network     IP   1
  NH-interface t1-0/2/1.0
feee::10:255:71:6/128                      Inter  Network     IP   21
  NH-interface t1-0/2/1.0
feee::10:255:71:11/128                     Intra  Network     IP   1
  NH-interface so-0/3/0.0

user@router3>  show interfaces terse
Interface       Admin Link Proto Local                 Remote
...
t1-0/2/1.0      up    up   inet  10.19.5.1/24
                           inet6 9009:5::1/64
                                 fe80::201:afff:fe00:86ca/64
...
so-0/3/0        up    up
so-0/3/0.0      up    up   inet  10.19.3.2/24
                           inet6 9009:3::2/64
                                 fe80::201:afff:fe00:86ca/64
so-0/3/1        up    up
so-0/3/2        up    up
so-0/3/2.0      up    up   inet  10.19.1.2/24
                           inet6 9009:1::2/64
                                 fe80::201:afff:fe00:86ca/64
...
lo0             up    up
lo0.0           up    up   inet  10.245.71.3         --> 0/0
                                 127.0.0.1           --> 0/0
                           inet6 fe80::201:afff:fe00:86ca
                                 feee::10:255:71:3
...
```

To provide a comparison between OSPFv3 show commands and legacy OSPFv2 show commands, the following is some sample output of the OSPFv2 connection between Routers 0 and 3:

```
user@router3> show ospf interface
Interface             State     Area            DR ID           BDR ID
Nbrs
lo0.0                 DRother  0.0.0.0         0.0.0.0         0.0.0.0
0
lo0.0                 DRother  0.0.0.0         0.0.0.0         0.0.0.0
0
so-0/3/2.0            PtToPt   0.0.0.0         0.0.0.0         0.0.0.0
1

user@router3> show ospf neighbor
  Address         Interface             State      ID              Pri  Dead
10.19.1.1        so-0/3/2.0            Full      10.245.71.4      128   38
```

```
user@router3> show ospf database
    OSPF link state database, area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router  *10.245.71.3      10.245.71.3      0x80000002    67  0x2  0x5c45  60
Router   10.245.71.4      10.245.71.4      0x80000002    74  0x2  0x267a  60

user@router3> show ospf route
Prefix               Path   Route       NH   Metric  NextHop       Nexthop

                     Type   Type        Type         Interface     addr/label
10.245.71.4          Intra  Router      IP   1       so-0/3/2.0
10.19.1.0/24         Intra  Network     IP   1       so-0/3/2.0
10.245.71.3/32       Intra  Network     IP   0       lo0.0
10.245.71.4/32       Intra  Network     IP   1       so-0/3/2.0
```

Router 4 Status

```
user@router4> show ospf3 interface
Interface              State     Area            DR-ID           BDR-ID
Nbrs
vl- 10.245.71.3         PtToPt   0.0.0.0         0.0.0.0         0.0.0.0
 1
at-0/3/0.0             PtToPt   0.0.0.1         0.0.0.0         0.0.0.0
0
fe-1/1/0.0             BDR      0.0.0.1         10.245.71.11    10.245.71.5
1
lo0.0                  DRother  0.0.0.1         0.0.0.0         0.0.0.0
0
t1-0/2/1.0             PtToPt   0.0.0.1         0.0.0.0         0.0.0.0
1
fe-0/0/0.0             BDR      0.0.0.2         10.245.71.6     10.245.71.5
1

user@router4>  show ospf3 neighbor
ID               Interface              State     Pri   Dead
10.245.71.3       vl -10.245.71.3         Full        0   32
  Neighbor-address 9009:5::1
10.245.71.11     fe-1/1/0.0             Full      128   37
  Neighbor-address fe80::290:69ff:fea0:809d
10.245.71.3      t1-0/2/1.0             Full      128   32
  Neighbor-address fe80::201:afff:fe00:86ca
10.245.71.6      fe-0/0/0.0             Full      128   35
  Neighbor-address fe80::290:69ff:fe94:c400

user@router4>  show ospf3 database
    OSPF link state database, area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Router      0.0.0.1          10.245.71.1      0x80000004   894  0x8bcd  40
Router      0.0.0.1          10.245.71.3      0x80000006   590  0x2357  72
Router      0.0.0.1          10.245.71.4      0x80000003  1190  0xc29b  40
Router     *0.0.0.1          10.245.71.5      0x80000003  1120  0xf774  40
Router      0.0.0.1          10.245.71.11     0x80000005   623  0x7f6b  56
InterArPfx  0.0.0.1          10.245.71.3      0x80000002  2114  0xa151  36
InterArPfx  0.0.0.2          10.245.71.3      0x80000002  2089  0xb33c  36
InterArPfx  0.0.0.3          10.245.71.3      0x80000002  1940  0x1fa1  36
InterArPfx  0.0.0.4          10.245.71.3      0x80000002  1791  0x9bef  44
InterArPfx  0.0.0.5          10.245.71.3      0x80000004   289  0xbf22  36
InterArPfx  0.0.0.6          10.245.71.3      0x80000002  1188  0x5c67  36
```

```
InterArPfx  0.0.0.7          10.245.71.3      0x80000002  1046  0xf088  44
InterArPfx  0.0.0.8          10.245.71.3      0x80000002   904  0xd3d6  36
InterArPfx  0.0.0.9          10.245.71.3      0x80000002   888  0xa3df  44
InterArPfx *0.0.0.1          10.245.71.5      0x80000003  2636  0xd70e  36
InterArPfx *0.0.0.2          10.245.71.5      0x80000002  2250  0xa746  36
InterArPfx *0.0.0.3          10.245.71.5      0x80000003   141  0x1cbb  36
InterArPfx *0.0.0.5          10.245.71.5      0x80000002  2109  0xdfcc  36
InterArPfx *0.0.0.6          10.245.71.5      0x80000002  2015  0xafd5  44
InterArPfx *0.0.0.7          10.245.71.5      0x80000002  1862  0x5f8a  36
InterArPfx *0.0.0.8          10.245.71.5      0x80000002  1716  0x427c  36
InterArPfx *0.0.0.9          10.245.71.5      0x80000002  1566  0xdcda  36
InterArPfx *0.0.0.10         10.245.71.5      0x80000002  1416  0x5929  44
InterArPfx *0.0.0.11         10.245.71.5      0x80000002  1266  0xc2af  44
InterArPfx *0.0.0.12         10.245.71.5      0x80000001  2641  0x863   44
InterArPfx  0.0.0.1          10.245.71.11     0x80000002  1825  0x7179  36
InterArPfx  0.0.0.2          10.245.71.11     0x80000002  1793  0xab34  36
InterArPfx  0.0.0.3          10.245.71.11     0x80000002  1658  0x447b  36
InterArPfx  0.0.0.4          10.245.71.11     0x80000002  1523  0xd89c  44
InterArPfx  0.0.0.5          10.245.71.11     0x80000002  1223  0x6b78  36
InterArPfx  0.0.0.6          10.245.71.11     0x80000002  1193  0xd6dd  36
InterArPfx  0.0.0.7          10.245.71.11     0x80000002  1058  0x532c  44
InterArPfx  0.0.0.8          10.245.71.11     0x80000002   893  0xa9f7  36
InterArPfx  0.0.0.9          10.245.71.11     0x80000002   758  0x7901  44
InterArRtr *0.0.0.1          10.245.71.5      0x80000001  2641  0xc89b  32
IntraArPfx  0.0.0.1          10.245.71.1      0x80000004   594  0x5f7   76
IntraArPfx  0.0.0.1          10.245.71.3      0x80000005   739  0x5cc1  64
IntraArPfx  0.0.0.1          10.245.71.4      0x80000003   890  0xbc43  64
IntraArPfx  0.0.0.1          10.245.71.11     0x80000003   923  0xd835  64
    OSPF link state database, area 0.0.0.1
 Type       ID               Adv Rtr           Seq       Age  Cksum  Len
Router      0.0.0.1          10.245.71.3      0x80000006   289  0xad3f  56
Router     *0.0.0.1          10.245.71.5      0x80000006   291  0xde02  56
Router      0.0.0.1          10.245.71.11     0x80000007   292  0x8853  56
Network     0.0.0.4          10.245.71.11     0x80000003   322  0xfd16  32
InterArPfx  0.0.0.1          10.245.71.3      0x80000002  1488  0xc722  36
InterArPfx  0.0.0.2          10.245.71.3      0x80000002  1339  0x7b2f  44
InterArPfx  0.0.0.3          10.245.71.3      0x80000002   589  0x877   44
InterArPfx *0.0.0.1          10.245.71.5      0x80000003    66  0x30a9  36
InterArPfx *0.0.0.3          10.245.71.5      0x80000001  2641  0x6212  44
InterArPfx  0.0.0.1          10.245.71.11     0x80000002  2092  0xa53b  36
InterArPfx  0.0.0.2          10.245.71.11     0x80000002  1958  0x5948  44
InterArPfx  0.0.0.3          10.245.71.11     0x80000002  1492  0x6f5e  36
InterArPfx  0.0.0.4          10.245.71.11     0x80000002  1357  0x7ff9  44
InterArRtr  0.0.0.1          10.245.71.3      0x80000002   439  0x6609  32
InterArRtr *0.0.0.1          10.245.71.5      0x80000001  2641  0xc89b  32
IntraArPfx  0.0.0.1          10.245.71.3      0x80000004   139  0x4a98  88
IntraArPfx *0.0.0.1          10.245.71.5      0x80000004   216  0x3691  76
IntraArPfx  0.0.0.1          10.245.71.11     0x80000005   157  0x2c5   76
IntraArPfx  0.0.0.5          10.245.71.11     0x80000003   457  0xfa59  44
    OSPF link state database, area 0.0.0.2
 Type       ID               Adv Rtr           Seq       Age  Cksum  Len
Router     *0.0.0.1          10.245.71.5      0x80000004   366  0x252e  40
Router      0.0.0.1          10.245.71.6      0x80000004  1492  0x64d   40
Network     0.0.0.2          10.245.71.6      0x80000003   892  0xfd22  32
InterArPfx *0.0.0.1          10.245.71.5      0x80000003  2636  0xd70e  36
InterArPfx *0.0.0.2          10.245.71.5      0x80000002  2179  0xa746  36
InterArPfx *0.0.0.3          10.245.71.5      0x80000002  2091  0xf3ba  36
InterArPfx *0.0.0.4          10.245.71.5      0x80000002  1938  0xc3c3  44
InterArPfx *0.0.0.5          10.245.71.5      0x80000002  1791  0x7378  36
InterArPfx *0.0.0.6          10.245.71.5      0x80000002  1641  0x566a  36
InterArPfx *0.0.0.7          10.245.71.5      0x80000002  1491  0xf0c8  36
```

```
InterArPfx *0.0.0.8          10.245.71.5      0x80000002  1341  0x6d17  44
InterArPfx *0.0.0.9          10.245.71.5      0x80000002  1191  0xd69d  44
InterArPfx *0.0.0.10         10.245.71.5      0x80000002  1049  0x6776  36
InterArPfx *0.0.0.11         10.245.71.5      0x80000002   979  0x1b83  44
InterArPfx *0.0.0.12         10.245.71.5      0x80000002   908  0x6772  36
InterArPfx *0.0.0.13         10.245.71.5      0x80000002   891  0x1b7f  44
InterArPfx *0.0.0.14         10.245.71.5      0x80000002   815  0x3195  36
InterArPfx *0.0.0.15         10.245.71.5      0x80000002   738  0x4131  44
InterArPfx *0.0.0.16         10.245.71.5      0x80000002   662  0x7fef  44
InterArRtr *0.0.0.1          10.245.71.5      0x80000002   591  0x6408  32
IntraArPfx  0.0.0.1          10.245.71.6      0x80000005  1192  0x42b9  52
IntraArPfx  0.0.0.3          10.245.71.6      0x80000003   592  0xfe61  44
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Extern      0.0.0.1          10.245.71.4      0x80000003   290  0x8449  36
Extern      0.0.0.1          10.245.71.6      0x80000003   292  0xdc9e  36
    OSPF Link-Local link state database, interface fe-0/0/0.0
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Link       *0.0.0.4          10.245.71.5      0x80000003   516  0x3b6   56
Link        0.0.0.2          10.245.71.6      0x80000004  1792  0x782   56
    OSPF Link-Local link state database, interface fe-1/1/0.0
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Link       *0.0.0.5          10.245.71.5      0x80000003   441  0x40dc  56
Link        0.0.0.4          10.245.71.11     0x80000004   592  0x73ab  56
    OSPF Link-Local link state database, interface t1-0/2/1.0
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Link        0.0.0.5          10.245.71.3      0x80000003  1639  0x4399  56
Link       *0.0.0.3          10.245.71.5      0x80000002  2321  0x618c  56

user@router4>  show ospf3 route
Prefix                                  Path   Route       NH   Metric
                                        type   type        type
10.245.71.1                             Intra  Router      IP   14
  NH-interface (null), NH-addr feee::10:255:71:3
10.245.71.3                             Intra  Area BR     IP   1
  NH-interface t1-0/2/1.0
10.245.71.4                             Intra  AS BR       IP   13
  NH-interface t1-0/2/1.0
10.245.71.6                             Intra  AS BR       IP   20
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe94:c400
10.245.71.6;0.0.0.2                     Intra  Transit     IP   20
  NH-interface fe-0/0/0.0
10.245.71.11                            Intra  Area BR     IP   2
  NH-interface t1-0/2/1.0
10.245.71.11;0.0.0.4                    Intra  Transit     IP   10
  NH-interface fe-1/1/0.0
9009:1::/64                             Intra  Network     IP   13
  NH-interface t1-0/2/1.0
9009:1::2/128                           Intra  Network     IP   1
  NH-interface t1-0/2/1.0
9009:2::/64                             Intra  Network     IP   14
  NH-interface t1-0/2/1.0
9009:2::2/128                           Intra  Network     IP   2
  NH-interface t1-0/2/1.0
9009:3::/64                             Intra  Network     IP   2
  NH-interface t1-0/2/1.0
9009:4::/64                             Intra  Network     IP   10
  NH-interface fe-1/1/0.0
9009:5::/64                             Intra  Network     IP   1
  NH-interface t1-0/2/1.0
9009:6::/64                             Intra  Network     IP   20
```

```
  NH-interface fe-0/0/0.0
9009:100::/64                                Ext2   Network     IP   0
  NH-interface t1-0/2/1.0
9009:110::/64                                Intra  Network     IP   16
  NH-interface t1-0/2/1.0
9009:120::/64                                Intra  Network     IP   14
  NH-interface t1-0/2/1.0
9009:130::/64                                Intra  Network     IP   4
  NH-interface t1-0/2/1.0
9009:140::/64                                Intra  Network     IP   3
  NH-interface at-0/3/0.0
9009:150::/64                                Ext2   Network     IP   0
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe94:c400
feee::10:255:71:1/128                        Intra  Network     IP   14
  NH-interface t1-0/2/1.0
feee::10:255:71:3/128                        Intra  Network     IP   1
  NH-interface t1-0/2/1.0
feee::10:255:71:4/128                        Intra  Network     IP   13
  NH-interface t1-0/2/1.0
feee::10:255:71:5/128                        Intra  Network     IP   0
  NH-interface lo0.0
feee::10:255:71:6/128                        Intra  Network     IP   20
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe94:c400
feee::10:255:71:11/128                       Intra  Network     IP   2
  NH-interface t1-0/2/1.0

user@router4>  show interfaces terse
Interface       Admin Link Proto Local                 Remote
fe-0/0/0        up    up
fe-0/0/0.0      up    up   inet  10.19.6.1/24
                           inet6 9009:6::1/64
                                 fe80::290:69ff:fe98:9000/64
...
t1-0/2/1        up    up
t1-0/2/1.0      up    up   inet  10.19.5.2/24
                           inet6 9009:5::2/64
                                 fe80::2a0:a5ff:fe3d:b63/64
...
fe-1/1/0        up    up
fe-1/1/0.0      up    up   inet  10.19.4.2/24
                           inet6 9009:4::2/64
                                 fe80::290:69ff:fe98:909d/64
...
lo0             up    up
lo0.0           up    up   inet  10.245.71.5         --> 0/0
                                 127.0.0.1           --> 0/0
                           inet6 fe80::2a0:a5ff:fe3d:b63
                                 feee::10:255:71:5
...
```

Router 5 Status

```
user@router5> show ospf3 interface
Interface             State     Area            DR-ID           BDR-ID
Nbrs
fe-0/0/0.0            DR       0.0.0.2         10.245.71.6     10.245.71.5
1
lo0.0                 DRother  0.0.0.2         0.0.0.0         0.0.0.0
0
```

```
user@router5> show ospf3 neighbor
ID              Interface             State     Pri   Dead
10.245.71.5     fe-0/0/0.0            Full      128   33
  Neighbor-address fe80::290:69ff:fe98:9000

user@router5>  show ospf3 database
    OSPF link state database, area 0.0.0.2
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Router      0.0.0.1          10.245.71.5      0x80000003  2237  0x272d  40
Router     *0.0.0.1          10.245.71.6      0x80000004  1082  0x64d   40
Network    *0.0.0.2          10.245.71.6      0x80000003   482  0xfd22  32
InterArPfx  0.0.0.1          10.245.71.5      0x80000003  2228  0xd70e  36
InterArPfx  0.0.0.2          10.245.71.5      0x80000002  1771  0xa746  36
InterArPfx  0.0.0.3          10.245.71.5      0x80000002  1683  0xf3ba  36
InterArPfx  0.0.0.4          10.245.71.5      0x80000002  1530  0xc3c3  44
InterArPfx  0.0.0.5          10.245.71.5      0x80000002  1383  0x7378  36
InterArPfx  0.0.0.6          10.245.71.5      0x80000002  1233  0x566a  36
InterArPfx  0.0.0.7          10.245.71.5      0x80000002  1083  0xf0c8  36
InterArPfx  0.0.0.8          10.245.71.5      0x80000002   933  0x6d17  44
InterArPfx  0.0.0.9          10.245.71.5      0x80000002   783  0xd69d  44
InterArPfx  0.0.0.10         10.245.71.5      0x80000002   641  0x6776  36
InterArPfx  0.0.0.11         10.245.71.5      0x80000002   570  0x1b83  44
InterArPfx  0.0.0.12         10.245.71.5      0x80000002   500  0x6772  36
InterArPfx  0.0.0.13         10.245.71.5      0x80000002   483  0x1b7f  44
InterArPfx  0.0.0.14         10.245.71.5      0x80000002   406  0x3195  36
InterArPfx  0.0.0.15         10.245.71.5      0x80000002   330  0x4131  44
InterArPfx  0.0.0.16         10.245.71.5      0x80000002   253  0x7fef  44
InterArRtr  0.0.0.1          10.245.71.5      0x80000002   183  0x6408  32
IntraArPfx *0.0.0.1          10.245.71.6      0x80000005   782  0x42b9  52
IntraArPfx *0.0.0.3          10.245.71.6      0x80000003   182  0xfe61  44
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Extern      0.0.0.1          10.245.71.4      0x80000002  1082  0x8648  36
Extern     *0.0.0.1          10.245.71.6      0x80000002  1682  0xde9d  36
    OSPF Link-Local link state database, interface fe-0/0/0.0
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Link        0.0.0.4          10.245.71.5      0x80000003   108  0x3b6   56
Link       *0.0.0.2          10.245.71.6      0x80000004  1382  0x782   56
user@router5>  show ospf3 route
Prefix                                  Path   Route       NH   Metric
                                        type   type        type
10.245.71.4                             Inter  AS BR       IP   33
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
10.245.71.5                             Intra  Area BR     IP   20
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
10.245.71.6;0.0.0.2                     Intra  Transit     IP   20
  NH-interface fe-0/0/0.0
9009:1::/64                             Inter  Network     IP   33
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
9009:1::2/128                           Inter  Network     IP   21
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
9009:2::/64                             Inter  Network     IP   34
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
9009:2::2/128                           Inter  Network     IP   22
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
9009:3::/64                             Inter  Network     IP   22
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
9009:4::/64                             Inter  Network     IP   30
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
9009:5::/64                             Inter  Network     IP   21
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
```

```
9009:6::/64                                Intra  Network     IP   20
  NH-interface fe-0/0/0.0
9009:100::/64                              Ext2   Network     IP   0
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
9009:110::/64                              Inter  Network     IP   36
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
9009:120::/64                              Inter  Network     IP   34
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
9009:130::/64                              Inter  Network     IP   24
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
9009:140::/64                              Inter  Network     IP   23
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
feee::10:255:71:1/128                      Inter  Network     IP   34
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
feee::10:255:71:3/128                      Inter  Network     IP   21
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
feee::10:255:71:4/128                      Inter  Network     IP   33
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
feee::10:255:71:5/128                      Inter  Network     IP   20
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
feee::10:255:71:6/128                      Intra  Network     IP   0
  NH-interface lo0.0
feee::10:255:71:11/128                     Inter  Network     IP   22
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000

user@router5>  show interfaces terse
Interface       Admin Link Proto Local                 Remote
...
fe-0/0/0        up    up
fe-0/0/0.0      up    up   inet  10.19.6.2/24
                           inet6 9009:6::2/64
                                 fe80::290:69ff:fe94:c400/64
...
lo0             up    up
lo0.0           up    up   inet  10.245.71.6         --> 0/0
                                 127.0.0.1           --> 0/0
                           inet6 fe80::2a0:a5ff:fe12:33a2
                                 feee::10:255:71:6
...
```

For More Information

For additional information about OSPFv3 for IPv6, see the following resources:

- *JUNOS Routing Protocols Configuration Guide*
- RFC 2328, *OSPF Version 2*
- RFC 2460, *Internet Protocol, Version 6 (IPv6) Specification*
- RFC 2740, *OSPF for IPv6*
- RFC 3513, *IP Version 6 Addressing Architecture*

Revision History

5 October 2009—10.0R1 Release. Roy Spencer.

13 June 2009—9.6R1 Release. Roy Spencer.

13 April 2009—9.5R1 Release. Roy Spencer.

15 January 2009—9.4R1 Release. Roy Spencer.

10 October 2008—9.3R1 Release. Roy Spencer.

10 July 2008—9.2R1 Release. Roy Spencer.

10 April 2008—9.1R1 Release. Roy Spencer.

1 February 2008—9.0R1 Release. Fawn Damitio.

17 August 2007—Added support for logical router administrators. 8.5 beta release. Fawn Damitio.

29 June 2007—8.4R1 Release. Fawn Damitio.

27 March 2007—8.3R1 Release. Fawn Damitio.

12 January 2007—Added support for MX960 Ethernet Services Routers. 8.2R1 Release. Fawn Damitio.

15 September 2006—8.1R1 Release. Richard Hendricks.

29 June 2006—8.0R1 Release. Richard Hendricks.

27 March 2006—7.6R1 Release. Richard Hendricks.

9 January 2006—7.5R1 Release. Richard Hendricks.

14 September 2005—7.4R1 Release. Richard Hendricks.

13 June 2005—7.3R1 Release. Richard Hendricks.

5 April 2005—Added support for the J Series Services Routers and included a cross-reference to the IPsec chapter, 7.2R1 Release. Richard Hendricks.

2 February 2005—7.1R1 Release. Richard Hendricks.

6 October 2004—7.0R1 Release. Richard Hendricks.

6 July 2004—6.4R1 Release. Richard Hendricks.

5 April 2004—6.3R1 Release. Richard Hendricks.

22 December 2003—6.2R1 Release. Richard Hendricks.

22 September 2003—6.1R1 Release. Richard Hendricks.

30 June 2003—6.0R1 Release. Richard Hendricks.

2 April 2003—5.7R1 Release. Richard Hendricks.

27 December 2002—5.6R1 Release. Richard Hendricks.

30 September 2002—5.5R1 Release. Richard Hendricks.

27 August 2002—Initial document written. Richard Hendricks.

Chapter 9

Multitopology Routing

This chapter covers the following topics:

- Multitopology Routing Overview on page 271
- Multitopology Routing System Requirements on page 274
- Multitopology Routing Terms and Acronyms on page 274
- Configuring Multitopology Routing on page 274
- Configuring Topologies on page 274
- Configuring Filter-Based Forwarding on page 275
- Configuring BGP for Multitopology Routing on page 275
- Option: Configuring OSPF for Multitopology Routing on page 276
- Option: Configuring Static Routes for Multitopology Routing on page 276
- Option: Configuring Route Resolution Policy on page 277
- Example: Multitopology Routing Configuration on page 277
- For More Information on page 282
- Revision History on page 282

Multitopology Routing Overview

Multitopology Routing (MTR) enables you to configure class-based forwarding for different types of traffic, such as voice, video, and data. Each type of traffic is defined by a topology that is used to create a new routing table for that topology. MTR provides the ability to generate forwarding tables based on the resolved entries in the routing tables for the custom topologies you create. In this way, packets of different classes can be routed independently from one another.

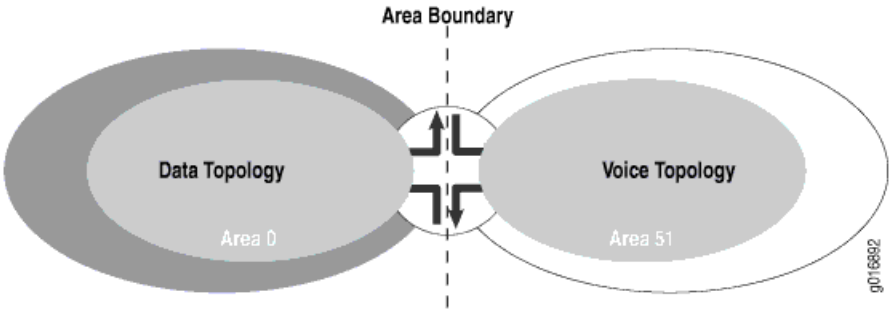
To run MTR, you must configure IP routing. MTR supports OSPFv2, static routes, and BGP. You must configure an interior gateway protocol (IGP), such as OSPFv2 or static routing, Configure BGP to add routes learned through BGP to the appropriate custom topologies. MTR also supports filter-based forwarding, which enables you to match traffic on the ingress interface with a specific type of forwarding class and then forward that traffic to the specified topology.

OSPF in MTR uses a single instance of OSPF to carry connectivity and IP reachability information for different topologies. That information is used to calculate shortest-path first (SPF) trees and routing tables. OSPF for MTR supports protocol extensions that include metrics that correspond to different topologies for link and prefix reachabilty

information. The type-of-service (TOS) metric field is used to advertise the topology-specific metric for links and prefixes belonging to that topology. The TOS field is redefined as MT-ID in the payload of router, summary, and Type 5 and Type 7 AS-external link-state advertisements (LSAs).

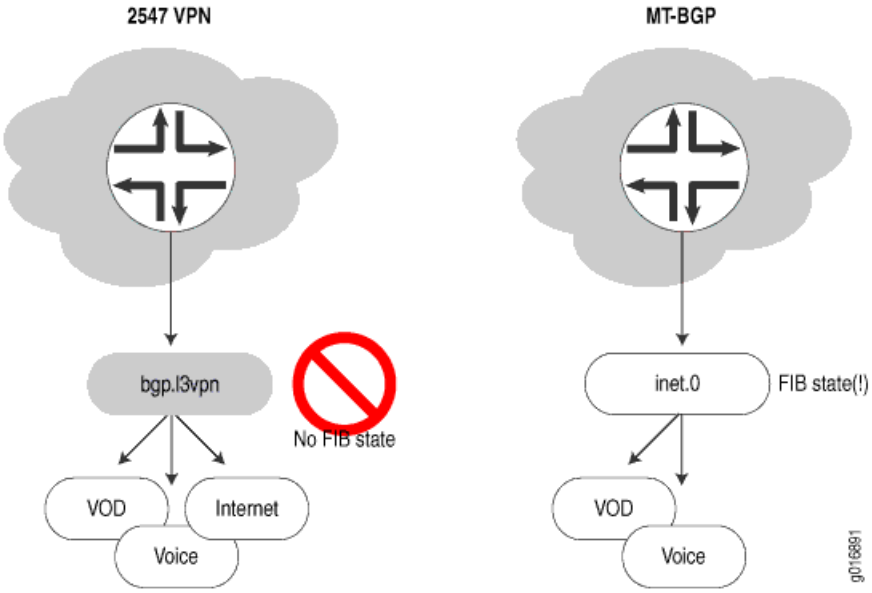
Under MTR, each OSPF interface continues to belong to a single area. Therefore, by default, all topologies share the same area boundaries. As a result, attributes of an area, such as stubbiness, are independent of the topology. By default, all topologies configured for OSPF are enabled on all interfaces. However, you can disable one or more configured topologies on an interface. You can thus allocate an interface for a specific topology. In Figure 19 on page 272, Area 51 includes an interface that is uniquely allocated to voice traffic, and Area 0 includes an interface that is uniquely allocated to data traffic. Each topology thus corresponds to a different OSPF area that shares a boundary.

Figure 19: MT-OSPF Area Boundary



BGP in MTR provides the ability to resolve BGP routes against configured topologies. An inbound policy is used to select routes for inclusion in the appropriate routing tables for the topologies. The default behavior for virtual private networks (VPNs) that use Multiprotocol Label Switching (MPLS) for forwarding packets over the backbone and BGP for distributing routes over the backbone is to place BGP route updates in the bgp.l3vpn routing table. Figure 20 on page 273 shows a BGP peer operating in an environment that conforms with RFC 2547, *BGP/MPLS VPNs*. The figure shows how a BGP peer configured for MTR performs secondary route resolution.

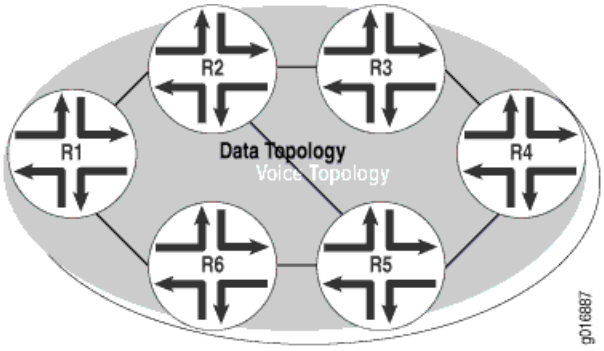
Figure 20: BGP Route Resolution



The BGP peer in a standard VPN topology places prefixes for routes it learns in the bgp.l3vpn routing table, which does not result in automatic updates to the forwarding table. Under BGP in MTR, when BGP receives a route from a peer it attempts to resolve that route against a route in the inet.0 routing table. If the route is resolved, it is placed in that table, which generates a forwarding state. If you have configured a community target identifier that matches the import policy for the topology, routing and forwarding states are added to the tables for the topology.

Because MTR provides support for BGP to perform secondary route resolution, as Figure 21 on page 273 below shows, MTR is able to create two distinct network paths for each type of traffic. Each router advertises BGP routes that need to be resolved against the interior gateway protocol (IGP) routes for each topology. Based on the IGP metrics configured for each topology, for all routes that originate from Router 4 (R4), the upper path between R1 and R4, which traverses R2 and R3, is selected for voice traffic, whereas the lower path between R1 and R4, which traverses R5 and R6, is selected for data traffic.

Figure 21: Route Resolution for MTR



Multitopology Routing System Requirements

To implement MTR, your system must meet these minimum requirements:

- JUNOS Release 9.0 or later
- Two Juniper Networks M Series, MX Series, or T Series routers

Multitopology Routing Terms and Acronyms

T

topology A subset of links in a network for which a separate set of routes is calculated. Those routes are installed in a routing table created specifically for a configured topology used to make routing and forwarding decisions.

Configuring Multitopology Routing

To implement MTR, you must configure the following:

- Configuring Topologies on page 274
- Configuring Filter-Based Forwarding on page 275
- Configuring BGP for Multitopology Routing on page 275
- Option: Configuring OSPF for Multitopology Routing on page 276
- Option: Configuring Static Routes for Multitopology Routing on page 276
- Option: Configuring Route Resolution Policy on page 277

Configuring Topologies

You must configure one or more topologies. For each topology, you specify a string value that defines the type of traffic as well as an interface family. You can also enable a topology for IPv4 multicast traffic by including the ipv4-multicast statement. To configure a topology, include the topologies statement at the [edit routing-options hierarchy level:

```
[edit routing-options]
topologies {
   family inet { # inet6 is also supported, but you must use static routes as the IGP.
      topology voice; # This action creates a routing table called :voice.inet.0
         # for all routes destined for the voice topology. A default topology is also
         # automatically created. Default topology routes are added to the inet.0
         # routing table.
   }
}
```

Configuring Filter-Based Forwarding

Configure a firewall filter that forces a lookup against the different routing tables. Any routes that match the forwarding class specified and then match a specified topology are installed in the routing table for that topology. To configure a firewall filter for MTR that performs filter-based forwarding, include the following configuration at the [edit firewall] hierarchy level:

```
[edit firewall]
family inet { # inet6 is also supported.
   filter topology-selection {
      term ef {
         from {
            forwarding-class expedited-forwarding; # The following class types are also
               # supported: assured-forwarding, best-effort, and network-control.
         }
         then {
            topology voice; # Specify the name of a configured topology.
            accept;
         }
      }
   }
}
```

You must apply the filter to an ingress interface. Include the following statements at the [edit interfaces] hierarchy level to apply the filter to an interface:

```
[edit interfaces]
fe-2/2/1 {
   unit 0 {
      family inet {
         filter {
            input topology-selection; # Specify the name of the filter configured under
               # the [edit firewall] hierarchy
         }
      }
   }
}
```

Configuring BGP for Multitopology Routing

Configure BGP to add routes learned through BGP into a configured topology and the default topology, which is created automatically. Routes for the default topology are installed in the inet.0 routing table. To configure BGP for Multitopology Routing, include the following statements:

```
[edit protocols bgp]
group internal {
   type internal;
   family inet {
      unicast {
         topology voice;
```

```
            community target :1:1; # Any route that has :1:1 as its target destination is
              # installed in the routing table for the voice topology. All received routes
              # are also automatically installed in the default topology.
        }
      }
    }
}
```

Option: Configuring OSPF for Multitopology Routing

To implement MTR, you must configure a interior gateway protocol (IGP) to route local network traffic. MTR supports both OSPFv2 and static routes. Only static routes support IPv6 addresses since MTR does not support OSPFv3. Configure OSPF to add routes from the default topology to the routing table for the specified topology. To enable OSPF for MTR, include the **topology** statement at the **[edit protocols ospf]** hierarchy level:

```
[edit ospf protocols]
topology voice { # Specify a topology name configured under the [edit routing-options]
  # hierarchy level.
  topology-id 127; # Specify a topology identifier from 32 through 127.
}
```

Optionally, you can configure a specific metric for a topology for any interface on which OSPF has been enabled. Any topology-specific metric that you configure applies to routes advertised from that interface that belong only to that topology. To configure a topology-specific metric for an OSPF interface, include the following statements at the **[edit protocols ospf]** hierarchy level:

```
[edit protocols ospf]
area 0.0.0.0 {
  interface fe-2/2/1 {
    metric 10; # You can specify a metric for the interface that overrides the default
      # value of 1.
    }
    topology voice {
      metric 15; # Specify a topology-specific metric from 1 through 65,535.
    }
  }
}
```

Option: Configuring Static Routes for Multitopology Routing

To configure a static route for MTR, you must specify the name of the routing table for the topology. Static routes for MTR support IPv4 and IPv6 addresses. To configure a static route for MTR, include the **rib** statement at the **[edit routing-options]** hierarchy level:

```
[edit routing-options]
rib :voice.inet6.0 { # Specify the routing-table name for the voice topology configured
  # at the [edit routing-options topologies] hierarchy level.
  route 200::a4:0/126 next-hop 200::c0a8:1df;
}
```

Option: Configuring Route Resolution Policy

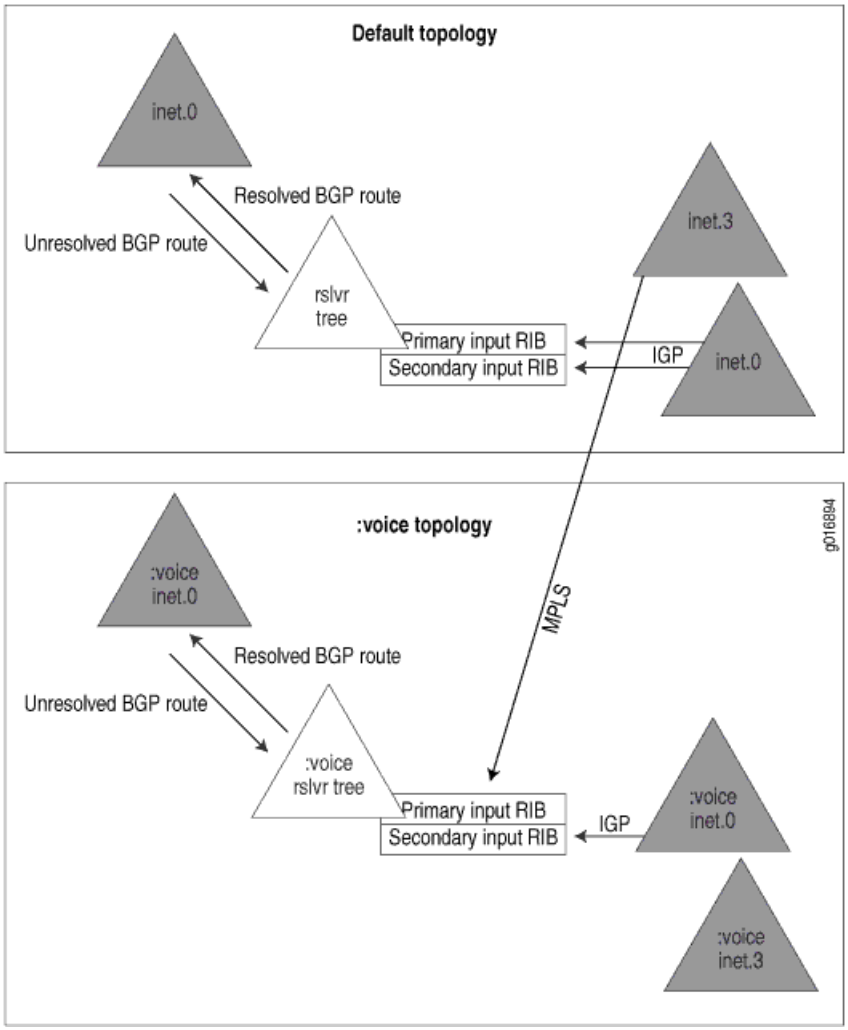
You can optionally configure a route resolution policy so that a routing table accepts routes from specific routing tables. For Multitopology Routing, you might want to configure a policy for the voice topology for IPv4 unicast traffic, for example, :voice.inet.0 to resolve routes through the inet.3 routing table. You might also want to override the default policy and not have the inet.0 routing table use the inet.3 routing table for route resolution. To configure a route resolution policy, include the resolution statement at the [edit routing-options] hierarchy level:

```
[edit routing-options]
resolution {
   rib inet.0 { # Specify the name of the routing table you want to modify.
      resolution-ribs inet.0; # Specify use of the inet.0 table to resolve routes rather
         # than the default policy, which uses the inet.3 table.
   }
   rib :voice.inet.0 { # Specify that the routing table for the voice topology for IPv4
      # unicast traffic be modified.
      resolution-ribs [ inet.3 :voice.inet.0 ]; # Specify that routes be resolved
         # using the inet.3 routing table for the voice topology.
   }
}
```

Example: Multitopology Routing Configuration

In this example, an MPLS network is running RSVP label-switched paths (LSPs) in the core. The network carries both best effort (BE) and expedited forwarding (EF) traffic, and the same destination prefixes are used for both types of traffic. A voice topology is created to enable the network to send EF traffic over the LSPs, but permit BE traffic to traverse the IP path. The voice, or EF traffic, is placed in the voice topology. The data, or BE traffic, is placed in the default topology. The voice and default topologies each create separate routing tables for storing routes, and each routing table creates a separate forwarding table. For each destination prefix, a different route is added to each topology-specific routing table. These routes are BGP routes. You configure filter-based forwarding so that the destination lookup is done in the two topology-specific routing tables based on the DSCP marking of the incoming packet. As Figure 22 on page 278 shows, MTR thus enables you to build two routes to the destination prefixes, one using LSP next hops for voice traffic and one using IP next hops for data traffic. The protocol next hop is resolved differently for each type of traffic. For voice EF traffic, routes from the inet.3 routing table are taken into account, and for BE data traffic, only routes from the inet.0 routing table are taken into account.

Figure 22: Route Resolution in Multitopology Routing



Configure the interfaces:

```
[edit]
interfaces {
   so-0/0/1 {
      unit 0 {
         family inet {
            address 1.13.1.1/24
         }
         family iso;
         family mpls;
      }
   }
   fe-2/2/0 {
      unit 0 {
         family inet {
            address 1.12.1.1/24
         }
```

```
            family iso;
            family mpls;
        }
    }
    fe/2/2/1 {
    unit 0 {
        family inet {
            filter {
                input topo_selection; # Apply a firewall filter on the ingress interface.
                    # This filter performs filter-based forwarding for the voice topology.
            }
            family iso;
            family mpls;
        }
    }
}
```

Configure the voice topology. Configure a route resolution policy so that IPv4 routes for data traffic are resolved through the inet.0 routing table, which functions as the routing table for the default topology. Configure a route resolution policy so that routes for voice traffic are resolved through the routing table for the voice topology (:voice.inet.0) and the MPLS routing table (inet.3).

```
[edit]
routing-options {
    autonomous-system 65300;
    resolution {
        rib inet.0 {
            resolution-ribs inet.0; # Specify use of the inet.0 routing table to resolve
                # IPv4 data traffic. This action prevents this traffic from being resolved using
                # the MPLS routing table (inet.3).
        }
        rib :voice.inet.0 {
            resolution-ribs [ inet.3 :voice.inet.0 ]; # Specify use of the MPLS routing
                # table (inet.3) and the routing table for the voice topology (:voice.inet.0)
                # to resolve IPv4 voice traffic. This action prevents voice traffic from being
                #resolved using the inet.0 routing table.
        }
    }
    topologies {
        family inet {
            topology voice;
        }
    }
}
```

Configure MPLS using RSVP label-switched paths. Configure BGP so that routes learned through BGP are installed in the appropriate routing table. In this example, the topology statement is used to install BGP routes for voice traffic into the routing table for the voice topology (:voice.inet.0). This action overrides the default behavior to resolve BGP routes only through the inet.0 or inet.3 routing tables. Configure an interior gateway protocol (IGP). In this example, you configure OSPF.

```
[edit]
protocols {
```

```
    rsvp {
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
    mpls {
        label-switched-path to_r3 {
            to 10.255.14.222;
            primary test;
        }
        path test {
            1.12.1.2 strict;
        }
        interface all;
    }
    bgp {
        group int {
            type internal;
            local-address 10.255.14.223;
            family inet {
                unicast {
                    topology voice {
                        community 70:1;
                    }
                }
            }
            neighbor 10.255.14.220;
            neighbor 10.255.14.218;
            neighbor 10.255.14.222;
        }
    }
    ospf {
        topology voice topology-id 32;
        traffic-engineering;
        area 0.0.0.0 {
            interface lo0.0 {
                passive;
            }
            interface all;
            interface fxp0.0 {
                disable;
            }
            interface fe-2/2/1.0 {
                metric 1;
            }
            interface fe-2/2/0.0 {
                metric 1;
            }
            interface so-0/0/1.0 {
                metric 1;
            }
        }
    }
}
```

Configure a class-of-service classifier on the ingress interface. In this example, the classifier type is inet-precedence, which evaluates incoming IPv4 packets and requires only the upper three bits of the DSCP field.

```
[edit]
class-of-service {
   interfaces {
      fe-2/2/1 {
         unit 0 {
            classifiers {
               inet-precedence default;
            }
         }
      }
   }
}
```

Configure filter-based forwarding. This filter is applied to the ingress interface. Traffic marked for expedited forwarding is forwarded to the routing table for the voice topology. All other traffic is forwarded to the routing table for the default topology.

```
[edit]
firewall {
   family inet {
      filter topo_selection {
         term ef {
            from {
               forwarding-class expedited-forwarding;
            }
            then {
               topology voice; # Forward expedited-forwarding traffic to the routing
                  # table for the voice topology (:voice.inet.0).
               accept;
            }
         }
         term default {
            then accept; # Forward all other traffic to the routing table for the default
               # topology (inet.0).
         }
      }
   }
}
```

Verifying Your Work

To verify proper operation of Multitopology Routing, use the following commands:

- show route summary
- show route table *routing-table-name*
- show route rib-groups

For More Information

For additional information about Multitopology Routing, see the following resources:

- *JUNOS Routing Protocols Configuration Guide*
- RFC 4915, *Multi-Topology (MT) Routing in OSPF*

Revision History

5 October 2009—10.0R1 Release. Roy Spencer.

13 June 2009—9.6R1 Release. Roy Spencer.

13 April 2009—9.5R1 Release. Roy Spencer.

15 January 2009—9.4R1 Release. Roy Spencer.

10 October 2008—9.3R1 Release. Roy Spencer.

10 July 2008—9.2R1 Release. Roy Spencer.

10 April 2008—9.1R1 Release. Ines Salazar.

1 February 2008—Initial document written. Ines Salazar.

Part 3
VPNs

- Layer 2 Circuits on page 285
- Multicast over Layer 3 VPNs on page 339
- Translational Cross-Connect and Layer 2.5 VPNs on page 473
- Virtual Private LAN Service on page 501

Chapter 10

Layer 2 Circuits

This chapter covers the following topics:

- Layer 2 Circuits Overview on page 286
- Layer 2 Circuits System Requirements on page 289
- Layer 2 Circuits Terms and Acronyms on page 289
- Configuring Layer 2 Circuits on page 290
- Configuring CCC Encapsulation on CE-Facing Ethernet Interfaces on page 290
- Configuring CCC Encapsulation on CE-Facing SONET/SDH Interfaces on page 291
- Configuring a CCC Encapsulation and a Layer 2 Circuit Mode on CE-Facing ATM2 IQ Interfaces on page 292
- Configuring the MPLS Family on Core Interfaces on page 293
- Configuring the Layer 2 Circuit Neighbor Address and Virtual Circuit Identifier on page 294
- Configuring LDP and an IGP to Transport Layer 2 Circuits on page 295
- Option: Applying Traffic Engineering to a Layer 2 Circuit on page 296
- Option: Mapping Layer 2 Protocol Control Information into a Layer 2 Circuit on page 297
- Option: Configuring APS for Layer 2 Circuits on page 297
- Option: Configuring Layer 2 Circuit Trunk Mode on ATM2 IQ Interfaces on page 298
- Option: Reserving LSP Bandwidth for a Layer 2 Circuit on page 300
- Option: Selecting an MTU for a Layer 2 Circuit on page 301
- Option: Configuring Local Interface Switching for a Layer 2 Circuit on page 302
- Option: Configuring Layer 2 Circuits Simultaneously over RSVP and LDP LSPs on page 302
- Layer 2 Circuit Configuration Examples on page 303
- Example: Ethernet-Based Layer 2 Circuit Configuration on page 303
- Example: SONET/SDH-Based Layer 2 Circuit Configuration on page 310
- Example: ATM2 IQ-Based Layer 2 Circuit Configuration on page 315
- Example: Layer 2 Circuit Traffic Engineering over Multiple LSPs Configuration on page 324

- Example: APS for a Layer 2 Circuit Configuration on page 334
- For More Information on page 336
- Revision History on page 337

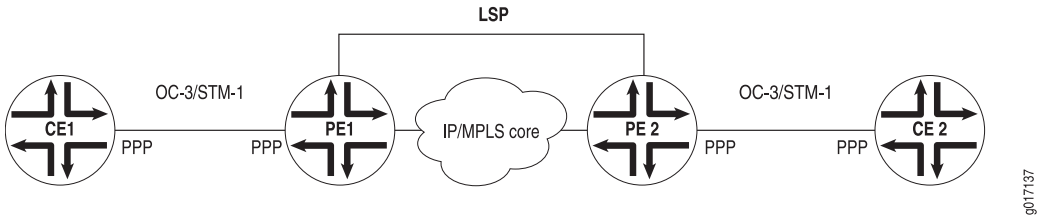
Layer 2 Circuits Overview

After the introduction and early adoption of Layer 3 virtual private networks (VPNs) based on RFC 4364 (also known as RFC 2547bis), many customers asked their service providers to offer VPNs that would preserve data at Layer 2. One of the Layer 2 VPN options that has emerged is known as a Layer 2 circuit. It is based on a series of Internet Engineering Task Force (IETF) drafts and RFCs authored by Luca Martini. These so-called "Martini-drafts" include Internet draft draft-martini-l2circuit-encap-mpls-11.txt, *Encapsulation Methods for Transport of Layer 2 Frames Over MPLS Networks* (expires August 2006) and Internet draft draft-martini-l2circuit-trans-mpls-19.txt, *Transport of Layer 2 Frames Over MPLS* (expires September 2006), and establish the basis for Juniper Networks implementation of Layer 2 circuits. This guide shows how to implement Layer 2 circuits in a variety of ways.

Layer 2 circuits allow for the creation of point-to-point Layer 2 connections over an IP and Multiprotocol Label Switching (MPLS)-based network. Physical circuits with the same Layer 2 encapsulations can be connected together across such a network. Layer 2 circuits can allow for the replacement of end-to-end Asynchronous Transfer Mode (ATM) networks, Frame Relay networks, and some portions of Time-Division Multiplexing (TDM) networks, with an IP and MPLS-based network.

In Figure 23 on page 286, an OC3/STM1 interface encapsulated with the Point-to-Point Protocol (PPP) on Router PE1 is connected over a Layer 2 circuit to reach an OC3/STM1 interface encapsulated with PPP on Router PE2. To enable the Layer 2 circuits to operate, the provider edge (PE) routers in Figure 23 on page 286 are part of an MPLS network. Routers PE1 and PE2 must also be Label Distribution Protocol (LDP) peers. Additionally, any interface on the PE routers that connects to a customer edge (CE) router must support circuit cross-connect (CCC) interface encapsulations.

Figure 23: Layer 2 Circuit Connection



Layer 2 circuits are very similar to Layer 2 VPNs. However, there are some significant differences:

- You configure Layer 2 VPNs in a routing instance. As a result, Layer 2 VPNs have unique site and VPN identifiers. However, Layer 2 circuits do not require a routing instance configuration and instead use an alternate method of identifying circuits. Layer 2 circuit peer relationships are established with three components: a logical

interface on the local PE router, the IP address of the remote PE router neighbor, and a virtual circuit identifier.

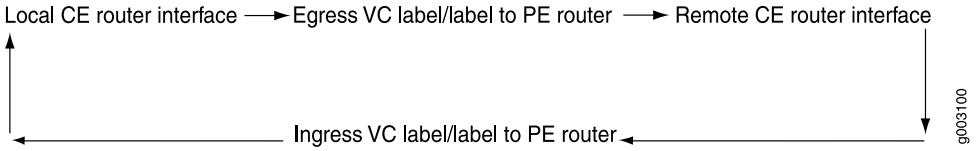
- Layer 2 VPNs, like Layer 3 VPNs, require Border Gateway Protocol (BGP) for transport of traffic between PE routers. In contrast, Layer 2 circuits do not require BGP. Instead, Layer 2 circuits rely on LDP and MPLS for their operation. As a result, Layer 2 circuits require less configuration than Layer 2 VPNs.

Layer 2 circuits are configured between two peers. The peers must use the same interior gateway protocol (IGP), such as Open Shortest Path First (OSPF) or Intermediate System-to-Intermediate System (IS-IS). Also, the peers must have a symmetrical Layer 2 configuration and belong to the same routing domain or autonomous system.

The basic building block for a Layer 2 circuit is a virtual circuit (VC). A VC is a point-to-point Layer 2 connection that is transported over MPLS or any other tunneling technology in a service provider network. A VC is similar to a CCC connection except that multiple VCs are transported over a single MPLS label-switched path (LSP) tunnel between two provider edge (PE) routers. In contrast, CCC only supports a single Layer 2 connection over a single LSP.

In Figure 24 on page 287, the basic inner workings of Layer 2 circuits are explained. Two customer edge (CE) router logical interfaces, one local and one remote, are running the same Layer 2 protocol. Packets are sent from the local CE router to the remote CE router over an egress label advertised by the remote PE router. The label is transported over an LDP LSP (or LDP tunneled through RSVP) to the remote PE router that is connected to the remote CE router. Return traffic from the remote CE router is sent over an ingress label advertised by the local PE router. Once again, the label rides over an LDP LSP (or LDP tunneled through RSVP) to the local PE router from the remote PE router.

Figure 24: Layer 2 Circuit Concept



The Layer 2 circuit framework requires LDP to be used as the signaling protocol for advertising ingress labels. In most cases, it is not necessary to transport the Layer 2 encapsulation across the network; rather, the Layer 2 header can be stripped at one PE router, and reproduced at the egress PE router. Such Layer 2 information is carried in a special Layer 2 circuit header called a *control word*.

In the Layer 2 circuit IETF drafts, the control word is optional for most Layer 2 protocols, except Frame Relay and ATM AAL5 where it is required. However, in JUNOS Release 5.6 and later, a control word for all forms of Layer 2 circuits is sent by default. If you are establishing a Layer 2 circuit between a router running JUNOS Release 5.5 or earlier and a router running JUNOS Release 5.6 or later, use of the control word is negotiated automatically.

The Layer 2 protocols that are supported for Layer 2 circuits are:

- ATM cell-relay mode and ATM Adaptation Layer 5 (AAL5) mode on ATM2 intelligent queuing (IQ) interfaces
- Cisco High-Level Data Link Control (HDLC), Frame Relay, and PPP on SONET/SDH-based interfaces
- Ethernet, VLAN, and Extended VLAN on Ethernet-based interfaces

For an Ethernet 802.1q VLAN or simple Ethernet, the entire Ethernet frame without the preamble or frame check sequence (FCS) is transported. For ATM cell-relay mode, ATM cells are transported without a SAR process. For Cisco HDLC, the frame is transported in its entirety except for HDLC flags and the FCS. For PPP, the frame is transported in its entirety except for any media-specific framing information.

For most protocols, a null control word consisting of all zeroes is sent between Layer 2 circuit neighbors. However, individual bits are available in a control word that can carry Layer 2 protocol control information. The control information is mapped into the control word, which allows the header of a Layer 2 protocol to be stripped from the frame. The remaining data and control word can be sent over the Layer 2 circuit, and the frame can be reassembled with the proper control information at the egress point of the circuit.

The Layer 2 protocols that map Layer 2 control information into special bit fields in the control word are as follows:

- Frame Relay—This control word supports the transport of discard eligible (DE), forward explicit congestion notification (FECN), and backward explicit congestion notification (BECN) information. (For configuration information, see “Option: Mapping Layer 2 Protocol Control Information into a Layer 2 Circuit” on page 297.)
- ATM AAL5 mode—This control word supports the transport of sequence number processing, ATM cell loss priority (CLP), and explicit forward congestion indication (EFCI) information. When you configure an AAL5 mode Layer 2 circuit, the control information is carried by default and no additional configuration is needed.
- ATM cell-relay mode—This control word supports sequence number processing only. When you configure a cell-relay mode Layer 2 circuit, the sequence number information is carried by default and no additional configuration is needed.

The JUNOS Software implementation of sequence number processing for ATM cell-relay mode and AAL5 mode is not the same as that described in Sec. 3.1.2 of the IETF draft *Encapsulation Methods for Transport of Layer 2 Frames Over MPLS Networks*. The differences are as follows:

- A packet with a sequence number of 0 is treated as out of sequence.
- Any packet which does not have the next incremental sequence number is considered out of sequence.
- When out-of-sequence packets arrive, the sequence number in the Layer 2 circuit control word increments by one and becomes the expected sequence number for the neighbor.

Layer 2 Circuits System Requirements

To implement Layer 2 circuits, your system must meet these minimum requirements:

- JUNOS Release 9.1 or later for nonstop routing (NSR) support.
- JUNOS Release 8.2 or later for M120 and MX Series routing platform support.
- JUNOS Release 7.3 or later for configuring Layer 2 circuits simultaneously over RSVP and LDP LSPs
- JUNOS Release 7.1 or later for local interface switching for Layer 2 circuits
- JUNOS Release 7.0 or later for specifying a unique maximum transmission unit (MTU) for each Layer 2 circuit
- JUNOS Release 6.4 or later for ATM2 IQ interface-based CoS and bandwidth reservation for trunks
- JUNOS Release 6.2 or later for Layer 2 circuit trunk mode on T Series and M320 routers and bandwidth reservation for Layer 2 circuits
- JUNOS Release 6.1 or later for Automatic Protection Switching (APS) for Layer 2 circuits
- JUNOS Release 6.0 or later for Layer 2 circuit traffic engineering, and Frame Relay or ATM control word mapping
- JUNOS Release 5.7 or later for ATM cell-relay mode or AAL5 Layer 2 circuits
- JUNOS Release 5.6 or later for Frame Relay, HDLC, and PPP-based Layer 2 circuits
- JUNOS Release 5.2 or later for Ethernet-based Layer 2 circuits
- Five Juniper Networks M Series, MX Series, or T Series routers

Layer 2 Circuits Terms and Acronyms

C

circuit cross-connect (CCC) A Juniper Networks method of exchanging frames between one router interface running a Layer 2 protocol and another router interface using the same Layer 2 protocol. For more information about CCC, see either the *JUNOS Network Interfaces Configuration Guide* or the *JUNOS MPLS Applications Configuration Guide*.

control word A 32-bit field used in Layer 2 circuits to transport sequence information, Layer 2 media control information, and padding.

L

Layer 2 circuit A method of transporting Layer 2 frames between provider edge (PE) routers across an MPLS backbone using LDP signaling.

Configuring Layer 2 Circuits

When you configure Layer 2 circuits, you can use Ethernet, SONET/SDH, and ATM2 IQ interfaces on a PE router. The specific steps you must take to configure these interface types for Layer 2 circuits are described as follows:

- Configuring CCC Encapsulation on CE-Facing Ethernet Interfaces on page 290
- Configuring CCC Encapsulation on CE-Facing SONET/SDH Interfaces on page 291
- Configuring a CCC Encapsulation and a Layer 2 Circuit Mode on CE-Facing ATM2 IQ Interfaces on page 292
- Configuring the MPLS Family on Core Interfaces on page 293
- Configuring the Layer 2 Circuit Neighbor Address and Virtual Circuit Identifier on page 294
- Configuring LDP and an IGP to Transport Layer 2 Circuits on page 295
- Option: Applying Traffic Engineering to a Layer 2 Circuit on page 296
- Option: Mapping Layer 2 Protocol Control Information into a Layer 2 Circuit on page 297
- Option: Configuring APS for Layer 2 Circuits on page 297
- Option: Configuring Layer 2 Circuit Trunk Mode on ATM2 IQ Interfaces on page 298
- Option: Reserving LSP Bandwidth for a Layer 2 Circuit on page 300
- Option: Selecting an MTU for a Layer 2 Circuit on page 301
- Option: Configuring Local Interface Switching for a Layer 2 Circuit on page 302
- Option: Configuring Layer 2 Circuits Simultaneously over RSVP and LDP LSPs on page 302

Configuring CCC Encapsulation on CE-Facing Ethernet Interfaces

On Ethernet-based CE-facing PE router interfaces, you must configure one of the three Ethernet CCC encapsulation types—Ethernet CCC, VLAN CCC, or Extended VLAN CCC. Use the following guidelines to configure an Ethernet-based interface CCC encapsulation:

- ethernet-ccc—Use Ethernet CCC encapsulation on Ethernet interfaces that must accept packets carrying standard Tag Protocol ID (TPID) values.
- extended-vlan-ccc—Use extended VLAN CCC encapsulation on Ethernet interfaces that have VLAN 802.1Q tagging and must accept packets carrying TPIDs 0x8100, 0x9100, and 0x9901.
- vlan-ccc—Use VLAN CCC encapsulation on Ethernet interfaces with VLAN tagging enabled. VLAN CCC encapsulation supports TPID 0x8100 only. You must configure this encapsulation type on both the physical interface and the logical interface.

- You can configure multiple logical interfaces on Ethernet interfaces configured for VLAN mode. The valid VLAN ID range for the logical interfaces depends upon your PIC type:
  - 1 to 1023—Aggregated Ethernet, 4-port, 8-port, and 12-port Fast Ethernet, and management and internal Ethernet interfaces
  - 1 to 4094—48-port Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet interfaces
  - VLAN ID 0 is reserved for tagging the priority of frames on all Ethernet PICs.
- For encapsulation type vlan-ccc, VLAN IDs from 1 through 511 are reserved for normal VLANs, and VLAN IDs from 512 through 1023 are reserved for CCC VLANs. For encapsulation type extended-vlan-ccc, all valid VLAN IDs per PIC type are valid for CCC VLANs.

For more information about VLAN IDs, see "Binding a VLAN ID to a Logical Interface" in the *JUNOS Network Interfaces Configuration Guide*.

To configure CCC interface encapsulation, include the encapsulation statement at the [edit interfaces *ethernet-interface-fpc/pic/port*] hierarchy level and select ethernet-ccc, vlan-ccc, or extended-vlan-ccc as the encapsulation type. If you select the VLAN CCC encapsulation, also include the vlan-ccc statement at the [edit interfaces *ethernet-interface-fpc/pic/port* unit *unit-number* encapsulation] logical interface hierarchy level. When using either VLAN CCC or extended VLAN CCC encapsulations, include the vlan-tagging statement at the [edit interfaces *ethernet-interface-fpc/pic/port*] hierarchy level.

```
[edit]
interfaces {
    fe-0/1/0 {
        vlan-tagging;
        encapsulation vlan-ccc;
        unit 0 {
            encapsulation vlan-ccc;
            vlan-id 600;
        }
    }
}
```

Configuring CCC Encapsulation on CE-Facing SONET/SDH Interfaces

On SONET/SDH interfaces in a PE router, you can use Frame Relay CCC, Cisco HDLC CCC, or PPP CCC encapsulation for Layer 2 circuits:

- To configure Frame Relay CCC interface encapsulation, include the encapsulation statement at the [edit interfaces so-*fpc/pic/port*] hierarchy level and select frame-relay-ccc as the encapsulation type. To enable Frame Relay CCC at the logical interface level, include the encapsulation frame-relay-ccc statement at the [edit interfaces so-*fpc/pic/port* unit *unit-number*] hierarchy level.
- To configure Cisco HDLC CCC interface encapsulation, include the encapsulation statement at the [edit interfaces so-*fpc/pic/port*] hierarchy level and select cisco-hdlc-ccc as the encapsulation type.

- To configure PPP CCC interface encapsulation, include the encapsulation statement at the [edit interfaces so-*fpc/pic/port*] hierarchy level and select ppp-ccc as the encapsulation type.

```
[edit]
interfaces {
   so-0/0/0 {
      encapsulation (frame-relay-ccc | cisco-hdlc-ccc | ppp-ccc);
      unit 0 {
         encapsulation frame-relay-ccc;
      }
   }
}
```

Configuring a CCC Encapsulation and a Layer 2 Circuit Mode on CE-Facing ATM2 IQ Interfaces

On ATM2 IQ interfaces in a PE router, you need to configure two encapsulations to enable Layer 2 circuits: one at the [edit interfaces at-*fpc/pic/port*] hierarchy level and the other at the [edit chassis fpc *fpc-slot* pic *pic-slot*] hierarchy level. There are two types of ATM2 IQ Layer 2 circuits: cell-relay mode and ATM Adaptation Layer 5 (AAL5) mode. For both modes, you must specify the Physical Interface Card (PIC) type with the pic-type atm2 statement at the [edit interfaces at-*fpc/pic/port* atm-options] hierarchy level. You can configure only one mode per PIC at a time. If you need to enable both ATM2 IQ Layer 2 circuit modes in the same router, you must configure the different modes on different PICs.

To configure a cell-relay mode Layer 2 circuit, include the atm-l2circuit-mode cell statement at the [edit chassis fpc *fpc-slot* pic *pic-slot*] hierarchy level and the encapsulation atm-ccc-cell-relay statement at both the [edit interfaces at-*fpc/pic/port*] physical hierarchy level and the [edit interfaces at-*fpc/pic/port* unit *unit-number*] logical interface hierarchy level.

```
[edit]
chassis {
   fpc 0 {
      pic 1 {
         atm-l2circuit-mode {
            cell;
         }
      }
   }
}
interfaces {
   at-0/1/0 {
      encapsulation atm-ccc-cell-relay;
      atm-options {
         cell-bundle-size 4;
         pic-type atm2;
         vpi 0;
      }
      unit 0 {
         encapsulation atm-ccc-cell-relay;
```

```
            vci 32;
            cell-bundle-size 10;
        }
    }
}
```

For ATM2 IQ Layer 2 circuit cell-relay mode only, you can adjust the cell bundle size at the physical interface level and the logical interface level. To configure, include the cell-bundle-size statement at either the [edit interfaces at-*fpc/pic/port* atm-options] physical interface hierarchy level or the [edit interfaces at-*fpc/pic/port* unit *unit-number*] logical interface hierarchy level. If the statement is included at both levels, the logical interface setting takes precedence. The default value for cell bundle size is 1 and the maximum value is 190. If you configure the cell bundle size statement, you should configure the same value on all ATM2 IQ neighbors.

To configure an AAL5 mode Layer 2 circuit, include the atm-l2circuit-mode aal5 statement at the [edit chassis fpc*fpc-slot* pic *pic-slot*] hierarchy level and the encapsulation atm-ccc-vc-mux statement at the [edit interfaces at-*fpc/pic/port*] hierarchy level:

```
[edit]
chassis {
    fpc 1 {
        pic 2 {
            atm-l2circuit-mode {
                aal5;
            }
        }
    }
}
interfaces {
    at-1/2/0 {
        atm-options {
            pic-type atm2;
            vpi 0;
        }
        unit 0 {
            encapsulation atm-ccc-vc-mux;
            vci 32;
        }
    }
}
```

For more information on how to configure interfaces with CCC encapsulation types, see the *JUNOS MPLS Applications Configuration Guide* or the *JUNOS Network Interfaces Configuration Guide*.

Configuring the MPLS Family on Core Interfaces

Because LDP is used as the signaling protocol to transport MPLS labels across the core of the network, you must include the family mpls statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level. Include the statement on all router interfaces in the path from the local PE router to the remote PE router across the core network that transports the Layer 2 circuit traffic.

```
[edit]
interfaces {
   interface-name  {
      unit logical-unit-number {
         family inet {
            address ip-address/prefix;
         }
         family mpls;
      }
   }
}
```

Configuring the Layer 2 Circuit Neighbor Address and Virtual Circuit Identifier

After you enable the PE router interfaces with the proper encapsulations, you then configure Layer 2 circuits (also referred to as VCs) between two or more PE router neighbors. To configure a Layer 2 circuit, include the l2circuit statement at the [edit protocols] hierarchy level.

Each Layer 2 circuit is represented by a logical interface on the local PE router, the IP address of the remote PE router neighbor, and a virtual circuit identifier. The logical interface connects the local PE router to the local CE router. The loopback address and router ID of the PE neighbor is commonly the neighbor's IP address. This address is also the destination end-point of the LSP tunnel, which transports the Layer 2 circuit to the neighbor. The virtual circuit ID uniquely identifies the VC to a specific neighbor.

This combination of logical interface, neighbor address, and virtual circuit ID is used to map a particular LDP forwarding equivalence class (FEC) received from a specific neighbor to a local VC. The egress label is added to a table and is used for sending traffic on that VC between the CE routers.

Both ends of a Layer 2 circuit must use the same Layer 2 technology because the Layer 2 encapsulation type is carried in the LDP FEC. The encapsulation type from a received FEC is matched against the local encapsulation type of the VC. If there is a mismatch, the VC is not established.

To add the IP address of the remote PE router neighbor into a Layer 2 circuit, include the neighbor *ip-address* statement at the [edit protocols l2circuit] hierarchy level. To map the remote neighbor to the local interface that connects to the CE router, include the interface statement at the [edit protocols l2circuit neighbor *ip-address*] hierarchy level. To select the identifier for the virtual circuit, include the virtual-circuit-identifier statement at the [edit protocols l2circuit neighbor *ip-address* interface *interface-name*] hierarchy level. To disable default control word processing, include the no-control-word statement at the [edit protocols l2circuit neighbor *ip-address* interface *interface-name*] hierarchy level. Finally, to assign the Layer 2 circuit to a community, include the community *community-name*statement at the [edit protocols l2circuit neighbor *ip-address* interface *interface-name*] hierarchy level.

NOTE: On M Series routers only, if you include the control-word statement at the [edit protocols l2circuit neighbor *address* interface *interface-name*] hierarchy level (the default setting for Layer 2 circuits), the software cannot rewrite MPLS EXP bits.

```
[edit]
protocols {
   l2circuit {
      traceoptions {
         file name [replace] [size size] [files files ] [nostamp];
         flag (error | topology | nlri | connections | route) [detail];
      }
      neighbor ip-address {
         interface interface-name {
            virtual-circuit-id identifier;
            no-control-word;
            community community-name;
         }
      }
   }
}
```

You do not need to specify the encapsulation type at the [edit protocols l2circuit] hierarchy level because it is already specified in the interface configuration.

Configuring LDP and an IGP to Transport Layer 2 Circuits

LDP is used as the signaling protocol to advertise the ingress MPLS label to the remote PE router. For this purpose, a remote LDP neighbor is established using the extended discovery mechanism described in RFC 3036, *LDP Specification*, and a session is established.

No new configuration is necessary in LDP because the LDP protocol recognizes the Layer 2 circuit configuration and initiates extended neighbor discovery for all Layer 2 circuit neighbors on the remote PE routers. This is very similar to the behavior of LDP when it is tunneled over RSVP. However, you must configure LDP on the lo0.0 interface for extended neighbor discovery to function correctly.

LDP relies on an underlying IGP, such as OSPF or IS-IS. Therefore, configure LDP and your IGP on all routers in the path from the local PE router to the remote PE router across the service provider backbone.

```
[edit]
protocols {
   ospf {
      traffic-engineering;
      area 0.0.0.0 {
         interface so-0/1/0.0;
         interface lo0.0;
      }
   }
   ldp {
      interface so-0/1/0.0;
```

```
        interface lo0.0;
    }
}
```

Option: Applying Traffic Engineering to a Layer 2 Circuit

To traffic engineer Layer 2 circuits over multiple LSPs, you must create a community, assign a set of Layer 2 circuits to that community, define a policy to send the community traffic over a desired LSP, and apply the policy to the forwarding table.

To create a community, include the community *community-name* statement at the [edit policy-options] hierarchy level. To assign a Layer 2 circuit to a community, include the community *community-name* statement at the [edit protocols l2circuit neighbor *ip-address* interface *interface-name*] hierarchy level. To create a policy that sends community traffic over a specific LSP, include the community *community-name*statement at the [edit policy-options policy-statement *policy-name* term *term-name* from] hierarchy level and the install-next-hop lsp *lsp-name* statement at the [edit policy-options policy-statement *policy-name* term *term-name* then] hierarchy level. To apply the policy to the forwarding table, include the export *policy-name* statement at the [edit routing-options forwarding-table] hierarchy level.

```
[edit]
routing-options {
    forwarding-table {
        export policy-name;
    }
}
protocols {
    l2circuit {
        neighbor ip-address {
            interface interface-name {
                virtual-circuit-id identifier;
                community community-name;
            }
        }
    }
}
policy-options {
    policy-statement policy-name {
        from community community-name;
        then {
            install-nexthop lsp lsp-name;
            accept;
        }
    }
}
community community-name members value;
```

Option: Mapping Layer 2 Protocol Control Information into a Layer 2 Circuit

The control word is defined in Internet draft draft-martini-l2circuit-encap-mpls-07.txt *Encapsulation Methods for Transport of Layer 2 Frames Over IP and MPLS Networks*. It is a set of fields that carry Layer 2 control information across a Layer 2 circuit. The following control word support is available for PE routers:

- Frame Relay—To carry Frame Relay FECN/BECN information in a Layer 2 circuit control word, include the translate-fecn-and-becn statement at the [edit interfaces *interface-name* encapsulation frame-relay-ccc unit *unit-number* encapsulation frame-relay-ccc family ccc] hierarchy level. To carry Frame Relay DE information in a Layer 2 circuit control word, include the translate-discard-eligible statement at the [edit interfaces *interface-name* encapsulation frame-relay-ccc unit *unit-number* encapsulation frame-relay-ccc family ccc] hierarchy level.

```
[edit]
interfaces {
   interface-name {
      encapsulation frame-relay-ccc;
      unit 0 {
         encapsulation frame-relay-ccc;
         point-to-point;
         dlci 512;
         family ccc {
            translate-fecn-and-becn;
            translate-discard-eligible;
         }
      }
   }
}
```

- ATM AAL5 mode—For ATM2 IQ interfaces, the ATM AAL5 control word contains bit fields to carry sequence numbers, ATM cell loss priority (CLP), and explicit forward congestion indication (EFCI) information. When you configure ATM Layer 2 circuits, the control word carries the sequence number, CLP, and EFCI information by default. No additional configuration is necessary.
- ATM cell-relay mode—For ATM2 IQ interfaces, the ATM cell-relay control word supports sequence number processing only. Once you configure a cell-relay mode Layer 2 circuit, the sequence number information is carried by default. No additional configuration is necessary.

Option: Configuring APS for Layer 2 Circuits

To apply Automatic Protection Switching (APS) to a Layer 2 circuit, you must configure an APS working circuit and a protect circuit on PE router interfaces that support SONET options (such as SONET/SDH, ATM, and ATM2 IQ interfaces) and circuit cross-connect (CCC) encapsulation types. Then, you must configure the working circuit as the primary Layer 2 circuit interface and the protect circuit as the protected Layer 2 circuit interface. Note that APS only protects the PE-CE link and not the entire Layer 2 circuit.

To configure an APS working circuit, include the working-circuit statement at the [edit interfaces *interface-name* sonet-options aps] hierarchy level. To configure an APS protect circuit, include the protect-circuit statement at the [edit interfaces *interface-name* sonet-options aps] hierarchy level. To configure the primary Layer 2 circuit interface, include the interface statement at the [edit protocols l2circuit neighbor *ip-address*] hierarchy level. To configure the protected Layer 2 circuit interface, include the protect-interface statement at the [edit protocols l2circuit neighbor *ip-address* interface *interface-name*] hierarchy level.

```
[edit]
interfaces {
   at-0/0/1 {
      description " APS protect circuit";
      encapsulation CCC-encapsulation-type;
      sonet-options {
         aps {
            protect-circuit name;
         }
      }
   }
   at-1/3/1 {
      description "APS working circuit";
      encapsulation CCC-encapsulation-type;
      sonet-options {
         aps {
            working-circuit name;
         }
      }
   }
}
protocols {
   l2circuit {
      neighbor ip-address {
         interface at-1/3/1.0 {
            protect-interface at-0/0/1.0;
            virtual-circuit-id number;
         }
      }
   }
}
```

Option: Configuring Layer 2 Circuit Trunk Mode on ATM2 IQ Interfaces

When you configure Layer 2 circuits on CE-facing ATM2 IQ interfaces in a PE router that connects to some vendors' ATM switches, you can create a trunk. The trunk bundles several ATM cell streams into one LSP, preserves the cell loss priority (CLP) and class-of-service (CoS) information of the cells within the experimental (EXP) bits of the MPLS header, and provides network-to-network interface (NNI) or user-to-network interface (UNI) information within a proprietary header. A physical interface supports a total of 32 logical trunks in NNI mode and 8 logical trunks when you use the UNI option. To configure a trunk, include the trunk statement at the [edit chassis fpc *fpc-slot* pic *pic-slot* atm-l2circuit-mode] hierarchy level, select NNI or UNI mode with the nni or uni statement, and specify a number of bits in the ATM header

that will carry an identifier with the id-width statement. You can choose a value from 1 through 8 for the identifier width.

```
[edit]
chassis {
   fpc fpc-slot {
      pic pic-clot {
         atm-l2circuit-mode {
            trunk {
               (nni | uni) {
                  id-width number;
               }
            }
         }
      }
   }
}
```

You can also configure several trunk options at the [edit interfaces at-*fpc/pic/port* unit *unit-number*] hierarchy level:

- To specify an ATM interface as the control channel for a Layer 2 circuit trunk, include the control-channel statement at the [edit interfaces at-*fpc/pic/port* unit *unit-number*] hierarchy level.
- To specify a trunk identifier, include the trunk-id statement at the [edit interfaces at-*fpc/pic/port* unit *unit-number*] hierarchy level. Trunk ID values range from 0 through 31.
- To configure the amount of bandwidth reserved for the trunk, include the trunk-bandwidth statement at the [edit interfaces at-*fpc/pic/port* unit *unit-number*] hierarchy level and specify a value from 1,000,000 bps (1 Mbps) through 542,526,792 bps.
- To apply a CoS scheduler map to the trunk, include the atm-scheduler-map statement at the [edit interfaces at-*fpc/pic/port* unit *unit-number*] hierarchy level. This statement must reference an ATM2 IQ interface-based scheduler map at the [edit interfaces at-*fpc/pic/port* atm-options scheduler-maps *map-name*] hierarchy level.

```
interfaces {
   at-fpc/pic/port {
      atm-options {
         pic-type atm2;
         scheduler-maps {
            map-name {
               ...
            }
         }
         unit unit-number {
            atm-scheduler-map map-name;
            control-channel;
            trunk-id id-number;
            trunk-bandwidth bandwidth;
         }
      }
   }
```

```
    }
```

You can configure a variety of CoS-related statements for an ATM2 IQ interface-based scheduler map. To select the CoS mode used for virtual circuits, include the vc-cos-mode statement at the [edit interfaces at-*fpc/pic/port* atm-options scheduler-maps *map-name*] hierarchy level. To specify forwarding class settings, include the priority, transmit-weight, and epd-threshold statements at the [edit interfaces at-*fpc/pic/port* atm-options scheduler-maps *map-name* forwarding-class *class-name*] hierarchy level. For more information about CoS, see the *JUNOS Class of Service Configuration Guide*.

```
[edit]
interfaces {
   at-fpc/pic/port {
      atm-options {
         pic-type atm2;
         scheduler-maps {
            map-name {
               vc-cos-mode (alternate | strict);
               forwarding-class  class-name {
                  priority (high | low);
                  transmit-weight (cells number-of-cells | percent percentage);
                  epd-threshold plp0-threshold  plp1 plp1-threshold;
               }
            }
         }
      }
   }
}
```

Option: Reserving LSP Bandwidth for a Layer 2 Circuit

You can specify the amount of bandwidth in bytes per second that must be available on an LSP for a specific Layer 2 circuit. By using a bandwidth constraint for a Layer 2 circuit, the router performs a type of call admission control. If an LSP exists that contains the required bandwidth, the Layer 2 circuit is established. If the bandwidth is not available on an LSP, the Layer 2 circuit is not established.

To configure bandwidth requirements for a Layer 2 circuit, include the bandwidth statement at the [edit protocols mpls label-switched-path *lsp-name*] hierarchy level and the [edit protocols l2circuit neighbor *ip-address* interface *interface-name*] hierarchy level.

```
[edit]
protocols {
   mpls {
      label-switched-path lsp-name {
         bandwidth traffic-class bytes-per-second;
      }
   }
   l2circuit {
      neighbor ip-address {
         interface interface-name {
            bandwidth bytes-per-second;
         }
```

```
        }
    }
}
```

You can also assign minimum bandwidth requirements for class-of-service (CoS) queues within a Layer 2 circuit and its corresponding LSP. Class type (CT) queues 0, 1, 2, and 3 in a Layer 2 circuit or LSP correspond to the standard four CoS queues available on M Series, MX Series, and T Series routers. To enable mapping of class type queues to the standard CoS queues, include the **bandwidth-model** statement at the [**edit protocols mpls diffserv-te**] hierarchy level. To assign specific bandwidth requirements to each class type queue, include the **ct0**, **ct1**, **ct2**, and **ct3** statements at the [**edit protocols mpls label-switched-path** *lsp-name* **bandwidth**] hierarchy level and the [**edit protocols l2circuit neighbor** *ip-address* **interface** *interface-name* **bandwidth**] hierarchy level.

```
[edit]
protocols {
    mpls {
        diffserv-te {
            bandwidth-model extended-mam;
        }
        label-switched-path lsp-name {
            bandwidth {
                ct0 100m;
                ct1 100m;
                ct2 50m;
                ct3 5m;
            }
        }
    }
    l2circuit {
        neighbor ip-address {
            interface interface-name {
                bandwidth {
                    ct0 100m;
                    ct1 100m;
                    ct2 50m;
                    ct3 5m;
                }
            }
        }
    }
}
```

For more information about class of service, see the *JUNOS Class of Service Configuration Guide*.

Option: Selecting an MTU for a Layer 2 Circuit

To configure the MTU for each individual Layer 2 circuit, include the **mtu** statement at the [**edit protocols l2circuit neighbor** *ip-address* **interface** *interface-name*] hierarchy level. If the MTU setting between Layer 2 circuit neighbors does not match, the Layer 2 circuit is torn down.

NOTE: If you configure an MTU value for an ATM cell relay interface on an ATM2 PIC and simultaneously configure an MTU value for a Layer 2 circuit that uses the same ATM2 PIC, the MTU value for the Layer 2 circuit takes precedence when calculating the cell bundle size and is advertised to Layer 2 circuit neighbors.

Option: Configuring Local Interface Switching for a Layer 2 Circuit

You can terminate a Layer 2 circuit locally on an ingress PE router. To configure a locally terminated circuit, include the local-switching statement at the [edit protocols l2circuit] hierarchy level. Select the Layer 2 circuit interfaces you want to connect locally, specify any APS protect interfaces, and configure an end interface. To select the Layer 2 circuit interfaces that are connected locally, include the interface statement at the [edit protocols l2circuit local-switching] hierarchy level. To configure an end interface, include the end-interface statement at the [edit protocols l2circuit local-switching interface *interface-name*] hierarchy level. To specify APS protect interfaces, include the protect-interface statement at the [edit protocols l2circuit local-switching interface *interface-name*] or [edit protocols l2circuit local-switching interface *interface-name* end-interface *interface-name*] hierarchy levels.

```
[edit]
protocols {
    l2circuit {
        local-switching {
            interface interface1 {
                protect-interface interface2;
                end-interface interface3 {
                    protect-interface interface4;
                }
            }
            interface interface5 {
                protect-interface interface6;
                end-interface interface7 {
                    protect-interface interface8;
                }
            }
        }
    }
}
```

Option: Configuring Layer 2 Circuits Simultaneously over RSVP and LDP LSPs

You can configure a Layer 2 circuit simultaneously over an RSVP LSP and an LDP LSP between the same two routers. To accomplish this, do the following:

- Configure two loopback addresses—a primary and a secondary loopback.
- Configure the RSVP LSP using the primary loopback address.

- Configure the LDP LSP using the secondary loopback address. You can accomplish this by advertising the secondary loopback address as a forwarding equivalence class (FEC) in LDP.
- Configure the transport tunnel endpoint (also known as a packet-switched network [PSN] tunnel endpoint in the IETF drafts) to be used for transporting the Layer 2 circuit traffic. To configure the tunnel endpoint, include the psn-tunnel-endpoint statement at the [edit protocols l2circuit] hierarchy level. By default, the tunnel endpoint is the same as the address of the Layer 2 circuit neighbor.

To verify that your configuration is operational, issue the show l2circuit connections command. This command has been enhanced to display tunnel endpoints. For more information about configuring Layer 2 circuits simultaneously over RSVP and LDP LSPs, see the *JUNOS VPNs Configuration Guide*.

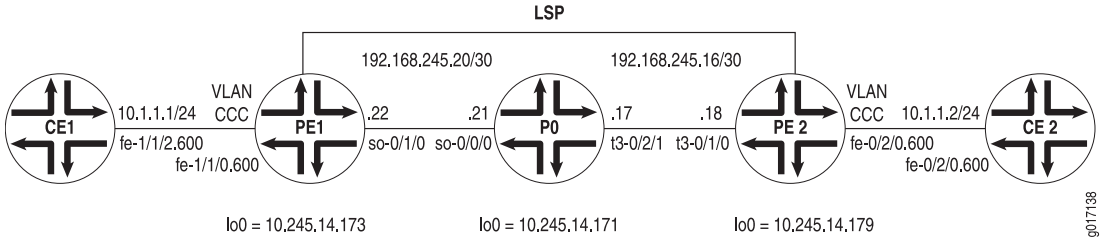
Layer 2 Circuit Configuration Examples

This section contains configuration examples and commands you can issue to verify Layer 2 circuit configurations:

- Example: Ethernet-Based Layer 2 Circuit Configuration on page 303
- Example: SONET/SDH-Based Layer 2 Circuit Configuration on page 310
- Example: ATM2 IQ-Based Layer 2 Circuit Configuration on page 315
- Example: Layer 2 Circuit Traffic Engineering over Multiple LSPs Configuration on page 324
- Example: APS for a Layer 2 Circuit Configuration on page 334

Example: Ethernet-Based Layer 2 Circuit Configuration

Figure 25: Ethernet-Based Layer 2 Circuit Topology Diagram



In Figure 25 on page 303, a Layer 2 circuit is established between routers PE1 and PE2 to deliver Layer 2 traffic between customer routers CE1 and CE2. A Layer 2 circuit VC connection is configured on the PE routers only. No special configuration is required on the CE routers, and the provider core P0 router only requires MPLS and LDP on the appropriate interfaces to enable labels to be shared between the PE routers.

On Router CE1, configure the Fast Ethernet interface to handle VLAN traffic. Be sure to use the same VLAN ID both here and on the Fast Ethernet interface of Router CE2.

Router CE1

```
[edit]
interfaces {
   fe-1/1/2 {
      description "to PE1 fe-1/1/0";
      vlan-tagging;
      unit 600 {
         vlan-id 600; # Be sure this VLAN ID matches the VLAN ID of your CE neighbor.
         family inet {
            address 10.1.1.1/24;
         }
      }
   }
}
```

On Router PE1, configure the Ethernet-based CE-facing interface with the CCC encapsulation type of your choice. If you use VLAN CCC, include the vlan-tagging statement at the [edit interfaces *ethernet-interface-fpc*/*pic* /*port*] hierarchy level. Also, include the encapsulation vlan-ccc statement at both the [edit interfaces *ethernet-interface-fpc*/*pic*/*port*] and [edit interfaces *ethernet-interface-fpc* /*pic*/*port* unit *unit-number*] hierarchy levels.

Establish your Layer 2 circuit with configuration of the l2circuit statement at the [edit protocols] hierarchy level. Remember to include in your Layer 2 circuit configuration the IP address of your remote PE neighbor (usually the loopback address of the neighbor), the interface connected to the CE router, and a virtual circuit identifier for this VC. Then, configure MPLS, LDP, and an IGP (such as OSPF) to enable signaling for your Layer 2 circuit.

Router PE1

```
[edit]
interfaces {
   so-0/1/0 {
      description "to P0 so-0/0/0";
      unit 0 {
         family inet {
            address 192.168.245.22/30;
         }
         family mpls; # Include the MPLS family on core-facing interfaces.
      }
   }
   fe-1/1/0 {
      description "to CE1 fe-1/1/2";
      vlan-tagging;
      encapsulation vlan-ccc; # Configure CCC encapsulation on CE-facing interfaces.
      unit 600 {
         encapsulation vlan-ccc; # Enable this encapsulation on the logical interface.
         vlan-id 600;
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.245.14.173/32;
         }
      }
   }
```

```
}
protocols {
   ospf {
      traffic-engineering;
      area 0.0.0.0 {
         interface so-0/1/0.0;
         interface lo0.0;
      }
   }
   ldp { # LDP is required as the signaling protocol for Layer 2 circuits.
      interface so-0/1/0.0;
      interface lo0.0;# You must include the loopback address in LDP.
   }
   l2circuit {
      neighbor 10.245.14.179 { # This points to the loopback of the PE neighbor.
         interface fe-1/1/0.600 { # Here you include the local CE-facing interface.
            virtual-circuit-id 5; # Be sure this ID matches the ID of your PE neighbor.
         }
      }
   }
}
```

On Router P0, configure LDP, MPLS, and OSPF on the interfaces connected to the PE routers. The core router provides the MPLS backbone needed to tunnel Layer 2 traffic from the ingress PR router to the egress PE router.

Router P0

```
[edit]
interfaces {
   so-0/0/0 {
      description "to PE1 so-0/1/0";
      unit 0 {
         family inet {
            address 192.168.245.21/30;
         }
         family mpls; # Include the MPLS family on core interfaces.
      }
   }
   t3-0/2/1 {
      description "to PE2 t3-0/1/0";
      unit 0 {
         family inet {
            address 192.168.245.17/30;
         }
         family mpls; # Include the MPLS family on core interfaces.
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.245.14.171/32;
         }
      }
   }
}
protocols {
   ospf {
```

```
        traffic-engineering;
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface t3-0/2/1.0;
        }
    }
    ldp { # LDP is required as the signaling protocol for Layer 2 circuits.
        interface so-0/0/0.0;
        interface t3-0/2/1.0;
    }
}
```

On Router PE2, complete the Layer 2 circuit by configuring statements to match those previously set on Router PE1. Configure the Ethernet-based CE-facing interface with CCC encapsulation. Again, you must include the vlan-tagging statement at the [edit interfaces *ethernet-interface-fpc/pic/port*] hierarchy level when you use VLAN CCC. Also, include the encapsulation vlan-ccc statement at both the [edit interfaces *ethernet-interface-fpc/pic/port*] and [edit interfaces *ethernet-interface-fpc/pic/port* unit *unit-number*] hierarchy levels.

Establish your Layer 2 circuit with configuration of the l2circuit statement at the [edit protocols] hierarchy level. Remember to include in your Layer 2 circuit configuration the IP address of your remote PE neighbor (Router PE1), the virtual circuit identifier previously configured on Router PE1, and the interface connected to the CE router. Finally, configure MPLS, LDP, and OSPF to enable signaling for your Layer 2 circuit.

Router PE2

```
[edit]
interfaces {
    t3-0/1/0 {
        description "P0 t3-0/2/1";
        unit 0 {
            family inet {
                address 192.168.245.18/30;
            }
            family mpls; # Include the MPLS family on core-facing interfaces.
        }
    }
    fe-0/2/0 {
        description "to CE2 fe-0/2/0";
        vlan-tagging;
        encapsulation vlan-ccc; # Configure CCC encapsulation on CE-facing interfaces.
        unit 600 {
            encapsulation vlan-ccc; # Enable this encapsulation on the logical interface.
            vlan-id 600;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.245.14.179/32;
            }
        }
    }
}
protocols {
```

```
    ospf {
      traffic-engineering;
      area 0.0.0.0 {
        interface t3-0/1/0.0;
        interface lo0.0;
      }
    }
    ldp { # LDP is required as the signaling protocol for Layer 2 circuits.
      interface t3-0/1/0.0;
      interface lo0.0; # You must include the loopback address in LDP.
    }
    l2circuit {
      neighbor 10.245.14.173 { # This points to the loopback of the PE neighbor.
        interface fe-0/2/0.600 { # Here you include the local CE-facing interface.
          virtual-circuit-id 5; # Be sure this ID matches the ID of your PE neighbor.
        }
      }
    }
}
```

On Router CE2, configure the Fast Ethernet interface to handle VLAN traffic. Be sure to use the same VLAN ID on this interface as the one seen on the Fast Ethernet interface of Router CE1.

Router CE2

```
[edit]
interfaces {
  fe-0/2/0 {
    description "to PE2 fe-0/2/0";
    vlan-tagging;
    unit 600 {
      vlan-id 600; # Be sure this VLAN ID matches the VLAN ID of your CE neighbor.
      family inet {
        address 10.1.1.2/24;
      }
    }
  }
}
```

Verifying Your Work

To verify proper operation of Layer 2 circuits, use the following commands:

- ping mpls l2circuit interface*interface-name*
- ping mpls l2circuit virtual-circuit *virtual-circuit-id* neighbor *ip-address*
- show l2circuit connections
- Options: [brief] | [down] | [extensive] | [history] | [instance] | [local-site] | [remote-site] |[status] | [summary] | [up] | [up-down]
- show ldp database

In addition to displaying bindings for IP prefixes, the show ldp database command also displays the bindings for the Layer 2 FECs.

The following sections show the output of these commands used with the configuration example:

- Router PE1 Status on page 308
- Router P0 Status on page 309
- Router PE2 Status on page 309

Router PE1 Status

```
user@PE1> show l2circuit connections
Layer-2 Circuit Connections:

Legend for connection status (St)   Legend for interface status
EI -- encapsulation invalid         UP -- operational
MM -- mtu mismatch                  Dn -- down
EM -- encapsulation mismatch        NP -- no present
OL -- no outgoing label             DS -- disabled
Dn -- down                          WE -- wrong encapsulation
VC-Dn -- Virtual circuit Down       UN -- uninitialized
UP -- operational
XX -- unknown

Neighbor: 10.245.14.179
    Interface                 Type  St     Time last up          # Up trans
    fe-1/1/0.600 (vc 5)       rmt   Up     Nov 30 00:54:55 2001           1
      Local interface: fe-1/1/0.600, Status: Up,  Encapsulation: VLAN
      Remote PE: 10.245.14.179,  Negotiated control-word: Yes (Null)
      Incoming label: 100007, Outgoing label: 100000

user@PE1>  show ldp database

Input label database, 10.245.14.173:0-10.245.14.171:0
  Label     Prefix
 100019     10.245.14.173/32
 100020     10.245.14.179/32
      3     10.245.14.171/32

Output label database, 10.245.14.173:0-10.245.14.171:0
  Label     Prefix
 100009     10.245.14.179/32
      3     10.245.14.173/32
 100008     10.245.14.171/32

Input label database, 10.245.14.173:0-10.245.14.179:0
  Label     Prefix
 100001     10.245.14.171/32
 100002     10.245.14.173/32
      3     10.245.14.179/32
 100000     L2CKT VLAN VC 5

Output label database, 10.245.14.173:0-10.245.14.179:0
  Label     Prefix
 100009     10.245.14.179/32
      3     10.245.14.173/32
 100008     10.245.14.171/32
 100007     L2CKT VLAN VC 5
```

Router P0 Status

```
user@P0> show ldp database

Input label database, 10.245.14.171:0-10.245.14.173:0
  Label     Prefix
      3     10.245.14.173/32
 100009     10.245.14.179/32
 100008     10.245.14.171/32

Output label database, 10.245.14.171:0-10.245.14.173:0
  Label     Prefix
 100019     10.245.14.173/32
 100020     10.245.14.179/32
      3     10.245.14.171/32

Input label database, 10.245.14.171:0-10.245.14.179:0
  Label     Prefix
 100001     10.245.14.171/32
      3     10.245.14.179/32
 100002     10.245.14.173/32

Output label database, 10.245.14.171:0-10.245.14.179:0
  Label     Prefix
 100019     10.245.14.173/32
 100020     10.245.14.179/32
      3     10.245.14.171/32
```

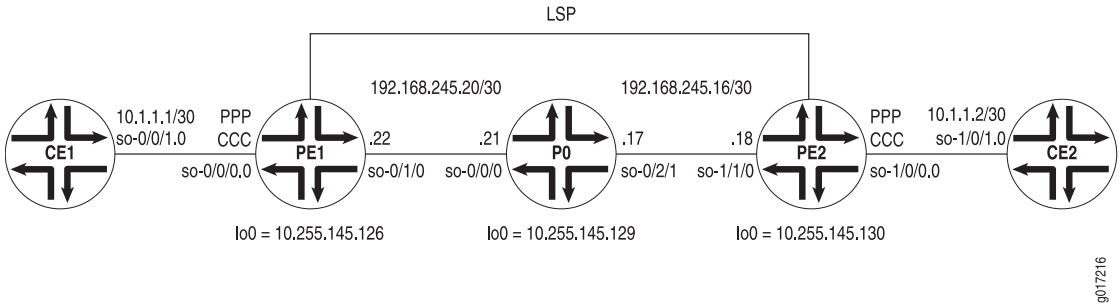
Router PE2 Status

```
user@PE2> show l2circuit connections

Layer-2 Circuit Connections:

Legend for connection status (St)   Legend for interface status
EI -- encapsulation invalid         UP -- operational
MM -- mtu mismatch                  Dn -- down
EM -- encapsulation mismatch        NP -- no present
OL -- no outgoing label             DS -- disabled
Dn -- down                          WE -- wrong encapsulation
VC-Dn -- Virtual circuit Down       UN -- uninitialized
UP -- operational
XX -- unknown

Neighbor: 10.245.14.173
    Interface                 Type  St     Time last up          # Up trans
    fe-0/2/0.600 (vc 5)       rmt   Up     Nov 30 00:54:54 2001           1
      Local interface: fe-0/2/0.600, Status: Up,  Encapsulation: VLAN
      Remote PE: 10.245.14.173,  Negotiated control-word: Yes (Null)
      Incoming label: 100000, Outgoing label: 100007

user@PE2>  show ldp database

Input label database, 10.245.14.179:0-10.245.14.171:0
  Label     Prefix
 100019     10.245.14.173/32
```

```
      3     10.245.14.171/32
 100020     10.245.14.179/32

Output label database, 10.245.14.179:0-10.245.14.171:0
  Label     Prefix
 100001     10.245.14.171/32
 100002     10.245.14.173/32
      3     10.245.14.179/32

Input label database, 10.245.14.179:0-10.245.14.173:0
  Label     Prefix
      3     10.245.14.173/32
 100008     10.245.14.171/32
 100009     10.245.14.179/32
 100007     L2CKT VLAN VC 5

Output label database, 10.245.14.179:0-10.245.14.173:0
  Label     Prefix
 100001     10.245.14.171/32
 100002     10.245.14.173/32
      3     10.245.14.179/32
 100000     L2CKT VLAN VC 5
```

Example: SONET/SDH-Based Layer 2 Circuit Configuration

Figure 26: SONET/SDH-Based Layer 2 Circuit Topology Diagram



In this second Layer 2 circuit example shown in Figure 26 on page 310, you configure a Layer 2 circuit for a SONET/SDH interface encapsulated with PPP.

On Router CE1, configure the SONET/SDH interface to handle PPP traffic. Be sure to use the same IP address prefix both here and on the SONET/SDH interface of Router CE2.

Router CE1

```
[edit]
interfaces {
   so-0/0/1 {
      description "to PE1 so-0/0/0";
      encapsulation ppp;
      unit 0 {
         family inet {
            address 10.1.1.1/30;
         }
      }
   }
```

```
}
```

On Router PE1, configure the PPP-based CE-facing interface with PPP CCC encapsulation. Establish your Layer 2 circuit with configuration of the l2circuit statement at the [edit protocols] hierarchy level. Remember to include in your Layer 2 circuit configuration the IP address of your remote PE neighbor (usually the loopback address of the neighbor), the interface connected to the CE router, and a virtual circuit identifier for this VC. Then, configure MPLS, LDP, and an IGP (such as OSPF) to enable signaling for your Layer 2 circuit.

Router PE1

```
[edit]
interfaces {
   so-0/0/0 {
      description "to CE1 so-0/0/1";
      encapsulation ppp-ccc; # Configure CCC encapsulation on CE-facing interfaces.
      unit 0;
   }
   so-0/1/0 {
      description "to P0 so-0/0/0";
      unit 0 {
         family inet {
            address 192.168.245.22/30;
         }
         family mpls; # Include the MPLS family on core-facing interfaces.
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.255.145.126/32;
         }
      }
   }
}
protocols {
   ospf {
      traffic-engineering;
      area 0.0.0.0 {
         interface so-0/1/0.0;
         interface lo0.0;
      }
   }
   ldp { # LDP is required as the signaling protocol for Layer 2 circuits.
      interface so-0/1/0.0;
      interface lo0.0;
   }
   l2circuit {
      neighbor 10.255.145.130 { # This points to the loopback of the PE neighbor.
         interface so-0/0/0.0 { # Here you include the local CE-facing interface.
            virtual-circuit-id 1; # Be sure this ID matches the ID of your PE neighbor.
         }
      }
   }
}
```

On Router P0, configure LDP, MPLS, and OSPF on the interfaces connected to the PE routers. The core router provides the MPLS backbone needed to tunnel Layer 2 traffic from the ingress PR router to the egress PE router.

Router P0

```
[edit]
interfaces {
   so-0/0/0 {
      description "to PE1 so-0/1/0";
      unit 0 {
         family inet {
            address 192.168.245.21/30;
         }
         family mpls; # Include the MPLS family on core interfaces.
      }
   }
   so-0/2/1 {
      description "to PE2 so-1/1/0";
      unit 0 {
         family inet {
            address 192.168.245.17/30;
         }
         family mpls; # Include the MPLS family on core interfaces.
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.255.145.129/32;
         }
      }
   }
}
protocols {
   ospf {
      traffic-engineering;
      area 0.0.0.0 {
         interface so-0/0/0.0;
         interface so-0/2/1.0;
      }
   }
   ldp { # LDP is required as the signaling protocol for Layer 2 circuits.
      interface so-0/0/0.0;
      interface so-0/2/1.0;
   }
}
```

On Router PE2, complete the Layer 2 circuit by configuring statements to match those previously set on Router PE1. Configure the PPP-based CE-facing interface with PPP CCC encapsulation. Complete your Layer 2 circuit with configuration of the l2circuit statement at the [edit protocols] hierarchy level. Remember to include in your Layer 2 circuit configuration the IP address of your remote PE neighbor (Router PE1), the interface connected to the CE router, and a virtual circuit identifier for this VC. Then, configure MPLS, LDP, and an IGP (such as OSPF) to enable signaling for your Layer 2 circuit.

Router PE2

```
[edit]
interfaces {
   so-1/0/0 {
      description "to CE1 so-1/0/1";
      encapsulation ppp-ccc; # Configure CCC encapsulation on CE-facing interfaces.
      unit 0;
   }
   so-1/1/0 {
      description "to P0 so-0/2/1";
      unit 0 {
         family inet {
            address 192.168.245.18/30;
         }
         family mpls; # Include the MPLS family on core-facing interfaces.
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.255.145.130/32;
         }
      }
   }
}
protocols {
   ospf {
      traffic-engineering;
      area 0.0.0.0 {
         interface so-1/1/0.0;
         interface lo0.0;
      }
   }
   ldp { # LDP is required as the signaling protocol for Layer 2 circuits.
      interface so-1/1/0.0;
      interface lo0.0;
   }
   l2circuit {
      neighbor 10.255.145.126 { # This points to the loopback of the PE neighbor.
         interface so-1/0/0.0 { # Here you include the local CE-facing interface.
            virtual-circuit-id 1; # Be sure this ID matches the ID of your PE neighbor.
         }
      }
   }
}
```

On Router CE2, configure the SONET/SDH interface to handle PPP traffic. Be sure to use the same IP address prefix both here and on the SONET/SDH interface of Router CE1.

Router CE2

```
[edit]
interfaces {
   so-1/0/1 {
      description "to PE2 so-1/0/0";
      encapsulation ppp;
      unit 0 {
         family inet {
```

```
            address 10.1.1.2/30;
          }
        }
      }
    }
```

Verifying Your Work

To verify proper operation of Layer 2 circuits, use the following commands:

- ping mpls l2circuit interface *interface-name*
- ping mpls l2circuit virtual-circuit *virtual-circuit-id* neighbor *ip-address*
- show l2circuit connections
- Options: [brief] | [down] | [extensive] | [history] | [instance] | [local-site] | [remote-site] |[status] | [summary] | [up] | [up-down]
- show ldp database
- show route table

The following section shows the output of these commands used with the configuration example:

```
user@PE1> show l2circuit connections
Layer-2 Circuit Connections:

Legend for connection status (St)
EI -- encapsulation invalid      NP -- interface not present
MM -- mtu mismatch               Dn -- down
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
OL -- no outgoing label          XX -- unknown
NC -- intf encaps not CCC/TCC

Legend for interface status
Up -- operational
Dn -- down

Neighbor: 10.255.145.130
     Interface                Type  St     Time last up          # Up trans
     so-0/0/0.0 (vc 1)        rmt   Up     Jan 26 14:13:54 2003           1
       Local interface: so-0/0/0.0, Status: Up, Encapsulation: PPP
       Remote PE: 10.255.145.130, Negotiated control-word: Yes (Null)
       Incoming label: 100000, Outgoing label: 100000

user@PE1>  show ldp database l2circuit
Input label database, 10.255.145.126:0--10.255.145.130:0
   Label     Prefix
  100000     L2CKT CtrlWord PPP VC 1

Output label database, 10.255.145.126:0--10.255.145.130:0
   Label     Prefix
  100000     L2CKT CtrlWord PPP VC 1

user@PE1> show ldp database l2circuit detail
```

```
Input label database, 10.255.145.126:0--10.255.145.130:0
   Label     Prefix
  100000     L2CKT CtrlWord PPP VC 1
             State: Active
             Age: 5:37

Output label database, 10.255.145.126:0--10.255.145.130:0
   Label     Prefix
  100000     L2CKT CtrlWord PPP VC 1
             State: Active
             Age: 5:37

user@PE1> show route table mpls.0

mpls.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both
0                  *[MPLS/0] 00:05:04, metric 1
                       Receive
1                  *[MPLS/0] 00:05:04, metric 1
                       Receive
2                  *[MPLS/0] 00:05:04, metric 1
                       Receive
100000             *[L2VPN/7] 00:04:50
                     > via so-0/0/0.0, Pop       Offset: 4
100016             *[LDP/9] 00:04:52, metric 1
                     > via so-0/1/0.0, Pop
100016(S=0)        *[LDP/9] 00:04:52, metric 1
                     > via so-0/1/0.0, Pop
so-0/0/0.0         *[L2VPN/7] 00:04:50
                     > via so-0/1/0.0, Push 100000 Offset: -4
```

Example: ATM2 IQ-Based Layer 2 Circuit Configuration

Figure 27: ATM2 IQ-Based Layer 2 Circuit Topology Diagram

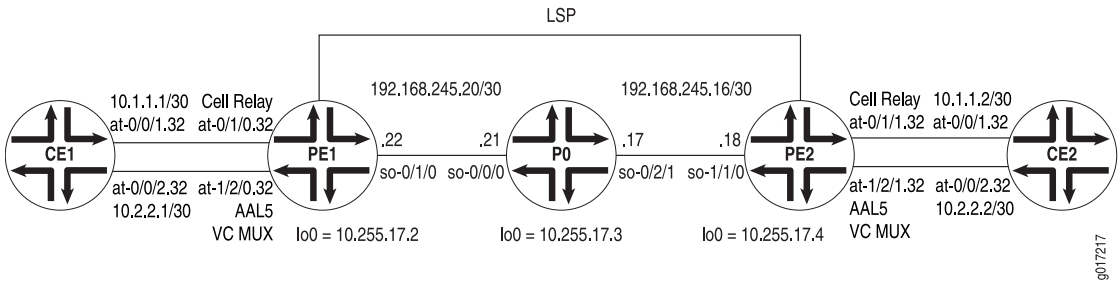


Figure 27 on page 315 shows a similar network topology to our previous two examples. In this example, Routers PE1 and PE2 use ATM cell-relay mode on a CE-facing interface and ATM AAL5 mode on a second CE-facing interface.

On Router CE1, configure the ATM2 IQ interfaces to handle ATM traffic. Interface at-0/0/1 handles standard ATM traffic while interface at-0/0/2 handles AAL5 traffic.

Router CE1

```
[edit]
interfaces {
   at-0/0/1 {
```

```
        description "to PE1 at-0/1/0";
        atm-options {
            pic-type atm2; # Layer 2 circuits are compatible with
            vpi 0; # ATM2 IQ interfaces.
        }
        unit 0 {
            vci 32;
            family inet {
                address 10.1.1.1/30;
            }
        }
    }
    at-0/0/2 {
        description "to PE1 at-1/2/0";
        atm-options {
            pic-type atm2; # Layer 2 circuits are compatible with
            vpi 0; # ATM2 IQ interfaces.
        }
        unit 0 {
            encapsulation atm-vc-mux;
            vci 32;
            family inet {
                address 10.2.2.1/30;
            }
        }
    }
}
```

On Router PE1, configure the ATM2 IQ-based CE-facing interfaces: one with ATM cell-relay mode CCC encapsulation and the other with ATM VC multiplexing CCC encapsulation. Also enable the corresponding Layer 2 circuit modes at the [edit chassis] hierarchy level. In this case, you must configure cell-relay mode on Physical Interface Card (PIC) 1 in Flexible PIC Concentrator (FPC) 0 and AAL5 mode on PIC 2 in FPC 1.

Establish your Layer 2 circuit with configuration of the l2circuit statement at the [edit protocols] hierarchy level. Remember to include in your Layer 2 circuit configuration the IP address of your remote PE neighbor (usually the loopback address of the neighbor), the interfaces connected to the CE router, and a virtual circuit identifier for each VC. In this case, you will establish one VC for cell-relay mode traffic and a second VC for AAL5 traffic. Then, configure MPLS, LDP, and an IGP (such as OSPF) to enable signaling for your Layer 2 circuit.

Router PE1

```
[edit]
chassis {
    fpc 0 {
        pic 1 {
            atm-l2circuit-mode {
                cell; # This dedicates FPC 0 PIC 1 to cell-relay mode.
            }
        }
    }
    fpc 1 {
        pic 2 {
            atm-l2circuit-mode {
```

```
            aal5; # This dedicates FPC 1 PIC 2 to AAL5 mode.
          }
        }
      }
    }
    interfaces {
      at-0/1/0 {
        description "to CE1 at-0/0/1";
        encapsulation atm-ccc-cell-relay; # Cell-relay requires cell-relay encapsulation.
        atm-options {
          cell-bundle-size 4; # This sets the cell bundle size for the interface.
          pic-type atm2; # Layer 2 circuits are compatible with
          vpi 0; # ATM2 IQ interfaces.
        }
        unit 0 {
          encapsulation atm-ccc-cell-relay; # Encapsulation for the logical interface.
          vci 32;
          cell-bundle-size 10; # The cell bundle size for the logical interface overrides
        }      # the physical interface setting.
      }
      at-1/2/0 {
        description "to CE1 at-0/0/2";
        atm-options {
          pic-type atm2; # Layer 2 circuits are compatible with
          vpi 0; # ATM2 IQ interfaces.
        }
        unit 0 {
          encapsulation atm-ccc-vc-mux; # AAL5 requires CCC VC MUX encapsulation.
          vci 32;
        }
      }
      so-0/1/0 {
        description "to P0 so-0/0/0";
        unit 0 {
          family inet {
            address 192.168.245.22/30;
          }
          family mpls; # Include the MPLS family on core-facing interfaces.
        }
      }
      lo0 {
        unit 0 {
          family inet {
            address 10.255.17.2/32;
          }
        }
      }
    }
    protocols {
      ospf {
        traffic-engineering;
        area 0.0.0.0 {
          interface so-0/1/0.0;
          interface lo0.0;
        }
      }
```

```
    ldp { # LDP is required as the signaling protocol for Layer 2 circuits.
      interface so-0/1/0.0;
      interface lo0.0;
    }
    l2circuit {
      neighbor 10.255.17.4 { # This points to the loopback of the PE neighbor.
        interface at-0/1/0.32 { # Here you include the local CE-facing interface.
          virtual-circuit-id 1; # Be sure this ID matches the ID of your PE neighbor.
        }
      }
      neighbor 10.255.17.4 { # This points to the loopback of the PE neighbor.
        interface at-1/2/0.32 { # Here you include the local CE-facing interface.
          virtual-circuit-id 2; # Be sure this ID matches the ID of your PE neighbor.
        }
      }
    }
  }
```

On Router P0, configure LDP, MPLS, and OSPF on the interfaces connected to the PE routers. The core router provides the MPLS backbone needed to tunnel Layer 2 traffic from the ingress PR router to the egress PE router.

Router P0

```
[edit]
interfaces {
  so-0/0/0 {
    description "to PE1 so-0/1/0";
    unit 0 {
      family inet {
        address 192.168.245.21/30;
      }
      family mpls; # Include the MPLS family on core interfaces.
    }
  }
  so-0/2/1 {
    description "to PE2 so-1/1/0";
    unit 0 {
      family inet {
        address 192.168.245.17/30;
      }
      family mpls; # Include the MPLS family on core interfaces.
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.255.17.3/32;
      }
    }
  }
}
protocols {
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface so-0/0/0.0;
      interface so-0/2/1.0;
```

```
        }
    }
    ldp { # LDP is required as the signaling protocol for Layer 2 circuits.
        interface so-0/0/0.0;
        interface so-0/2/1.0;
    }
}
```

On Router PE2, complete the Layer 2 circuit by configuring statements to match those previously set on Router PE1.

Configure the ATM2 IQ-based CE-facing interfaces: one with ATM cell-relay mode CCC encapsulation and the other with ATM VC multiplexing CCC encapsulation. Also enable the corresponding Layer 2 circuit modes at the [edit chassis] hierarchy level. In this case, you must configure cell-relay mode on PIC 1 in FPC 0 and AAL5 mode on PIC 2 in FPC 1.

Complete your Layer 2 circuit with configuration of the l2circuit statement at the [edit protocols] hierarchy level. Remember to include in your Layer 2 circuit configuration the IP address of your remote PE neighbor (Router PE1), the interfaces connected to the CE router, and a virtual circuit identifier for each VC. In this case, you will establish one VC for cell-relay mode traffic and a second VC for AAL5 traffic. Then, configure MPLS, LDP, and an IGP (such as OSPF) to enable signaling for your Layer 2 circuit.

Router PE2

```
[edit]
chassis {
    fpc 0 {
        pic 1 {
            atm-l2circuit-mode {
                cell; # This dedicates FPC 0 PIC 1 to cell-relay mode.
            }
        }
    }
    fpc 1 {
        pic 2 {
            atm-l2circuit-mode {
                aal5; # This dedicates FPC 1 PIC 2 to AAL5 mode.
            }
        }
    }
}
interfaces {
    at-0/1/1 {
        description "to CE2 at-1/0/1";
        encapsulation atm-ccc-cell-relay; # Cell-relay requires cell-relay encapsulation.
        atm-options {
            cell-bundle-size 4; # This sets the cell bundle size for the physical interface.
            pic-type atm2; # Layer 2 circuits are compatible with
            vpi 0; # ATM2 IQ interfaces.
        }
        unit 0 {
            encapsulation atm-ccc-cell-relay; # Also configure the encapsulation
            vci 32; # on the logical interface.
            cell-bundle-size 10; # The cell bundle size for the logical interface overrides
```

```
        }         # the physical interface setting.
    }
    at-1/2/1 {
        description "to CE2 at-1/0/2";
        atm-options {
            pic-type atm2; # Layer 2 circuits are compatible with
            vpi 0; # ATM2 IQ interfaces.
        }
        unit 0 {
            encapsulation atm-ccc-vc-mux; # AAL5 requires CCC VC MUX encapsulation.
            vci 32;
        }
    }
    so-1/1/0 {
        description "to P0 so-0/2/1";
        unit 0 {
            family inet {
                address 192.168.245.18/30;
            }
            family mpls; # Include the MPLS family on core-facing interfaces.
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.17.4/32;
            }
        }
    }
}
protocols {
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface so-1/1/0.0;
            interface lo0.0;
        }
    }
    ldp {    # LDP is required as the signaling protocol for Layer 2 circuits.
        interface so-1/1/0.0;
        interface lo0.0;
    }
    l2circuit {
        neighbor 10.255.17.2 { # This points to the loopback of the PE neighbor.
            interface at-0/1/1.32 { # Here you include the local CE-facing interface.
                virtual-circuit-id 1; # Be sure this ID matches the ID of your PE neighbor.
            }
        }
        neighbor 10.255.17.2 { # This points to the loopback of the PE neighbor.
            interface at-1/2/1.32 { # Here you include the local CE-facing interface.
                virtual-circuit-id 2; # Be sure this ID matches the ID of your PE neighbor.
            }
        }
    }
}
```

On Router CE2, configure the ATM2 IQ interfaces to handle ATM traffic. Interface at-1/0/1 handles standard ATM traffic while interface at-1/0/2 handles AAL5 traffic.

Router CE2

```
[edit]
interfaces {
   at-1/0/1 {
      description "to PE2 at-0/1/1";
      atm-options {
         pic-type atm2; # Layer 2 circuits are compatible with
         vpi 0;    # ATM2 IQ interfaces.
      }
      unit 0 {
         vci 32;
         family inet {
            address 10.1.1.2/30;
         }
      }
   }
   at-1/0/2 {
      description "to PE2 at-1/2/1";
      atm-options {
         pic-type atm2; # Layer 2 circuits are compatible with
         vpi 0;   # ATM2 IQ interfaces.
      }
      unit 0 {
         encapsulation atm-vc-mux;
         vci 32;
         family inet {
            address 10.2.2.2/30;
         }
      }
   }
}
```

Verifying Your Work

To verify proper operation of Layer 2 circuits, use the following commands:

- ping mpls l2circuit interface *interface-name*
- ping mpls l2circuit virtual-circuit *virtual-circuit-id* neighbor *ip-address*
- show l2circuit connections
- Options: [brief] | [down] | [extensive] | [history] | [instance] | [local-site] | [remote-site] |[status] | [summary] | [up] | [up-down]
- show interfaces
- show route table l2circuit.0
- show ldp database l2circuit detail

This is what the operational command output looks like for cell-relay mode on Router PE1:

```
user@PE1> show l2circuit connections
Layer-2 Circuit Connections:

Legend for connection status (St)
EI -- encapsulation invalid      NP -- interface not present
MM -- mtu mismatch               Dn -- down
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
OL -- no outgoing label          XX -- unknown
NC -- intf encaps not CCC/TCC

Legend for interface status
Up -- operational
Dn -- down

Neighbor: 10.255.17.4
    Interface                 Type  St     Time last up          # Up trans
    at-0/1/0.0 (vc 32)        rmt   Up     Jan 22 15:15:52 2003           1
      Local interface: at-0/1/0.0, Status: Up, Encapsulation: ATM CELL (VC Mode)

      Remote PE: 10.255.17.4, Negotiated control-word: Yes (Non-null)
      Incoming label: 100000, Outgoing label: 100000

user@PE1>  show route table l2circuit.0 detail

l2circuit.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
10.255.17.4:CtrlWord:9:32:Local/96 (1 entry, 1 announced)
         *L2CKT   Preference: 7
                Next hop: via so-0/2/0.0 weight 1, selected
                Label-switched-path PE1-PE2
                Protocol next hop: 10.255.17.4 Indirect next hop: 85135e8 367
                State: <Active Int>
                Local AS:    69
                Age: 2:34
                 Task: l2 circuit
                Announcement bits (1): 0-LDP
                AS path: I
                VC Label 100000, MTU 0,  cell-bundle size 80

10.255.17.4:CtrlWord:9:32:Remote/96 (1 entry, 1 announced)
         *LDP     Preference: 9
                Next hop type: Discard
                State: <Active Int>
                Local AS:    69
                Age: 28:11
                 Task: LDP
                Announcement bits (1): 1-l2 circuit
                AS path: I
                VC Label 100000, MTU 0,  cell-bundle size 80

user@PE1>  show interfaces at-0/1/0.0 extensive
  Logical interface at-0/1/0.0 (Index 66) (SNMP ifIndex 40) (Generation 4)
    Flags: Point-To-Point SNMP-Traps  Encapsulation: ATM-CCC-Cell-Relay
     L2 circuit cell bundle size: 10 , bundle timeout: 125 usec, timeout count:
0
    L2 circuit out-of-sequence count: 0
    Traffic statistics:
[...]

user@PE1>  show interfaces media at-0/1/0
```

```
Physical interface: at-0/1/0, Enabled, Physical link is Up
 Interface index: 154, SNMP ifIndex: 50
 Link-level type: ATM-CCC-Cell-Relay,  MTU: 4482, Clocking: Internal, SONET mode,

 L2 circuit mode: Cell, Speed: OC12, Loopback: None
[...]

user@PE1>  show ldp database l2circuit detail
Input label database, 10.255.17.2:0--10.255.17.4:0
    Label     Prefix
   100000     L2CKT CtrlWord ATM CELL (VC Mode) VC 32
              Cell bundle size: 80
              State: Active
              Age: 9:48

Output label database, 10.255.17.2:0--10.255.17.4:0
    Label     Prefix
   100000     L2CKT CtrlWord ATM CELL (VC Mode) VC 32
              Cell bundle size: 80
              State: Active
              Age: 9:48
```

This is what the operational command output looks like on Router PE1 if AAL5 mode is used:

```
user@PE1> show l2circuit connections
Layer-2 Circuit Connections:

Legend for connection status (St)
EI -- encapsulation invalid      NP -- interface not present
MM -- mtu mismatch               Dn -- down
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
OL -- no outgoing label          XX -- unknown
NC -- intf encaps not CCC/TCC

Legend for interface status
Up -- operational
Dn -- down

Neighbor: 10.255.17.4
    Interface                 Type  St     Time last up          # Up trans
    at-1/2/0.0 (vc 32)        rmt   Up     Feb 18 18:00:00 2003           1
      Local interface: at-1/2/0.0, Status: Up, Encapsulation: ATM AAL5
      Remote PE: 10.255.17.4, Negotiated control-word: Yes (Non-null)
      Incoming label: 100016, Outgoing label: 100032

user@PE1> show interfaces media at-0/1/0
Physical interface: at-0/1/0, Enabled, Physical link is Up
 Interface index: 154, SNMP ifIndex: 50
 Link-level type: ATM-PVC, MTU: 4482, Clocking: Internal, SONET mode,
 L2 circuit mode: AAL5, Speed: OC12, Loopback: None, Payload scrambler: Enabled
[...]

user@PE1>  show interfaces at-1/2/0.0 extensive
  Logical interface at-1/2/0.0 (Index 68) (SNMP ifIndex 40) (Generation 38)
    Flags: Point-To-Point SNMP-Traps  Encapsulation: ATM-CCC-VCMUX
    L2 circuit out-of-sequence count: 0
    Traffic statistics:[...]
```

Example: Layer 2 Circuit Traffic Engineering over Multiple LSPs Configuration

Figure 28: Layer 2 Circuit Traffic Engineering Topology Diagram

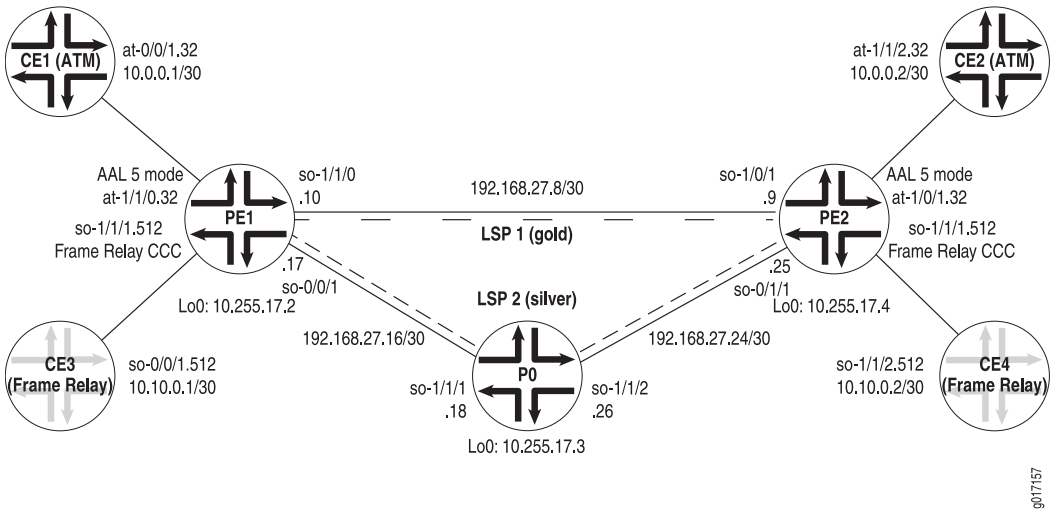


Figure 28 on page 324 shows a network topology designed to traffic engineer different Layer 2 circuits over select LSPs. Across provider edge routers PE1 and PE2, an ATM AAL5 mode Layer 2 circuit connects customer edge routers CE1 and CE2, and a Frame Relay Layer 2 circuit connects routers CE3 and CE4. To maintain traffic separation, the ATM traffic is mapped onto LSP1 with a community named gold, and the Frame Relay traffic is mapped onto LSP2 with a community named silver. LSP1 takes the direct route between routers PE1 and PE2, while LSP2 travels from Router PE1 to PE2 through Router P0.

In addition to traffic engineering, you can send Layer 2 control information in the control word of a Layer 2 circuit. In this case, Frame Relay discard eligible (DE), forward explicit congestion notification (FECN), and backward explicit congestion notification (BECN) information is mapped into the control word. Likewise, ATM cell loss priority (CLP) and explicit forward congestion indicator (EFCI) information is mapped into the control word.

To traffic engineer Layer 2 circuits over multiple LSPs, you assign a set of Layer 2 circuits to a community and then apply a policy to send the community traffic over a desired LSP. To create communities, include the community *community-name* statement at the [edit policy-options] hierarchy level. To assign a Layer 2 circuit to a community, include the community *community-name* statement at the [edit protocols l2circuit neighbor *neighbor-id* interface *interface-name*] hierarchy level. To send community traffic over a specific LSP, include the community *community-name* statement at the [edit policy-options policy-statement *policy-name* term *term-name* from] hierarchy level and the install-next-hop lsp *lsp-name* statement at the [edit policy-options policy-statement *policy-name* term *term-name* then] hierarchy level.

On Router CE1, configure the ATM2 IQ interface at-0/0/1.32 to handle ATM AAL5 traffic:

Router CE1

```
[edit]
interfaces {
   at-0/0/1 {
      description "to PE1 at-1/1/0";
      atm-options {
         pic-type atm2; # Layer 2 circuits are compatible with
         vpi 0; # ATM2 IQ interfaces.
      }
      unit 0 {
         encapsulation atm-vc-mux; # Use ATM VC MUX encapsulation on the CE.
         point-to-point;
         vci 0.32;
         family inet {
            address 10.0.0.1/30;
         }
      }
   }
}
```

On Router CE3, configure the SONET/SDH interface at so-0/0/1 to handle Frame Relay traffic:

Router CE3

```
[edit]
interfaces
so-0/0/1 {
   description “to PE1 so-1/1/1”
   encapsulation frame-relay; # Use Frame Relay encapsulation on the CE router.
   unit 0 {
      encapsulation frame-relay;
      point-to-point;
      dlci 512;
      family inet {
         address 10.10.0.1/30;
      }
   }
}
```

On Router PE1, configure the ATM2 IQ-based CE1-facing interface at-1/1/0 with ATM VC multiplexing CCC encapsulation on the logical interface. Also enable the corresponding Layer 2 circuit modes at the [edit chassis] hierarchy level. In this case, you must configure AAL5 mode on PIC 1 in FPC 1. Once you configure the ATM2 IQ-based Layer 2 circuit, the CLP and EFCI bits are mapped to the control word by default.

Next, configure the Frame Relay interface so-1/1/1 with Frame Relay CCC encapsulation on both the physical and logical interface. Map the DE, FECN, and BECN bits to the control word with the translate-fecn-and-becn and translate-discard-eligible statements at the [edit interfaces so-*fpc/pic/port* unit *unit-number* family ccc] hierarchy level.

Establish your Layer 2 circuits with configuration of the l2circuit statement at the [edit protocols] hierarchy level. Remember to include in your Layer 2 circuit configuration the IP address of your remote PE neighbor (usually the loopback address of the neighbor), the interfaces connected to the CE router, and a virtual circuit identifier for each VC. In this case, you will establish one VC for ATM AAL5 traffic

and a second VC for Frame Relay traffic. Then, configure MPLS, LDP, and an IGP (such as OSPF) to enable signaling for your Layer 2 circuit. Two LSPs are established for the ATM and Frame Relay traffic: LSP1 for ATM traffic going directly to Router PE2 and LSP 2 for Frame Relay traffic going through Router P0 before going on to Router PE2.

Finally, configure a community for traffic separation for the ATM and Frame Relay Layer 2 circuits. Assign community gold to the ATM VC and community silver to the Frame Relay VC. Remember to give the communities numerical values and configure a routing policy to match the communities to specific LSPs. This policy is applied as an export policy for the forwarding table at the [edit routing-options] hierarchy level.

Router PE1

```
[edit]
chassis {
    fpc 1 {
        pic 1 {
            atm-l2circuit-mode {
                aal5; # This dedicates FPC 1 PIC 1 to AAL5 mode.
            }
        }
    }
}
interfaces {
    at-1/1/0 {
        description "to CE1 at-0/0/2";
        atm-options {
            pic-type atm2; # Layer 2 circuits are compatible with
            vpi 0; # ATM2 IQ interfaces.
        }
        unit 0 {
            encapsulation atm-ccc-vc-mux; # CLP/EFCI bits are mapped to control word.
            vci 0.32;
        }
    }
    so-0/1/0 {
        description "to P0 so-0/0/0";
        unit 0 {
            family inet {
                address 192.168.27.17/30;
            }
            family mpls; # Include the MPLS family on core-facing interfaces.
        }
    }
    so-1/1/0 {
        description "to PE2 so-1/0/1";
        unit 0 {
            family inet {
                address 192.168.27.10/30;
            }
            family mpls; # Include the MPLS family on core-facing interfaces.
        }
    }
    so-1/1/1 {
        description "to CE3 so-0/0/1";
        dce;
```

```
        encapsulation frame-relay-ccc;
        unit 0 {
            encapsulation frame-relay-ccc;
            point-to-point;
            dlci 512;
            family ccc {
                translate-fecn-and-becn; # Option to map FECN/BECN bits to control word.
                translate-discard-eligible; # Option to map DE bit to control word.
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.17.2/32;
            }
        }
    }
}
routing-options {
    forwarding-table {
        export layer2communities; # This applies communities to the Layer 2 circuits.
    }
}
protocols {
    mpls {
        label-switched-path lsp1 { # ATM LSP 1 goes directly to PE2.
            to 10.255.17.4;
            primary direct;
        }
        label-switched-path lsp2 { # Frame Relay LSP 2 goes through P0.
            to 10.255.17.4;
            primary thruP0;
        }
        path direct {
            192.168.27.9 strict;
        }
        path thruP0 {
            192.168.27.18 strict;
            192.168.27.25 strict;
        }
        interface so-0/0/1.0;
        interface so-1/1/0.0;
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface so-0/0/1.0;
            interface so-1/1/0.0;
            interface lo0.0;
        }
    }
    ldp { # LDP is required as the signaling protocol for Layer 2 circuits.
        interface so-0/0/1.0;
        interface so-1/1/0.0;
        interface lo0.0;
```

```
    }
    l2circuit {
      neighbor 10.255.17.4 {# This points to the loopback of the PE neighbor.
        interface at-1/1/0.32 { # Here you include the local CE-facing interface.
          virtual-circuit-id 1; # Be sure this ID matches the ID of your PE neighbor.
          community gold;# Assigns the ATM Layer 2 circuit to the gold community.
        }
      }
      neighbor 10.255.17.4 {# This points to the loopback of the PE neighbor.
        interface so-1/1/1.512 { # Here you include the local CE-facing interface.
          virtual-circuit-id 2; # Be sure this ID matches the ID of your PE neighbor.
          community silver; # Assigns the Frame Relay Layer 2 circuit to silver.
        }
      }
    }
  }
  policy-options {
    policy-statement layer2communities { # Here you map the communities to LSPs.
      term 10 {
        from community gold;  # Apply community gold to LSP 1.
        then {
          install-nexthop lsp lsp1;
          accept;
        }
        term 20 {
          from community silver; # Apply community silver to LSP 2.
          then {
            install-nexthop lsp lsp2;
            accept;
          }
          community gold members 103:1; # Assign numerical value to community
            gold.
          community silver members 103:2; # Assign numerical value to community
            silver.
        }
      }
    }
  }
```

On Router P0, configure LDP, MPLS, and OSPF on the interfaces connected to the PE routers. The core router provides the MPLS backbone needed to tunnel Layer 2 traffic from the ingress PR router to the egress PE router. Only LSP 2 for Frame Relay passes through Router P0.

Router P0

```
[edit]
interfaces {
  so-1/1/1 {
    description "to PE1 so-0/0/1";
    unit 0 {
      family inet {
        address 192.168.27.18/30;
      }
      family mpls; # Include the MPLS family on core interfaces.
    }
  }
  so-1/1/2 {
```

```
        description "to PE2 so-0/1/1";
        unit 0 {
          family inet {
            address 192.168.27.26/30;
          }
          family mpls; # Include the MPLS family on core interfaces.
        }
      }
      lo0 {
        unit 0 {
          family inet {
            address 10.255.17.3/32;
          }
        }
      }
    }
    protocols {
      mpls {
        interface so-1/1/1.0;
        interface so-1/1/2.0;
      }
      ospf {
        traffic-engineering;
        area 0.0.0.0 {
          interface so-1/1/1.0;
          interface so-1/1/2.0;
        }
      }
      ldp {# LDP is required as the signaling protocol for Layer 2 circuits.
        interface so-1/1/1.0;
        interface so-1/1/2.0;
      }
    }
```

On Router PE2, complete the Layer 2 circuit by configuring statements to match those previously set on Router PE1.

Establish your Layer 2 circuits with configuration of the l2circuit statement at the [edit protocols] hierarchy level. Remember to include in your Layer 2 circuit configuration the IP address of your remote PE neighbor (usually the loopback address of the neighbor), the interfaces connected to the CE router, and a virtual circuit identifier for each VC. In this case, you will establish one VC for ATM AAL5 traffic and a second VC for Frame Relay traffic. Then, configure MPLS, LDP, and an IGP (such as OSPF) to enable signaling for your Layer 2 circuit. Two LSPs are established for the ATM and Frame Relay traffic: LSP1 for ATM traffic going directly to Router PE2 and LSP 2 for Frame Relay traffic going through Router P0 before going on to Router PE2.

Finally, configure a community for traffic separation for the ATM and Frame Relay Layer 2 circuits. The ATM VC has community gold and the Frame Relay VC has community silver. Remember to give the communities numerical values and configure a routing policy to match the communities to specific LSPs. This policy is applied as an export policy for the forwarding table at the [edit routing-options] hierarchy level.

Router PE2 [edit]

```
chassis {
   fpc 1 {
      pic 0 {
         atm-l2circuit-mode {
            aal5; # This dedicates FPC 1 PIC 0 to AAL5 mode.
         }
      }
   }
}
interfaces {
   at-1/0/1 {
      description "to CE2 at-1/1/2";
      atm-options {
         pic-type atm2; # Layer 2 circuits are compatible with
         vpi 0; # ATM2 IQ interfaces.
      }
      unit 0 {
         encapsulation atm-ccc-vc-mux; # CLP and EFCI appear in the control word.
         vci 0.32;
      }
   }
   so-0/1/1 {
      description "to P0 so-1/1/2";
      unit 0 {
         family inet {
            address 192.168.27.25/30;
         }
         family mpls; # Include the MPLS family on core-facing interfaces.
      }
   }
   so-1/0/1 {
      description "to PE1 so-1/1/0";
      unit 0 {
         family inet {
            address 192.168.27.9/30;
         }
         family mpls; # Include the MPLS family on core-facing interfaces.
      }
   }
   so-1/1/1 {
      description "to CE4 so-1/1/2";
      dce;
      encapsulation frame-relay-ccc;
      unit 0 {
         encapsulation frame-relay-ccc;
         point-to-point;
         dlci 512;
         family ccc {
            translate-fecn-and-becn; # Option to map FECN/BECN bits to control word.
            translate-discard-eligible; # Option to map DE bit to control word.
         }
      }
   }
   lo0 {
      unit 0 {
         family inet {
```

```
            address 10.255.17.4/32;
          }
        }
      }
    }
    routing-options {
      forwarding-table {
        export layer2communities; # This maps communities to the Layer 2 circuits.
      }
    }
    protocols {
      mpls {
        label-switched-path lsp1 { # ATM LSP 1 goes directly to Router PE2.
          to 10.255.17.2;
          primary direct;
        }
        label-switched-path lsp2 { # Frame Relay LSP 2 goes through Router P0.
          to 10.255.17.2;
          primary thruP0;
        }
        path direct {
          192.168.27.10 strict;
        }
        path thruP0 {
          192.168.27.26 strict;
          192.168.27.17 strict;
        }
        interface so-0/1/1.0;
        interface so-1/0/1.0;
      }
      ospf {
        traffic-engineering;
        area 0.0.0.0 {
          interface so-0/1/1.0;
          interface so-1/0/1.0;
          interface lo0.0;
        }
      }
      ldp { # LDP is required as the signaling protocol for Layer 2 circuits.
        interface so-0/1/1.0;
        interface so-1/0/1.0;
        interface lo0.0;
      }
      l2circuit {
        neighbor 10.255.17.2 {# This points to the loopback of the PE neighbor.
          interface at-1/0/1.32 { # Here you include the local CE-facing interface.
            virtual-circuit-id 1; # Be sure this ID matches the ID of your PE neighbor.
            community gold;# Assigns the ATM Layer 2 circuit to the gold community.
          }
        }
        neighbor 10.255.17.2 {# This points to the loopback of the PE neighbor.
          interface so-1/1/1.512 { # Here you include the local CE-facing interface.
            virtual-circuit-id 2; # Be sure this ID matches the ID of your PE neighbor.
            community silver; # Assigns the Frame Relay Layer 2 circuit to silver.
          }
        }
```

```
    }
  }
  policy-options {
    policy-statement layer2communities { # Here you map communities to LSPs.
      term 10 {
        from community gold; # Apply community gold to LSP 1.
        then {
          install-nexthop lsp lsp1;
          accept;
        }
        term 20 {
          from community silver; # Apply community silver to LSP 2.
          then {
            install-nexthop lsp lsp2;
            accept;
          }
          community gold members 103:1; # Assign numerical value to community
            gold.
          community silver members 103:2; # Assign numerical value to community
            silver.
        }
      }
    }
  }
```

On Router CE2, configure the ATM2 IQ interfaces to handle ATM traffic. Interface at-1/0/1 handles AAL5 traffic.

Router CE2

```
[edit]
interfaces {
  at-1/1/2 {
    description "to PE2 at-1/0/1";
    atm-options {
      pic-type atm2; # Layer 2 circuits are compatible with
      vpi 0; # ATM2 IQ interfaces.
    }
    unit 0 {
      encapsulation atm-vc-mux; # Use ATM VC MUX encapsulation on the CE.
      point-to-point;
      vci 0.32;
      family inet {
        address 10.0.0.2/30;
      }
    }
  }
}
```

On Router CE4, configure the SONET/SDH interface at so-1/1/2 to handle Frame Relay traffic:

Router CE3

```
[edit]
interfaces {
  so-1/1/2 {
    description " to PE2 so-1/1/1";
    encapsulation frame-relay-ccc; # Use Frame Relay encapsulation on the CE.
```

```
        unit 0 {
          encapsulation frame-relay-ccc;
          point-to-point;
          dlci 512;
          family inet {
            address 10.10.0.2/30;
          }
        }
      }
    }
```

Verifying Your Work

To verify proper operation of traffic engineered Layer 2 circuits, use the following command:

```
show route table mpls.0 detail
```

On Router PE1, you can see that ATM traffic is part of the gold community that has a value of 103:1 and is associated with LSP 1. Likewise, Frame Relay traffic is part of the silver community that has a value of 103:2 and is associated with LSP 2:

```
user@PE1> show route table mpls.0 detail
mpls.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
at-1/1/0.32  (1 entry, 1 announced)
          *L2CKT  Preference: 7
                  Next hop: 192.168.27.9 via so-1/1/0.0 weight 1, selected
                 Label-switched-path lsp1
                 Label operation: Push 100032 Offset: -4
                 Next hop: via so-0/0/1.0 weight 1
                 Label-switched-path lsp2
                 Label operation: Push 100032 Offset: -4
                 Protocol next hop: 10.255.17.4
                 Push 100032 Offset: -4
                  Indirect next hop: 8576bd0 300
                 State: <Active Int>
                 Age: 7:18
                 Task: Common L2 VC
                 Announcement bits (2): 0-KRT 1-Common L2 VC
                 AS path: I
                 Communities: 103:1 # This is the  gold  community.

so-1/1/1.512 (1 entry, 1 announced)
          *L2CKT  Preference: 7
                  Next hop: 192.168.27.9 via so-1/1/0.0 weight 1
                  Label-switched-path lsp1
                  Label operation: Push 100048 Offset: -4
                  Next hop: via so-0/0/1.0 weight 1, selected
                  Label-switched-path lsp2
                  Label operation: Push 100048 Offset: -4
                  Protocol next hop: 10.255.17.4
                  Push 100048 Offset: -4
                   Indirect next hop: 860f1f8 293
                  State: <Active Int>
                  Age: 5:15
                  Task: Common L2 VC
                  Announcement bits (2): 0-KRT 1-Common L2 VC
```

```
AS path: I
Communities: 103:2 # This is the  silver  community.
```

Example: APS for a Layer 2 Circuit Configuration

Figure 29: APS for a Layer 2 Circuit Topology Diagram

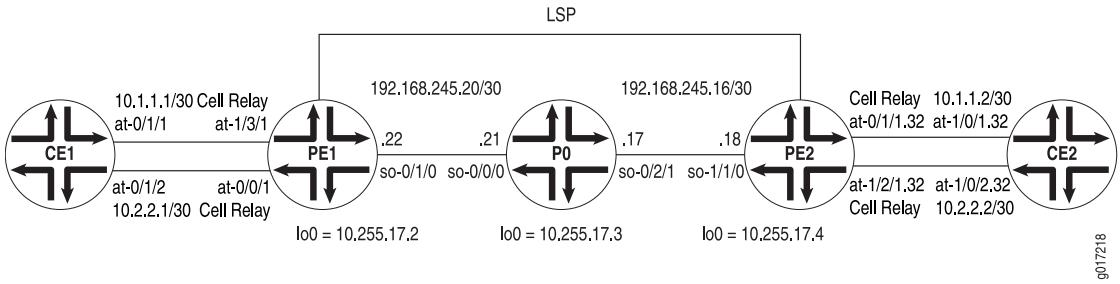


Figure 29 on page 334 shows that APS is configured on a PE router to protect a PE-CE link in a Layer 2 circuit. This example shows only the PE router configuration and assumes that you have preconfigured a full Layer 2 circuit topology. For more information about configuring Layer 2 circuits, see *JUNOS Layer 2 Circuits Feature Guide*.

On Router PE1, configure ATM2 IQ interface at-0/0/1 as an APS protect circuit and ATM2 IQ interface at-1/3/1 as a working circuit. Also, configure the working circuit interface as the primary interface for your Layer 2 circuit and configure the protect circuit interface as the protected interface for your Layer 2 circuit.

Router PE1

```
[edit]
chassis {
   fpc 0 {
      pic 0 {
         atm-l2circuit-mode {
            cell; # This dedicates FPC 0 PIC 0 to cell-relay mode.
         }
      }
   }
   fpc 1 {
      pic 3 {
         atm-l2circuit-mode {
            cell; # This dedicates FPC 1 PIC 3 to cell-relay mode.
         }
      }
   }
}
interfaces {
   at-0/0/1 {
      description "To CE1 at-0/1/2";
      encapsulation atm-ccc-cell-relay;
      sonet-options {
         aps {
            protect-circuit TEST; # This interface is the APS protect circuit.
         }
      }
```

```
        atm-options {
            pic-type atm2;
            promiscuous-mode;
        }
        unit 0 {
            allow-any-vci;
        }
    }
    at-1/3/1 {
        description "To CE1 at-0/1/1";
        encapsulation atm-ccc-cell-relay;
        sonet-options {
            aps {
                working-circuit TEST; # This interface is the APS working circuit.
            }
        }
        atm-options {
            pic-type atm2;
            promiscuous-mode;
        }
        unit 0 {
            allow-any-vci;
        }
    }
}
protocols {
    l2circuit {
        neighbor 10.255.17.4 {
            interface at-1/3/1.0 { # The Layer 2 circuit interface is the working circuit.
                protect-interface at-0/0/1.0; # The protect-interface is the protect circuit.
                virtual-circuit-id 100;
            }
        }
    }
}
```

Verifying Your Work

To verify proper operation of APS for Layer 2 circuits, use the following command:

```
show l2circuit connections
```

After you configure the Layer 2 circuit and the APS working and protect circuits, you can see which APS circuit is active for the Layer 2 circuit with the show l2circuit connections command. The first local interface that is displayed is always the active circuit. If the second local interface field indicates Protect-Inactive, the working circuit is active, as shown in this output sample.

```
user@PE0> show l2circuit connections
Layer-2 Circuit Connections:

Legend for connection status (St)
EI -- encapsulation invalid      NP -- interface h/w not present
MM -- mtu mismatch               Dn -- down
```

```
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
OL -- no outgoing label          XX -- unknown
NC -- intf encaps not CCC/TCC
CB -- rcvd cell-bundle size bad

Legend for interface status
Up -- operational
Dn -- down

Neighbor: 10.255.17.4
Interface                 Type  St     Time last up          # Up trans
 at-1/3/1.0(vc 100)        rmt   Up     Sep  3 17:48:25 2003           1
   Local interface: at-1/3/1.0, Status: Up, Encapsulation: ATM CELL (PORT Mode)
   Remote PE: 10.255.17.4, Negotiated control-word: Yes (Null)
   Incoming label: 100368, Outgoing label: 100112
   Local interface: at-0/0/1.0, Status: Dn, Encapsulation: ATM CELL (PORT Mode),

   Protect-Inactive
```

Conversely, if the first local interface is marked with the Protect-Active indicator, and the second local interface indicates Primary-Inactive, the protect circuit is active, as shown here:

```
user@PE0> show l2circuit connections
Layer-2 Circuit Connections:

Legend for connection status (St)
EI -- encapsulation invalid      NP -- interface h/w not present
MM -- mtu mismatch               Dn -- down
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
OL -- no outgoing label          XX -- unknown
NC -- intf encaps not CCC/TCC
CB -- rcvd cell-bundle size bad

Legend for interface status
Up -- operational
Dn -- down

Neighbor: 10.255.17.4
 Interface                 Type  St     Time last up          # Up trans
 at-1/3/1.0(vc 100)        rmt   Up     Sep  3 17:51:06 2003           2
   Local interface: at-0/0/1.0, Status: Up,  Encapsulation: ATM CELL (PORT Mode),

    Protect-Active
   Remote PE: 10.255.17.4, Negotiated control-word: No
   Incoming label: 100368, Outgoing label: 100112
   Local interface: at-1/3/1.0, Status: Dn,  Encapsulation: ATM CELL (PORT Mode),

    Primary-Inactive
```

For More Information

For additional information about Layer 2 circuits, see the following:

- *JUNOS Network Interfaces Configuration Guide*
- *JUNOS Class of Service Configuration Guide*

- *JUNOS VPNs Configuration Guide*
- RFC 3036, *LDP Specification*
- RFC 4447, *Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)* (except section 5.3)
- RFC 4448, *Encapsulation Methods for Transport of Ethernet over MPLS Networks*
- Internet draft draft-martini-atm-encap-mpls-01.txt, *Encapsulation Methods for Transport of ATM Cells/Frame Over IP and MPLS Networks* (expires December 2002)
- Internet draft draft-martini-l2circuit-encap-mpls-11.txt, *Encapsulation Methods for Transport of Layer 2 Frames over MPLS Networks* (expires August 2006)
- Internet draft draft-martini-l2circuit-trans-mpls-19.txt, *Transport of Layer 2 Frames Over MPLS* (expires September 2006)
- Internet draft draft-kompella-ppvpn-l2vpn-03.txt, *Layer 2 VPNs Over Tunnels* (expires October 2003)

Revision History

5 October 2009—10.0R1 Release. Roy Spencer.

13 June 2009—9.6R1 Release. Roy Spencer.

13 April 2009—9.5R1 Release. Roy Spencer.

15 January 2009—9.4R1 Release. Roy Spencer.

10 October 2008—9.3R1 Release. Roy Spencer.

10 July 2008—9.2R1 Release. Roy Spencer.

10 April 2008—9.1R1 Release. Roy Spencer.

1 February 2008—9.0R1 Release. Fawn Damitio.

29 June 2007—8.4R1 Release. Fawn Damitio.

27 March 2007—8.3R1 Release. Fawn Damitio.

12 January 2007—Added support for M120 router and MX960 routers. 8.2R1 Release. Fawn Damitio.

15 September 2006—Updated the references to the Layer 2 Circuits IETF drafts and RFCs, 8.1R1 Release. Richard Hendricks.

29 June 2006—8.0R1 Release. Richard Hendricks.

27 March 2006—7.6R1 Release. Richard Hendricks.

9 January 2006—7.5R1 Release. Richard Hendricks.

14 September 2005—7.4R1 Release. Richard Hendricks.

13 June 2005—Added support for configuring Layer 2 circuits simultaneously over RSVP and LDP LSPs, 7.3R1 Release. Richard Hendricks.

5 April 2005—7.2R1 Release. Richard Hendricks.

2 February 2005—Added local interface switching for Layer 2 circuits, 7.1R1 Release. Richard Hendricks.

6 October 2004—Added support for specifying a unique MTU for each Layer 2 circuit, 7.0R1 Release. Richard Hendricks.

6 July 2004—Added ATM2 IQ interface-based CoS and additional trunk-related statements, 6.4R1 Release. Richard Hendricks.

5 April 2004—Added ping mpls l2circuit commands, 6.3R1 Release. Richard Hendricks.

22 December 2003—Added Layer 2 circuit trunk mode and bandwidth reservation for Layer 2 circuits, 6.2R1 Release. Richard Hendricks.

22 September 2003—Added APS for Layer 2 circuits information and updated the traffic engineering example, 6.1R1 Release. Richard Hendricks.

30 June 2003—Added traffic engineering and control word mapping for Frame Relay and ATM2, 6.0R1 Release. Walter Goralski.

2 April 2003—Added PPP, HDLC, ATM Cell Mode, and AAL5 information, 5.7R1 Release. Richard Hendricks.

10 July 2002—Reformatted document. Richard Hendricks.

21 November 2001—Initial document written. Bill Nowak.

Chapter 11

Multicast over Layer 3 VPNs

This chapter covers the following topics:

- Multicast over Layer 3 VPNs Overview on page 340
- Multiprotocol BGP-Based Multicast VPNs: Next-Generation on page 341
- Dual PIM Multicast VPNs: Draft Rosen on page 341
- System Requirements for Multiprotocol BGP-Based Multicast VPNs: Next-Generation on page 342
- System Requirements for Dual PIM Multicast VPNs: Draft Rosen on page 343
- Multicast over Layer 3 VPNs Terms and Acronyms on page 343
- Building Blocks for Configuring Multiprotocol BGP-Based Multicast VPNs: Next-Generation on page 344
- Creating a Unique Logical Loopback Interface for the Routing Instance for MBGP MVPNs on page 345
- Configuring Interfaces for Layer 3 VPNs on page 345
- Configuring BGP, MPLS, RSVP, and an IGP on the PE and Core Routers for MBGP MVPNs on page 345
- Creating a Routing Instance for a Multiprotocol BGP-Based Multicast VPN on page 346
- Option: Configuring Sender and Receiver Sites on page 346
- Option: Specifying Route Targets on page 347
- Configuring Provider Tunnels on page 349
- Enabling Multicast VPN in BGP on page 349
- Configuring Intra-AS Inclusive Point-to-Multipoint Traffic Engineering LSPs on page 350
- Configuring Intra-AS Selective Provider Tunnels on page 352
- Configuring the Master PIM Instance on the PE Router for BGP-Based Multicast VPNs on page 354
- Configuring the Router's IPv4 Bootstrap Router Priority on page 355
- Configuring MBGP MVPNs to Support IPv6 Multicast Traffic on page 355
- Example: Configuring MBGP MVPN Senders and Receivers using PIM SM and RSVP-TE Provider Tunnels on page 358
- Example: Configuring MBGP Multicast VPNs on page 365

- Example: Configuring MBGP Multicast VPN Topology Variations on page 385
- MBGP Multicast VPN Extranets Overview on page 397
- MBGP Multicast VPN Extranets Configuration Guidelines on page 398
- Example: Configuring MBGP Multicast VPN Extranets on page 398
- Dual PIM Draft-Rosen Multicast VPN Operation on page 440
- Configuring Draft-Rosen Multicast VPNs on page 442
- Configuring BGP, MPLS, RSVP, and an IGP on the PE and Core Routers for Draft Rosen VPNs on page 442
- Creating a Unique Logical Loopback Interface for the Routing Instance for Draft Rosen VPNs on page 443
- Configuring the Master PIM Instance on the PE Router in the Service Provider Network on page 443
- Configuring PIM and the VPN Group Address in a Routing Instance on page 444
- Option: Configuring PIM Sparse Mode Graceful Restart for a Layer 3 VPN on page 445
- Option: Configuring Multicast Distribution Trees for Data on page 445
- Option: Configuring MSDP Within a Layer 3 VPN on page 446
- Draft-Rosen Multicast VPNs Examples on page 447
- Example: Basic IPv4 Multicast over a Layer 3 VPN Configuration on page 448
- Example: IPv4 Multicast with Interprovider VPNs Configuration on page 461
- For More Information on page 471
- Revision History on page 471

Multicast over Layer 3 VPNs Overview

The JUNOS Software provides three ways to configure IP version 4 (IPv4) multicast over Layer 3 virtual private networks (VPNs):

- Multiprotocol BGP-based multicast VPNs: next-generation, defined by a set of sender sites and a set of receiver sites and use BGP as the signaling protocol. We recommend using this method to configure multicast on Layer 3 VPNs because it is a simpler implementation than draft-rosen multicast VPNs.
- Draft-rosen multicast VPNs with service provider tunnels operating in any-source multicast (ASM) mode—Described in RFC 4364, *BGP/MPLS IP Virtual Private Networks (VPNs)* and based on Section Two of the IETF Internet draft draft-rosen-vpn-mcast-06.txt, *Multicast in MPLS/BGP VPNs*. This information is provided to you in case you already have dual PIM multicast VPNs configured on your network.
- Draft-rosen multicast VPNs with service provider tunnels operating in source-specific multicast (SSM) mode—Described in RFC 4364, *BGP/MPLS IP Virtual Private Networks (VPNs)* and based on the IETF Internet draft draft-rosen-vpn-mcast-07.txt, *Multicast in MPLS/BGP IP VPNs*. For more information about these types of draft-rosen multicast VPNs, see the *JUNOS Multicast Protocols Configuration Guide*.

The reader should be familiar with Layer 3 VPN operation on Juniper Networks routers, as well as standard PIM configurations. For more information on Protocol Independent Multicast (PIM) and Multicast Source Discovery Protocol (MSDP) and their usage in a Layer 3 VPN, see the *JUNOS Multicast Protocols Configuration Guide*. For more information on Layer 3 VPN configuration, see the *JUNOS VPNs Configuration Guide*. Both manuals are located at http://www.juniper.net/techpubs/software/index.html.

Multiprotocol BGP-Based Multicast VPNs: Next-Generation

Multiprotocol BGP-based multicast VPNs (also referred to as next-generation Layer 3 VPN multicast) constitute the next evolution after dual multicast VPNs (draft-rosen) and provide a simpler solution for administrators who want to configure multicast over Layer 3 VPNs.

The main characteristics of multiprotocol BGP-based multicast VPNs are:

- They extend Layer 3 VPN service (RFC 2547) to support IP multicast for Layer 3 VPN service providers
- They follow the same architecture as specified by RFC 2547 for unicast VPNs. Specifically, BGP is used as the control plane.
- They eliminate the requirement for the virtual router (VR) model, which is specified in Internet draft draft-rosen-vpn-mcast, *Multicast in MPLS/BGP VPNs*, for multicast VPNs.
- They rely on RFC-based unicast with extensions for intra-AS and inter-AS communication.

Multiprotocol BGP-based VPNs are defined by two sets of sites: a sender set and a receiver set. Hosts within a receiver site set can receive multicast traffic and hosts within a sender site set can send multicast traffic. A site set can be both receiver and sender, which means that hosts within such a site can both send and receive multicast traffic. Multiprotocol BGP-based VPNS can span organizations (so the sites can be intranets or extranets), can span service providers, and can overlap.

Site administrators configure multiprotocol BGP-based VPNs based on customer requirements and the existing BGP and MPLS VPN infrastructure. For more detailed information about multiprotocol BGP-based VPN configuration statements, see the *JUNOS VPNs Configuration Guide*.

Dual PIM Multicast VPNs: Draft Rosen

JUNOS Software supports Layer 3 VPNs based on the Internet draft draft-rosen-rfc2547bis, *BGP/MPLS VPNs*. This Internet draft defines a mechanism by which service providers can use their IP backbones to provide Layer 3 VPN services to their customers. The sites that make up a Layer 3 VPN are connected over a provider's existing public Internet backbone.

VPNs based on draft-rosen-rfc2547bis are also known as BGP/MPLS VPNs because BGP is used to distribute VPN routing information across the provider's backbone, and MPLS is used to forward VPN traffic across the backbone to remote VPN sites.

Customer networks, because they are private, can use either public addresses or private addresses, as defined in RFC 1918, *Address Allocation for Private Internets*. When customer networks that use private addresses connect to the public Internet infrastructure, the private addresses might overlap with the private addresses used by other network users. BGP/MPLS VPNs solve this problem by prefixing a VPN identifier to each address from a particular VPN site, thereby creating an address that is unique both within the VPN and within the public Internet. In addition, each VPN has its own VPN-specific routing table that contains the routing information for that VPN only.

In a unicast environment for Layer 3 VPNs, all VPN states are contained within the provider edge (PE) routers. With multicast over Layer 3 VPNs, two PIM adjacencies are established: one between the customer edge (CE) and PE routers through a VPN routing and forwarding (VRF) routing instance, the second between the main PE routers and their service provider core neighbors.

The set of master PIM adjacencies throughout the service provider's network makes up the forwarding path, and eventually forms a rendezvous point (RP) multicast distribution tree. The tree is rooted at the RP contained within the service provider's network. Because of this, core provider transit routers within the service provider's network must maintain multicast state information for the VPNs.

For multicast in Layer 3 VPNs to work correctly, there must be two types of rendezvous points. The VPN customer rendezvous point (VPN C-RP) is an RP that resides within a VPN that connects the segments of a customer network. The service provider rendezvous point (SP-RP) resides within the service provider network itself. Because a PE router connects to both the customer network and the service provider network, a PE router can act as an SP-RP, a VPN C-RP, or both.

NOTE: If you configure auto-RP or bootstrap router (BSR) on a PE router, the PE router cannot act as a VPN C-RP in a routing instance, but can learn about another router acting as the VPN C-RP.

System Requirements for Multiprotocol BGP-Based Multicast VPNs: Next-Generation

To implement multiprotocol BGP-based multicast VPNs, your system must meet these minimum requirements:

- JUNOS Release 9.2 or later for PIM dense mode, bootstrap router (BSR), auto-RP, and configuration of a PE router as the VPN C-RP. (Configuration of a PE router as the VPN C-RP is not a requirement in JUNOS Release 10.0 and later if you configure RPT-SPT mode. See the *JUNOS VPNs Configuration Guide*.)
- JUNOS Release 8.5 or later for point-to-multipoint traffic engineering provider tunnels (next-generation multicast VPN only).
- JUNOS Release 8.4 or later.
- Any hardware needed in your network to enable your Juniper Networks routers to act as PE routers.
- On M Series and T Series routers, a Tunnel Services PIC for any provider core router acting as an SP-RP.

- On M Series and T Series routers, a Tunnel Services PIC for any PE router where GRE tunneling is needed.
- On M Series and T Series routers, a Tunnel Services PIC for any CE or PE router acting as a DR or VPN C-RP.
- On M Series and T Series routers, a Tunnel Services PIC is required for GRE tunneling, as specified in Section Two of the IETF Internet draft *Multicast in MPLS/BGP VPNs*.

System Requirements for Dual PIM Multicast VPNs: Draft Rosen

- JUNOS Release 8.2 or later for support on MX Series routing platforms.
- JUNOS Release 7.2 or later for MSDP in a Layer 3 VPN.
- JUNOS Release 7.1 or later for multicast distribution trees for data.
- JUNOS Release 6.4 or later for PIM sparse mode graceful restart and configuring a PE router as the VPN C-RP.
- JUNOS Release 5.5 or later for PIM dense mode and logical loopback interfaces.
- JUNOS Release 5.3 or later for PIM sparse mode.
- Any hardware needed in your network to enable your Juniper Networks routers to act as PE routers.
- On M Series and T Series routers, a Tunnel Services PIC for any provider core router acting as an SP-RP.
- On M Series and T Series routers, a Tunnel Services PIC for any PE router where GRE tunneling is needed.
- On M Series and T Series routers, a Tunnel Services PIC for any CE or PE router acting as a DR or VPN C-RP.
- On M Series and T Series routers, a Tunnel Services PIC for GRE tunneling, as specified in Section Two of the IETF Internet draft *Multicast in MPLS/BGP VPNs*.
- For point-to-multipoint traffic engineering, a Tunnel Services PIC or vrf-tabel-label configuration.

Multicast over Layer 3 VPNs Terms and Acronyms

I

inclusive tree In multiprotocol BGP multicast VPNs, a single multicast distribution tree in the backbone that carries all the multicast traffic from a specified set of one or more multicast VPNs. An inclusive tree that carries the traffic of more than one multicast VPNs is an *aggregate inclusive tree*. An inclusive tree contains, as its members, all the PEs that attach to receiver sites of any of the multicast VPNs using the tree.

M

master PIM instance The global instance of PIM that is configured at the [edit protocols pim] hierarchy level.

multicast domain The set of VPN routing and forwarding (VRF) instances associated with interfaces that can send multicast traffic to one another.

S

selective tree In multiprotocol BGP multicast VPNs, a single multicast distribution tree in the backbone that carries traffic that belongs to a specified set of one or more multicast groups from one or more multicast VPNs. Such a tree is referred to as an *aggregate selective tree* when the multicast groups belong to different multicast VPNs.

SP-RP The rendezvous point (RP) for the service provider (this RP is not contained within the VPN).

V

VPN C-RP The customer RP for the VPN (this RP is contained within the VPN).

Building Blocks for Configuring Multiprotocol BGP-Based Multicast VPNs: Next-Generation

Configuring multiprotocol BGP-based (MBGP) multicast VPNs (MVPNs) (also referred to as next-generation Layer 3 VPN multicast) involves a series of major steps. Each major step can have variations; for example, you can choose to configure different types of provider tunnels. Think of the major steps as building blocks that can be used together in various ways.

To implement multiprotocol BGP-based multicast VPNs, you must perform one or more of the following major steps:

- Creating a Unique Logical Loopback Interface for the Routing Instance for MBGP MVPNs on page 345
- Configuring Interfaces for Layer 3 VPNs on page 345
- Configuring BGP, MPLS, RSVP, and an IGP on the PE and Core Routers for MBGP MVPNs on page 345
- Creating a Routing Instance for a Multiprotocol BGP-Based Multicast VPN on page 346
- Option: Configuring Sender and Receiver Sites on page 346
- Option: Specifying Route Targets on page 347
- Configuring Provider Tunnels on page 349
- Enabling Multicast VPN in BGP on page 349
- Configuring Intra-AS Inclusive Point-to-Multipoint Traffic Engineering LSPs on page 350

- Configuring Intra-AS Selective Provider Tunnels on page 352
- Configuring the Master PIM Instance on the PE Router for BGP-Based Multicast VPNs on page 354
- Configuring the Router's IPv4 Bootstrap Router Priority on page 355

Creating a Unique Logical Loopback Interface for the Routing Instance for MBGP MVPNs

To facilitate the PIM protocol within a Layer 3 VPN, configure a unique loopback interface for the routing instance at the [edit interfaces lo0 unit] hierarchy level:

```
[edit interfaces]
lo0 {
    unit 1 {
        family inet {
            address  ip-address;
        }
    }
}
```

Configuring Interfaces for Layer 3 VPNs

Configure the Layer 3 VPN logical interfaces and specify the family as inet:

```
[edit interfaces]
interface-name {
    unit logical-unit-number {
        family inet;
    }
}
```

Configuring BGP, MPLS, RSVP, and an IGP on the PE and Core Routers for MBGP MVPNs

To send multicast traffic across a Layer 3 VPN, you must configure network protocols to handle *intradomain routing* (an interior gateway protocol [IGP], such as Open Shortest Path First [OSPF] or Intermediate System-to-Intermediate System [IS-IS]), *interdomain routing* (Border Gateway Protocol [BGP]), *label switching* (Multiprotocol Label Switching [MPLS]), and *path signaling* (Resource Reservation Protocol [RSVP]). For more information about these protocols and examples of how to configure these protocols to support a Layer 3 VPN, see the *JUNOS VPNs Configuration Guide*.

NOTE: In multicast Layer 3 VPNs, the multicast PE routers must use the primary loopback address (or router ID) for sessions with their internal BGP peers. If the PE routers use a route reflector with next-hop self-configured, Layer 3 multicast over VPN does not work because PIM cannot transmit upstream interface information for multicast sources behind remote PE routers into the network core. Multicast Layer 3 VPNs require the BGP next-hop address of the VPN route to match the BGP next-hop address of the loopback VRF instance address.

Creating a Routing Instance for a Multiprotocol BGP-Based Multicast VPN

By default a multiprotocol BGP-based multicast VPN routing instance attaches to both sender and receiver sites. You can also manually configure the instance to attach to only sender or only receiver sites.

To create a multicast VPN routing instance, include the mvpn statement at the [edit routing-instances *routing-instance name* protocols] hierarchy level:

```
[edit]
routing-instances {
   vpn-a {
      instance-type vrf;
      protocols {
         mvpn { # Enables BGP/MPLS multicast VPN configuration.
         }
      }
   }
}
```

NOTE: You cannot configure PIM within a nonforwarding instance. If you try to do so, the router displays a commit check error and does not complete the configuration commit process.

Option: Configuring Sender and Receiver Sites

By default, multiprotocol BGP-based VPNs are attached to both sender and receiver sites. To specify that a VPN be attached only to a sender site or only to a receiver site, include the receiver-site or sender-site statement at the [edit routing instances *routing-instance-name* protocols mvpn] hierarchy level:

```
[edit]
routing-instances {
   vpn-a {
      instance-type vrf;
      protocols {
         mvpn {
            receiver-site;
            sender-site;
         }
      }
```

```
    }
}
```

Option: Specifying Route Targets

Specifying route targets for sender and receiver sites is useful when there is a mix of sender only, receiver only, and sender and receiver sites. This is because a sender site's routing table is used for exporting routes from a sender site and importing routes from a receiver site. A receiver site's routing table is used for exporting routes from a receiver site and importing routes from a sender site. A sender and receiver site does both. The route targets configured under multicast VPNs apply only to multicast VPN AD routes of type 1, 2, 3, and 5.

A PE router with sites in a specific multicast VPN must determine whether a received automatic discovery route is from a sender site or receiver site based on the following:

- If the PE router is configured to be only in a sender site, route targets are imported only from receiver sites. Imported automatic discovery routes must be from a receiver site.
- If the PE router is configured to be only in a receiver site, route targets are imported only from sender sites. Imported automatic discovery routes must be from a sender site.
- If the PE router is configured to be both in sender and receiver sites, the following guidelines apply:
  - Along with an import route target, you can optionally configure whether the route target is from a receiver or a sender site.
  - If a configuration is not provided, an imported automatic discovery route is treated as belonging to both the sender site and the receiver site.

For more information on route targets, see the *JUNOS VPNs Configuration Guide*.

In the following example, the export target is used for multicast VPN autodiscovery routes. It overrides the default VRF export target if the **unicast** statement is not included. The unicast statement applies the VRF export target and multicast VPN export route target to multicast VPN autodiscovery routes. The **apply-groups** statement specifies the groups from which to inherit configuration data. The a**pply-groups-except** statement specifies the groups from which to not inherit configuration data. The **import-target** statement specifies the import target for multicast VPN autodiscovery routes. It overrides the default VRF import target if the **unicast** statement is not included.

To specify route targets, include the **route-target** statement at the [**edit routing-instances** *routing-instance-name* **protocols mvpn**] hierarchy level:

```
[edit]
routing-instances {
    vpn-a {
        protocols {
            mvpn {
                route-target {
                    export-target {
```

```
                target target-community;
                unicast; #
                apply-groups group-name;
                apply-groups-except group-name;
            }
            import-target { #
                target target-value { # Target community.
                    receiver target-value; # Target community used when importing receiver
                        # site routes.
                    sender target-value; # Target community used when importing sender
                        # site routers.
                }
                unicast {
                    receiver;
                    sender;
                }
                apply-groups group-name;
                apply-groups-except group-name;
            }
        }
    }
  }
}
```

Existing VRF import or VRF export policies for importing and exporting VPN routes might prevent import or export of multicast VPN routes if the policies reflect routes based on the protocol type. The workaround is to change the policy to not reflect routes based on the protocol type or to use additional multicast VPN-specific configuration.

If the VRF import policy does not import BGP routes, multicast VPN routes of type 1, 2, 3, or 5 imported by BGP are rejected. There are two workarounds:

- You can add a term to allow routes from the BGP protocol with **family inet-mvpn**:

```
term 2 {
    from {
        protocol bgp;
        family inet-mvpn;
        community vpn_blue;
    }
    then {
        accept;
    }
}
```

  The customer can configure an import target under **mvpn**. The multicast VPN route of type 1, 2, 3, or 5 matching the configured target are imported.

```
protocol mvpn {
    import-target {
        target target:2:2;
    }
}
```

- If the VRF export policy uses a policy qualifier of type protocol to reject routes, the multicast VPN routes of type 1, 2, 3, or 5 are not exported. This is because JUNOS Software does not support a policy qualifier for MVPN. The workaround is to configure an export target under MVPN without including the unicast statement:

```
protocol mvpn {
   export-target {
      target target:2:2;
      }
   }
}
```

Configuring Provider Tunnels

The source address for a PIM-SM provider tunnel is the loopback address of the loopback interface in inet.0.

To configure a provider tunnel, include the provider-tunnel statement at the [edit routing-instances *routing-instance-name*] hierarchy level:

```
[edit]
routing-instances {
   vpn-a {
      provider-tunnel {
         pim-asm {
            apply-groups group-name;
            apply-groups-except group-name;
            group-address address;
         }
      }
   }
}
```

Enabling Multicast VPN in BGP

You also must enable multicast VPN by including the inet-mvpn or inet6-mvpn statements at the [edit protocols bgp family] hierarchy level:

```
[edit]
protocols {
   bgp {
      family {
         inet-mvpn; # Enables IPv4 multicast VPN.
         inet6-mvpn; # Enables IPv6 multicast VPN.
      }
   }
}
```

Configuring Intra-AS Inclusive Point-to-Multipoint Traffic Engineering LSPs

Point-to-multipoint traffic engineering LSPs are supported as the data plane for intra-AS inclusive provider tunnels. A multicast VPN can be configured to use inclusive trees or selective trees or a combination of both. Aggregation is not supported for point-to-multipoint traffic engineering LSPs.

NOTE: Configure either LDP or regular MPLS LSPs between PE routers to ensure VPN unicast connectivity. Point-to-multipoint LSPs are used for multicast data forwarding only.

You must configure the following when configuring point-to-multipoint LSPs in provider tunnels:

- The BGP multicast VPN control plane, as described in "Creating a Routing Instance for a Multiprotocol BGP-Based Multicast VPN" on page 346.
- Point-to-multipoint traffic engineering as the provider tunnel technology on each PE configured for multicast VPN that belongs to the sender site.
- Either a VT interface or a vrf-table-label on the multicast VPN instance. For more information about configuring VT interfaces, see the *JUNOS VPNs Configuration Guide*.
- Point-to-multipoint traffic engineering support on each P router.

On each PE router, a point-to-multipoint traffic engineering LSP must be configured for every multicast VPN instance that belongs to a sender site set. This means that if there are four multicast VPN instances configured on a PE router and three of these multicast VPN instances belong to sender site sets, three point-to-multipoint traffic engineering LSPs must be configured on this PE router. The PE would be the root of the three point-to-multipoint traffic engineering LSPs, and the leaves of the LSPs would be determined either dynamically or through a static configuration.

If the multicast VPN instance is configured for dynamic leaf discovery, the leaves are automatically discovered through intra-AS autodiscovery routes. The point-to-multipoint LSPs can be signaled using default attributes or configured attributes. If you configure the multicast VPN instance to use default attributes, the LSPs cannot be signaled with bandwidth reservation and do not support CAC. Point-to-multipoint LSPs with configured attributes support both bandwidth reservation and CAC. In addition, they can be configured to support traffic engineering attributes such as fast-reroute.

If the multicast VPN instance is configured for static leaf discovery, you configure the leafs statically. Point-to-multipoint LSPs that are configured statically support all traffic engineering attributes.

To configure dynamic leaf discovery, include the label-switched-path-template statement at the [edit routing-instance *routing-instance-name* provider-tunnel rsvp-te] hierarchy level. Dynamic discovery can be configured by using default attributes with the

default-template statement at the [edit routing-instance *routing-instance-name* provider-tunnel rsvp-te label-switched-path-template] hierarchy level.

If you want to signal with bandwidth reservation, use CAC, or use other traffic engineering options such as link protection, configure a template for dynamic leaf discovery by including the label-switched-path-template *template-name* statement at the [edit protocols mpls] hierarchy level:

```
[edit]
protocols {
   mpls {
      label-switched-path mvpn-example-p2mp-temlate {
         template;
         p2mp;
         link-protection;
         optimize-timer 50;
         traceoptions {
            file mvpn-a-p2mp-lsp.log;
            flag all;
         }
      }
   }
}
```

You can apply the configured or default template by including the template name at the [edit routing-instance *routing-instance-name* provider-tunnel rsvp-te label-switched-path-template] hierarchy level. Be sure to either configure a VT interface or include the vrf-table-label statement in the routing instance.

```
[edit]
routing-instance {
   routing-instance configured-dynamic-example {
   instance-type vrf;
   interface ge-1/0/0.0;
   route-distinguisher 10.255.71.1:100;
   vrf-table-label;
      provider-tunnel {
         rsvp-te label switched-path-template mvpn-example-p2mp-temlate;
      }
   }
}
```

To configure static LSPs, include the label-switched-path *label-switched-path* statement at the [edit protocols mpls] hierarchy level:

```
[edit]
protocols {
   mpls {
      label-switched-path vpls-example-p2mp-s21_lsp_a {
         to 192.168.1.1
         p2mp example-static-lsp;
      }
      label-switched-path vpls-example-p2mp-s21_lsp_b {
         to 192.168.1.2;
         p2mp example-static-lsp;
```

```
        }
    }
}
```

To apply statically configured LSPs, include the static statement at the [edit routing-instance *routing-instance-name* provider-tunnel rsvp-te static-lsp *static-lsp-name*] hierarchy level:

```
[edit]
routing-instance example-static {
    provider-tunnel {
        rsvp-te {
            static-lsp example-static-lsp;
        }
    }
}
```

Configuring Intra-AS Selective Provider Tunnels

Point-to-multipoint traffic engineering LSPs are supported as the data plane for selective provider tunnels. A multicast VPN can be configured to use inclusive trees or selective trees or a combination of both. Aggregation is not supported for point-to-multipoint traffic engineering LSPs.

NOTE: Configure either LDP or regular MPLS LSPs between PE routers to ensure VPN unicast connectivity. Point-to-multipoint LSPs are used for multicast data forwarding only.

You must configure the following when configuring point-to-multipoint LSPs in provider tunnels:

- The BGP multicast VPN control plane, as described in "Creating a Routing Instance for a Multiprotocol BGP-Based Multicast VPN" on page 346.
- Point-to-multipoint traffic engineering as the provider tunnel technology on each PE configured for multicast VPN that belongs to the sender site.
- Either a VT interface or a vrf-table-label on the multicast VPN instance. For more information about configuring VT interfaces, see the *JUNOS VPNs Configuration Guide*.
- Point-to-multipoint traffic engineering support on each P router.

When selective trees are used, there must be a separate point-to-multipoint traffic engineering LSP for each multicast distribution tree in the backbone that carries traffic belonging to a specified set of one or more multicast groups, from one or more multicast VPNs. Multiple groups can be bound to the same selective point-to-multipoint LSP if the selective point-to-multipoint LSP leaves are statically configured. If the leaves are dynamically discovered, only one source or group can be bound to it.

Selective point-to-multipoint LSPs can be statically configured or triggered by a bandwidth threshold. If the threshold rate is configured, a S-PMSI autodiscovery route

is generated for a particular (C-S, C-G) if it falls in the range specified by (C-S prefix, C-G prefix) and its data rate exceeds the configured threshold rate.

Below is an example configuration for point-to-multipoint LSPs on a selective tunnel with statically configured leafs:

```
[edit]
routing-instances {
   selective-tunnel-example {
      instance-type vrf;
      route-distinguisher 10.255.71.2:100;
      protocols {
         vpls {
            tunnel-services { # This enables vt interfaces for this routing instance.
         }
      }
      provider-tunnel {
         rsvp-te {
            label-switched-path-template {
               mvpn_template;
            }
         }
         }
         selective {
            group 225.10.10.1/32 {
               source 192.2.1.2/32 {
                  rsvp-te {
                     static-lsp lsp1;
                  }
               }
            }
            group 226.10.10.1/32 {
               source 192.2.1.2/32 {
                  rsvp-te {
                     static-lsp lsp1;
                  }
               }
            }
         }
      }
   }
}
```

The following example shows an example with dynamic selective trees and the default template:

```
[edit]
routing-instances {
   dynamic-selective-tunnel-example {
      instance-type vrf;
      route-distinguisher 10.255.71.2:100;
      protocols {
         vpls {
            tunnel-services {
         }
      }
```

```
        provider-tunnel {
            rsvp-te {
                label-switched-path-template {
                    default-template;
                }
            }
            selective {
                group 225.10.10.1/32 {
                    source 192.2.1.2/32 {
                        rsvp-te {
                            label-switched-path-template default-template;
                        }
                    }
                }
                group 226.10.10.1/32 {
                    source 192.2.1.2/32 {
                        rsvp-te {
                            label-switched-path-template default-template;
                        }
                    }
                }
            }
        }
    }
}
```

Configuring the Master PIM Instance on the PE Router for BGP-Based Multicast VPNs

To configure the master PIM instance that communicates with other PIM neighbors, include the pim statement at the [edit protocols] hierarchy level. BGP-based multicast VPNs support sparse mode, dense mode, or sparse-dense mode. The first example shown enables PIM sparse mode.

```
[edit protocols]
pim {
    interface all {
        mode sparse;
        version 2;
    }
}
```

The next example shown enables PIM dense mode.

```
[edit protocols]
pim {
    interface all {
        mode dense;
    }
}
```

Configuring the Router's IPv4 Bootstrap Router Priority

By default, the router has a bootstrap priority of 0, which means the router can never be the bootstrap router. To modify this priority, include the bootstrap-priority statement. The router with the highest priority value is elected to be the bootstrap router.

```
[edit protocols]
pim {
    bootstrap-priority number;
}
```

Configuring MBGP MVPNs to Support IPv6 Multicast Traffic

Multiprotocol BGP-based (MBGP) multicast VPNs (MVPNs) (also referred to as next-generation Layer 3 VPN multicast) can transport IPv6 multicast customer traffic across an IPv4 core network using RSVP-TE tunnels. Transporting IPv6 multicast customer traffic across an IPv4 core network does not require that IPv6 support or multicast support be configured in the core network.

The following features are supported for IPv6 multicast transport:

- PIM sparse mode
  - Static rendezvous points
  - Bootstrap router protocol
  - Embedded RP
- PIM dense mode
- PIM source-specific multicast
- MLD on the provider edge (PE) router to host interface
- RSVP-TE, PIM-SSM, or PIM-ASM inclusive provider tunnels
- RSVP-TE selective provider tunnels
- BGP mesh topologies or route reflectors

The configuration required to support IPv6 multicast traffic across an MBGP MVPN is largely the same as the configuration to support IPv4 multicast traffic. For an example of configuring an MBGP MVPN to transport IPv4 multicast customer traffic, see "Example: Configuring MBGP Multicast VPNs" on page 365.

To transport IPv6 multicast customer traffic across a service provider IPv4 core network:

1. If RSVP-TE LSPs and the bootstrap router protocol are used, enable the IPv6 address family on all core-facing interfaces on all PE routers participating in the MVPN. It is not necessary to configure an IPv6 address.

   ```
   [edit protocols bgp family inet6-mvpn]
   interface so-0/1/0 {
       unit 0 {
   ```

```
        family inet6 {
        }
    }
}
```

2. Enable MBGP to carry multicast VPN NLRI for the IPv6 address family and enable VPN signaling on all PE routers participating in the MVPN.

```
[edit protocols]
bgp {
    group vsix_mcast {
        family inet6-mvpn {
            signaling;
        }
    }
}
```

3. Enable MBGP to carry Layer 3 VPN NLRI for the IPv6 address family for unicast routes on all PE routers participating in the MVPN.

```
[edit protocols]
bgp {
    group vsix_mcast {
        family inet6-vpn {
            unicast;
        }
    }
}
```

4. Allow IPv6 routes to be resolved over an MPLS network by converting all routes stored in the **inet.3** routing table to IPv4-compatible IPv6 addresses and then copying them into the **inet6.3** routing table on all PE routers participating in the MVPN.

```
[edit protocols]
mpls {
    ipv6-tunneling;
    interface so-0/1/0;
    interface fe-0/0/0;
}
```

   The **inet6.3** routing table is used to resolve next-hop addresses for both IPv6 and IPv6 VPN routes.

5. By default, the routers support MLD version 1 (MLDv1). If you want to use MLDv2 on directly connected customer hosts, configure MLD version 2 on the PE router to host interfaces.

```
[edit protocols]
mld {
    version 2;
}
```

6. Configure either a static rendezvous point (RP) or the bootstrap router (BSR) protocol using the following:

   - If you are using a static RP, configure the local RP IPv6 address on the router that is the rendezvous point.

     ```
     [edit protocols pim rp]
     local {
        family inet6 {
           address ::10.12.53.12;
        }
     }
     ```

   - If you are using a static RP, configure the RP IPv6 address on all PE routers in the MVPN that are not the rendezvous point.

     ```
     [edit protocols pim rp]
     static {
        address ::10.12.53.12;
     }
     ```

   - If you are using the BSR protocol for automatic RP discovery, configure a bootstrap router to support the IPv6 address family.

     ```
     [edit protocols pim rp]
     bootstrap {
        family inet6;
     }
     ```

7. (Optional) If you are using a route reflector and the route reflector does not have a full mesh of RSVP LSPs tunnels, it might be necessary to add a static route. The static route is used to ensure that no updates are dropped by BGP dbecause no next-hop attribute are present. Configure the static route in the inet6.3 routing table with the 0::0/0 prefix. Set the next-hop attribute to discard and give the route a metric of 65000. The high metric used in this example allows the router to use the other routes with a lower metric if they exist.

   ```
   [edit routing-options]
   rib inet6.3 {
      static {
         route 0::0/0 {
            discard;
            metric 65000;
         }
      }
   }
   ```

8. After you have configured the network to transport IPv6 multicast customer traffic across an IPv4 core network and allowed time for the routes to be discovered, use the show route table *instance_name*.mvpn-inet6.0 command to see the type 1 autodiscovery routes.

9. Use the show mvpn instance command to verify that tthe provider tunnels have been established.

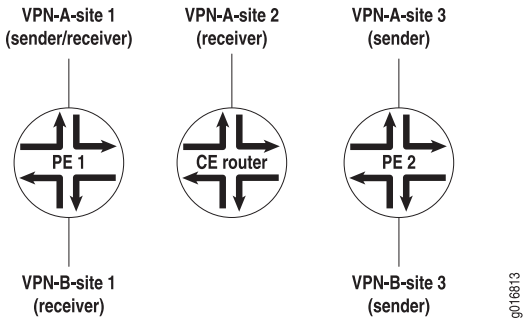
10. Use the show route table bgp.mvpn-inet6.0 command to verify that the correct BGP routes have been learned from MVPN.
11. After multicast traffic is flowing, use the show multicast route instance *instance_name* extensive inet6 command to verify that the multicast state, group address, source address, upstream interface list, and downstream interface list are correct.
12. Use the show route table *instance_name*.inet6.1 command to verify that the multicast forwarding table is correct.

Example: Configuring MBGP MVPN Senders and Receivers using PIM SM and RSVP-TE Provider Tunnels

This section contains a configuration example and commands you can issue to verify multiprotocol BGP multicast VPN sender and receiver site routers.

In the example shown in Figure 30 on page 358, there are three routers: PE1, the CE router, and PE2. Each router supports a specific role as a sender, receiver, or sender and receiver. This example does not represent a complete network.

Figure 30: Multiprotocol BGP Multicast VPN Senders and Receivers



Router PE1 is configured as a multicast sender and receiver site in VPN A, and as a multicast receiver site in VPN B. The relevant configuration for Router PE1 follows.

Router PE1

```
[edit]
routing-instances {
   vpn-a {
      instance-type vrf;
      interface so-6/0/0.0;
      interface so-6/0/1.0;
      provider tunnel {
         rsvp-te {
            label-switched-path-template default-template;
         }
      }
      protocols {
         mvpn {
            route-target {
               export-target unicast target:1:4;
               import-target unicast sender target target:1:4 receiver;
```

```
            }
          }
        }
        route-distinguisher 65535:0;
        vrf-target target:1:1;
        routing-options {
          auto-export;
          static {
            route 172.16.0.0/16 next-hop so-0/0/0.0;
            route 172.17.0.0/16 next-hop so-6/0/1.0;
          }
        }
      }
    }

    [edit]
    routing-instance {
      vpn-b {
        instance-type vrf;
        interface ge-0/3/0.0;
        provider-tunnel {
          pim-sm {
            group-address 224.1.1.2;
          }
        }
        protocols {
          mvpn {
            receiver-site;
            router-target {
              export target:1:5;
              import unicast;
            }
          }
        }
        route-distinguisher 65535:1;
        vrf-target target:1:2;
        routing-options {
          auto-export;
        }
      }
    }
```

Router PE2 is configured as a multicast sender-only site. The relevant configuration for Router PE2 follows.

Router PE2

```
[edit]
routing-instances {
  vpn-a {
    instance-type vrf;
    interface so-1/0/0.0;
    provider-tunnel {
      rsvp-te {
        label-switched-path-template default-template;
      }
    }
    protocols {
```

```
            mvpn {
               sender-site;
               route-target {
                  export target:1:4;
                  import unicast;
               }
            }
            route-distinguisher 65535:2;
            vrf-target target:1:1;
            routing-options {
               auto-export;
               static {
                  route 172.16.0.0/16 next-hop so-0/0/0.1;
                  route 172.17.0.0/16 next-hope so-6/0/1.0;
               }
            }
         }
      }
   }

   [edit]
   routing-instance {
      vpn-b {
         instance-type vrf;
         interface ge-0/3/0.0;
         provider-tunnel {
            pim-sm {
               group-address 224.1.1.2;
            }
         }
         protocols {
            mvpn {
               sender-site;
               router-target {
                  export target:1:5;
                  import unicast;
               }
            }
         }
         route-distinguisher 65535:1;
         vrf-target target:1:2;
         routing-options {
            auto-export;
         }
      }
   }
```

Verifying Your Work

To verify correct operation of multiprotocol BGP multicast VPNs, use the following commands:

- show mvpn c-multicast
- show mvpn instance

- show mvpn neighbor

The following sections show the output of these commands used with the configuration example:

- show mvpn c-multicast on page 361
- show mvpn instance on page 362
- show mvpn neighbor on page 364

show mvpn c-multicast

```
Router> show mvpn c-multicast
Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  C-mcast IPv4 (S:G)            Ptnl                          St
  192.168.195.78/32:225.5.5.5/32 PIM-SM:10.255.14.144, 239.1.1.1         RM
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-B
  C-mcast IPv4 (S:G)            Ptnl                          St
  192.168.195.94/32:226.6.6.6/32 PIM-SM:10.255.14.144, 239.2.0.0         RM

Router> show mvpn c-multicast extensive
Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  C-mcast IPv4 (S:G)            Ptnl                          St
  192.168.195.78/32:225.5.5.5/32 PIM-SM:10.255.14.144, 239.1.1.1         RM
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-B
  C-mcast IPv4 (S:G)            Ptnl                          St
  192.168.195.94/32:226.6.6.6/32 PIM-SM:10.255.14.144, 239.2.0.0         RM

Router> show mvpn c-multicast summary
Instance: VPN-A
  C-multicast IPv4 route count: 1
Instance: VPN-B
```

```
  C-multicast IPv4 route count: 2
```

show mvpn instance

```
Router> show mvpn instance
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  Provider tunnel: I-P-tnl:PIM-SM:10.255.14.144, 239.1.1.1
  Neighbor                              I-P-tnl
  10.255.14.160                         PIM-SM:10.255.14.160, 239.1.1.1
  10.255.70.17                          PIM-SM:10.255.70.17, 239.1.1.1
  C-mcast IPv4 (S:G)            Ptnl                              St
  192.168.195.78/32:225.5.5.5/32 PIM-SM:10.255.14.144, 239.1.1.1         RM
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-B
  Provider tunnel: I-P-tnl:PIM-SM:10.255.14.144, 239.2.0.0
  Neighbor                              I-P-tnl
  10.255.14.160                         PIM-SM:10.255.14.160, 239.2.0.0
  10.255.70.17                          PIM-SM:10.255.70.17, 239.2.0.0
  C-mcast IPv4 (S:G)            Ptnl                              St
  192.168.195.94/32:226.6.6.6/32 PIM-SM:10.255.14.144, 239.2.0.0         RM

Router> show mvpn instance extensive
Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  Provider tunnel: I-P-tnl:PIM-SM:10.255.14.144, 239.1.1.1
  Neighbor                              I-P-tnl
  10.255.14.160                         PIM-SM:10.255.14.160, 239.1.1.1
  10.255.70.17                          PIM-SM:10.255.70.17, 239.1.1.1
  C-mcast IPv4 (S:G)            Ptnl                              St
  192.168.195.78/32:225.5.5.5/32 PIM-SM:10.255.14.144, 239.1.1.1         RM
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-B
  Provider tunnel: I-P-tnl:PIM-SM:10.255.14.144, 239.2.0.0
  Neighbor                              I-P-tnl
  10.255.14.160                         PIM-SM:10.255.14.160, 239.2.0.0
```

```
  10.255.70.17                          PIM-SM:10.255.70.17, 239.2.0.0
  C-mcast IPv4 (S:G)            Ptnl                          St
  192.168.195.94/32:226.6.6.6/32 PIM-SM:10.255.14.144, 239.2.0.0         RM

Router> show mvpn instance
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  Provider tunnel: I-P-tnl:RSVP-TE P2MP:10.255.71.190, 27859,10.255.71.190
  Neighbor                              I-P-tnl
  10.255.71.2                          RSVP-TE P2MP:10.255.71.2, 39143,10.255.71.2

  C-mcast IPv4 (S:G)            Ptnl                          St
  192.1.1.2/32:225.10.10.1/32   RSVP-TE P2MP:10.255.71.2, 39143,10.255.71.2
    DS
  0.0.0.0/0:225.10.10.1/32
Router> show mvpn instance summary
Instance: VPN-A
  Neighbor count: 2
  C-multicast IPv4 route count: 1
Instance: VPN-B
  Neighbor count: 4
  C-multicast IPv4 route count: 2

Router> show mvpn instance VPN-A
Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  Provider tunnel: I-P-tnl:PIM-SM:10.255.14.144, 239.1.1.1
  Neighbor                              I-P-tnl
  10.255.14.160                         PIM-SM:10.255.14.160, 239.1.1.1
  10.255.70.17                          PIM-SM:10.255.70.17, 239.1.1.1
  C-mcast IPv4 (S:G)            Ptnl                          St
  192.168.195.78/32:225.5.5.5/32 PIM-SM:10.255.14.144, 239.1.1.1         RM

Router> show mvpn instance VPN-A extensive
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  Provider tunnel: I-P-tnl:PIM-SM:10.255.14.144, 239.1.1.1
  Neighbor                              I-P-tnl
  10.255.14.160                         PIM-SM:10.255.14.160, 239.1.1.1
  10.255.70.17                          PIM-SM:10.255.70.17, 239.1.1.1
  C-mcast IPv4 (S:G)            Ptnl                          St
  192.168.195.78/32:225.5.5.5/32 PIM-SM:10.255.14.144, 239.1.1.1         RM

Router> show mvpn instance VPN-A summary
Instance: VPN-A
  Neighbor count: 2
```

```
  C-multicast IPv4 route count: 1
```

show mvpn neighbor

```
Router> show mvpn neighbor
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  Neighbor                              I-P-tnl
  10.255.14.160                         PIM-SM:10.255.14.160, 239.1.1.1
  10.255.70.17                          PIM-SM:10.255.70.17, 239.1.1.1
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-B
  Neighbor                              I-P-tnl
  10.255.14.160                         PIM-SM:10.255.14.160, 239.2.0.0
  10.255.70.17                          PIM-SM:10.255.70.17, 239.2.0.0

Router> show mvpn neighbor extensive
Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  C-mcast IPv4 (S:G)            Ptnl                            St
  192.168.195.78/32:225.5.5.5/32 PIM-SM:10.255.14.144, 239.1.1.1         RM
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-B
  C-mcast IPv4 (S:G)            Ptnl                            St
  192.168.195.94/32:226.6.6.6/32 PIM-SM:10.255.14.144, 239.2.0.0         RM
```

The following is output for a p2mp configuration.

```
Router> show mvpn neighbor
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
```

```
Instance: VPN-A
  Neighbor                                I-P-tnl
  10.255.71.2                          RSVP-TE P2MP:10.255.71.2, 39143,10.255.71.2
```

Example: Configuring MBGP Multicast VPNs

This example provides a step-by-step procedure to configure multicast services across a multiprotocol BGP (MBGP) Layer 3 virtual private network.

- Before You Begin on page 365
- Overview and Topology on page 365
- Configuration on page 366

Before You Begin

Depending on the devices you are using, you might be required to configure static routes to:

- The multicast sender
- The Fast Ethernet interface to which the sender is connected on the multicast receiver
- The multicast receiver
- The Fast Ethernet interface to which the receiver is connected on the multicast sender

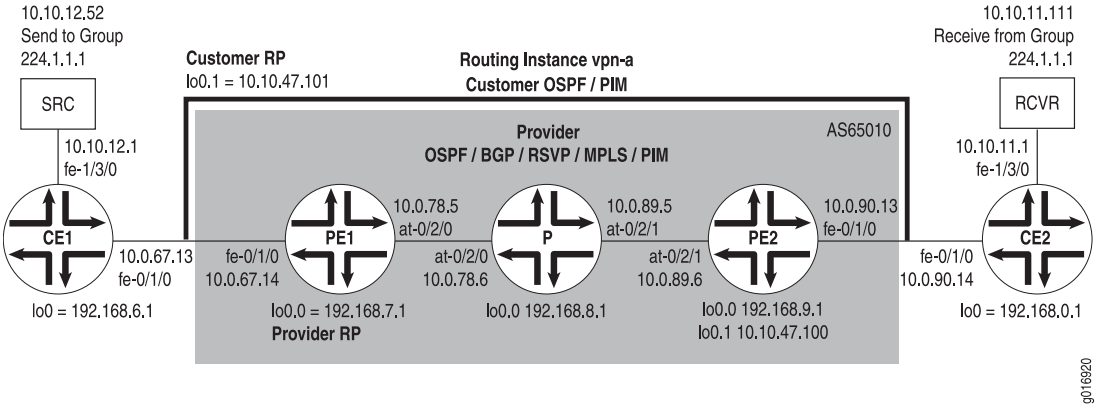
Overview and Topology

This example configures the following technologies:

- IPv4
- BGP
- OSPF
- RSVP
- MPLS
- PIM sparse mode
- Static RP

The topology of the network is shown in Figure 31 on page 366.

Figure 31: Multicast Over Layer 3 VPN Example Topology



Configuration

NOTE: In any configuration session it is a good practice to periodically verify that the configuration can be committed using the commit check command.

In this example, the router being configured is identified using the following command prompts:

- ce1 identifies the customer edge 1 (CE1) router
- pe1 identifies the provider edge 1 (PE1) router
- p identifies the provider core (P) router
- ce2 identifies the customer edge 2 (CE2) router
- pe2 identifies the provider edge 2 (PE2) router

To configure MBGP multicast VPNs for the network shown in Figure 31 on page 366, perform the following steps:

- Configuring Interfaces on page 367
- Configuring OSPF on page 368
- Configuring BGP on page 369
- Configuring RSVP on page 370
- Configuring MPLS on page 371
- Configuring the VRF Routing Instance on page 372
- Configuring PIM on page 373
- Configuring the Provider Tunnel on page 374
- Configuring the Rendezvous Point on page 375

Configuring Interfaces

Step-by-Step Procedure

1. On each router, configure an IP address on the loopback logical interface 0 (lo0.0).

```
user@ce1# set interfaces lo0 unit 0 family inet address 192.168.6.1/32 primary

user@pe1# set interfaces lo0 unit 0 family inet address 192.168.7.1/32
    primary

user@p# set interfaces lo0 unit 0 family inet address 192.168.8.1/32 primary

user@pe2# set interfaces lo0 unit 0 family inet address 192.168.9.1/32
    primary

user@ce2# set interfaces lo0 unit 0 family inet address 192.168.0.1/32 primary
```

   Use the show interfaces terse command to verify that the IP address is correct on the loopback logical interface.

2. On the PE and CE routers, configure the IP address and protocol family on the Fast Ethernet interfaces. Specify the inet protocol family type.

```
user@ce1# set interfaces fe-1/3/0 unit 0 family inet address 10.10.12.1/24
user@ce1# set interfaces fe-0/1/0 unit 0 family inet address 10.0.67.13/30

user@pe1# set interfaces fe-0/1/0 unit 0 family inet address 10.0.67.14/30

user@pe2# set interfaces fe-0/1/0 unit 0 family inet address 10.0.90.13/30

user@ce2# set interfaces fe-0/1/0 unit 0 family inet address 10.0.90.14/30
user@ce2# set interfaces fe-1/3/0 unit 0 family inet address 10.10.11.1/24
```

   Use the show interfaces terse command to verify that the IP address is correct on the Fast Ethernet interfaces.

3. On the PE and P routers, configure the ATM interfaces' VPI and maximum VCs. If the default PIC type is different on directly connected ATM interfaces, configure the PIC type to be the same. Configure the logical interface VCI, protocol family, local IP address, and destination IP address.

```
user@pe1# set interfaces at-0/2/0 atm-options pic-type atm1
user@pe1# set interfaces at-0/2/0 atm-options vpi 0 maximum-vcs 256
user@pe1# set interfaces at-0/2/0 unit 0 vci 0.128
user@pe1# set interfaces at-0/2/0 unit 0 family inet address 10.0.78.5/32
    destination 10.0.78.6

user@p# set interfaces at-0/2/0 atm-options pic-type atm1
user@p# set interfaces at-0/2/0 atm-options vpi 0 maximum-vcs 256
user@p# set interfaces at-0/2/0 unit 0 vci 0.128
```

```
user@p# set interfaces at-0/2/0 unit 0 family inet address 10.0.78.6/32
  destination 10.0.78.5
user@p# set interfaces at-0/2/1 atm-options pic-type atm1
user@p# set interfaces at-0/2/1 atm-options vpi 0 maximum-vcs 256
user@p# set interfaces at-0/2/1 unit 0 vci 0.128
user@p# set interfaces at-0/2/1 unit 0 family inet address 10.0.89.5/32
  destination 10.0.89.6

user@pe2# set interfaces at-0/2/1 atm-options pic-type atm1
user@pe2# set interfaces at-0/2/1 atm-options vpi 0 maximum-vcs 256
user@pe2# set interfaces at-0/2/1 unit 0 vci 0.128
user@pe2# set interfaces at-0/2/1 unit 0 family inet address 10.0.89.6/32
  destination 10.0.89.5
```

Use the show configuration interfaces command to verify that the ATM interfaces' VPI and maximum VCs are correct and that the logical interface VCI, protocol family, local IP address, and destination IP address are correct.

Configuring OSPF

Step-by-Step Procedure

1. On the P and PE routers, configure the provider instance of Open Shortest Path First (OSPF). Specify the lo0.0 and ATM core-facing logical interfaces. The provider instance of OSPF on the PE router forms adjacencies with the OSPF neighbors on the other PE router and the P router.

```
user@pe1# set protocols ospf area 0.0.0.0 interface at-0/2/0.0
user@pe1# set protocols ospf area 0.0.0.0 interface lo0.0

user@p# set protocols ospf area 0.0.0.0 interface lo0.0
user@p# set protocols ospf area 0.0.0.0 interface all
user@p# set protocols ospf area 0.0.0.0 interface fxp0 disable

user@pe2# set protocols ospf area 0.0.0.0 interface lo0.0
user@pe2# set protocols ospf area 0.0.0.0 interface at-0/2/1.0
```

Use the show ospf interfaces command to verify that the lo0.0 and ATM core-facing logical interfaces are configured for OSPF.

2. On the CE routers, configure the customer instance of Open Shortest Path First (OSPF). Specify the loopback and Fast Ethernet logical interfaces . The customer instance of OSPF on the CE routers form adjacencies with the neighbors within the VPN routing instance of OSPF on the PE routers.

```
user@ce1# set protocols ospf area 0.0.0.0 interface fe-0/1/0.0
user@ce1# set protocols ospf area 0.0.0.0 interface fe-1/3/0.0
user@ce1# set protocols ospf area 0.0.0.0 interface lo0.0

user@ce2# set protocols ospf area 0.0.0.0 interface fe-0/1/0.0
user@ce2# set protocols ospf area 0.0.0.0 interface fe-1/3/0.0
user@ce2# set protocols ospf area 0.0.0.0 interface lo0.0
```

Use the show ospf interfaces command to verify that the correct loopback and Fast Ethernet logical interfaces have been added to the OSPF protocol.

3. On the P and PE routers, configure OSPF traffic engineering support for the provider instance of OSPF.

   The shortcuts statement enables the master instance of OSPF to use a label-switched path as the next hop.

   ```
   user@pe1# set protocols ospf traffic-engineering shortcuts

   user@p# set protocols ospf traffic-engineering shortcuts

   user@pe2# set protocols ospf traffic-engineering shortcuts
   ```

   Use the show ospf overview or show configuration protocols ospf command to verify that traffic engineering support is enabled.

Configuring BGP

Step-by-Step Procedure

1. On the P router, configure BGP for the VPN. The local address is the local lo0.0 address. The neighbor addresses are the PE routers' lo0.0 addresses.

   The unicast statement enables the router to use BGP to advertise network layer reachability information (NLRI). The signaling statement enables the router to use BGP as the signaling protocol for the VPN.

   ```
   user@p# set protocols bgp group group-mvpn type internal
   user@p# set protocols bgp group group-mvpn local-address 192.168.8.1
   user@p# set protocols bgp group group-mvpn family inet unicast
   user@p# set protocols bgp group group-mvpn family inet-mvpn signaling
   user@p# set protocols bgp group group-mvpn neighbor 192.168.9.1
   user@p# set protocols bgp group group-mvpn neighbor 192.168.7.1
   ```

   Use the show configuration protocols bgp command to verify that the router has been configured to use BGP to advertise NLRI.

2. On the PE and P routers, configure the BGP local autonomous system number.

   ```
   user@pe1# set routing-options autonomous-system 0.65010

   user@p# set routing-options autonomous-system 0.65010

   user@pe2# set routing-options autonomous-system 0.65010
   ```

   Use the show configuration routing-options command to verify that the BGP local autonomous system number is correct.

3. On the PE routers, configure BGP for the VPN. Configure the local address as the local lo0.0 address. The neighbor addresses are the lo0.0 addresses of the P router and the other PE router.

```
user@pe1# set protocols bgp group group-mvpn type internal
user@pe1# set protocols bgp group group-mvpn local-address 192.168.7.1
user@pe1# set protocols bgp group group-mvpn family inet-vpn unicast
user@pe1# set protocols bgp group group-mvpn family inet-mvpn signaling
user@pe1# set protocols bgp group group-mvpn neighbor 192.168.9.1
user@pe1# set protocols bgp group group-mvpn neighbor 192.168.8.1

user@pe2# set protocols bgp group group-mvpn type internal
user@pe2# set protocols bgp group group-mvpn local-address 192.168.9.1
user@pe2# set protocols bgp group group-mvpn family inet-vpn unicast
user@pe2# set protocols bgp group group-mvpn family inet-mvpn signaling
user@pe2# set protocols bgp group group-mvpn neighbor 192.168.7.1
user@pe2# set protocols bgp group group-mvpn neighbor 192.168.8.1
```

Use the show bgp group command to verify that the BGP configuration is correct.

4. On the PE routers, configure a policy to export the BGP routes into OSPF.

```
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

user@pe2# set policy-options policy-statement bgp-to-ospf from protocol bgp
user@pe2# set policy-options policy-statement bgp-to-ospf then accept
```

Use the show policy bgp-to-ospf command to verify that the policy is correct.

Configuring RSVP

Step-by-Step Procedure

1. On the PE routers, enable RSVP on the interfaces that participate in the LSP. Configure the Fast Ethernet and ATM logical interfaces.

```
user@pe1# set protocols rsvp interface fe-0/1/0.0
user@pe1# set protocols rsvp interface at-0/2/0.0

user@pe2# set protocols rsvp interface fe-0/1/0.0
user@pe2# set protocols rsvp interface at-0/2/1.0
```

2. On the P router, enable RSVP on the interfaces that participate in the LSP. Configure the ATM logical interfaces.

```
user@p# set protocols rsvp interface at-0/2/0.0
user@p# set protocols rsvp interface at-0/2/1.0
```

Use the show configuration protocols rsvp command to verify that the RSVP configuration is correct.

Configuring MPLS

Step-by-Step Procedure

1. On the PE routers, configure an MPLS LSP to the PE router that is the LSP egress point. Specify the IP address of the lo0.0 interface on the router at the other end of the LSP. Configure MPLS on the ATM, Fast Ethernet, and lo0.0 interfaces.

   To help identify each LSP when troubleshooting, configure a different LSP name on each PE router. In this example we use the name to-pe2 as the name for the LSP configured on PE1 and to-pe1 as the name for the LSP configured on PE2.

   ```
   user@pe1# set protocols mpls label-switched-path to-pe2 to 192.168.9.1
   user@pe1# set protocols mpls interface fe-0/1/0.0
   user@pe1# set protocols mpls interface at-0/2/0.0
   user@pe1# set protocols mpls interface lo0.0
   ```

   ```
   user@pe2# set protocols mpls label-switched-path to-pe1 to 192.168.7.1
   user@pe2# set protocols mpls interface fe-0/1/0.0
   user@pe2# set protocols mpls interface at-0/2/1.0
   user@pe2# set protocols mpls interface lo0.0
   ```

   Use the show configuration protocols mpls and show route label-switched-path to-pe1 commands to verify that the MPLS and LSP configuration is correct.

   After the configuration is committed, use the show mpls lsp name to-pe1 and show mpls lsp name to-pe2 commands to verify that the LSP is operational.

2. On the P router, enable MPLS. Specify the ATM interfaces connected to the PE routers.

   ```
   user@p# set protocols mpls interface at-0/2/0.0
   user@p# set protocols mpls interface at-0/2/1.0
   ```

   Use the show mpls interface command to verify that MPLS is enabled on the ATM interfaces.

3. On the PE and P routers, configure the protocol family on the ATM interfaces associated with the LSP. Specify the mpls protocol family type.

   ```
   user@pe1# set interfaces at-0/2/0 unit 0 family mpls
   ```

   ```
   user@p# set interfaces at-0/2/0 unit 0 family mpls
   user@p# set interfaces at-0/2/1 unit 0 family mpls
   ```

   ```
   user@pe2# set interfaces at-0/2/1 unit 0 family mpls
   ```

   Use the show mpls interface command to verify that the MPLS protocol family is enabled on the ATM interfaces associated with the LSP.

Configuring the VRF Routing Instance

Step-by-Step Procedure

1. On the PE routers, configure a routing instance for the VPN and specify the vrf instance type. Add the Fast Ethernet and lo0.1 customer-facing interfaces. Configure the VPN instance of OSPF and include the BGP-to-OSPF export policy.

```
user@pe1# set routing-instances vpn-a instance-type vrf
user@pe1# set routing-instances vpn-a interface lo0.1
user@pe1# set routing-instances vpn-a interface fe-0/1/0.0
user@pe1# set routing-instances vpn-a protocols ospf export bgp-to-ospf
user@pe1# set routing-instances vpn-a protocols ospf area 0.0.0.0 interface
  all
```

```
user@pe2# set routing-instances vpn-a instance-type vrf
user@pe2# set routing-instances vpn-a interface lo0.1
user@pe2# set routing-instances vpn-a interface fe-0/1/0.0
user@pe2# set routing-instances vpn-a protocols ospf export bgp-to-ospf
user@pe2# set routing-instances vpn-a protocols ospf area 0.0.0.0 interface
  all
```

   Use the show configuration routing-instances vpn-a command to verify that the routing instance configuration is correct.

2. On the PE routers, configure a route distinguisher for the routing instance. A route distinguisher allows the router to distinguish between two identical IP prefixes used as VPN routes. Configure a different route distinguisher on each PE router. This example uses 65010:1 on PE1 and 65010:2 on PE2.

```
user@pe1# set routing-instances vpn-a route-distinguisher 65010:1
```

```
user@pe2# set routing-instances vpn-a route-distinguisher 65010:2
```

   Use the show configuration routing-instances vpn-a command to verify that the route distinguisher is correct.

3. On the PE routers, configure default VRF import and export policies. Based on this configuration, BGP automatically generates local routes corresponding to the route target referenced in the VRF import policies. This example uses 2:1 as the route target.

NOTE: You must configure the same route target on each PE router for a given VPN routing instance.

```
user@pe1# set routing-instances vpn-a vrf-target target:2:1
```

```
user@pe2# set routing-instances vpn-a vrf-target target:2:1
```

   Use the show configuration routing-instances vpn-a command to verify that the route target is correct.

4. On the PE routers, configure the VPN routing instance for multicast support.

```
user@pe1# set routing-instances vpn-a protocols mvpn

user@pe2# set routing-instances vpn-a protocols mvpn
```

Use the show configuration routing-instance vpn-a command to verify that the VPN routing instance has been configured for multicast support.

5. On the PE routers, configure an IP address on the loopback logical interface 1 (lo0.1) used in the customer routing instance VPN.

```
user@pe1# set interfaces lo0 unit 1 family inet address 10.10.47.101/32

user@pe2# set interfaces lo0 unit 1 family inet address 10.10.47.100/32
```

Use the show interfaces terse command to verify that the IP address on the loopback logical interface 1 is correct.

Configuring PIM

Step-by-Step Procedure

1. On the PE and P routers, enable the provider instance of PIM. Add the core-facing ATM interfaces. On the PE routers, also configure the lo0.0 interface. Specify the mode as sparse and the version as 2.

```
user@pe1# set protocols pim interface at-0/2/0.0 mode sparse
user@pe1# set protocols pim interface at-0/2/0.0 version 2
user@pe1# set protocols pim interface lo0.0 mode sparse
user@pe1# set protocols pim interface lo0.0 version 2

user@p# set protocols pim interface at-0/2/0.0 mode sparse
user@p# set protocols pim interface at-0/2/0.0 version 2
user@p# set protocols pim interface at-0/2/1.0 mode sparse
user@p# set protocols pim interface at-0/2/1.0 version 2

user@pe2# set protocols pim interface at-0/2/1.0 mode sparse
user@pe2# set protocols pim interface at-0/2/1.0 version 2
user@pe2# set protocols pim interface lo0.0 mode sparse
user@pe2# set protocols pim interface lo0.0 version 2
```

Use the show pim interfaces command to verify that PIM sparse-mode is enabled on the core-facing ATM interfaces.

2. On the PE routers, enable the VPN customer instance of PIM. Configure the lo0.1 and the customer-facing Fast Ethernet interface. Specify the mode as sparse and the version as 2.

```
user@pe1# set routing-instances vpn-a protocols pim interface lo0.1 mode
  sparse
user@pe1# set routing-instances vpn-a protocols pim interface lo0.1 version
  2
user@pe1# set routing-instances vpn-a protocols pim interface fe-0/1/0.0
  mode sparse
```

```
user@pe1# set routing-instances vpn-a protocols pim interface fe-0/1/0.0
    version 2


user@pe2# set routing-instances vpn-a protocols pim interface lo0.1 mode
    sparse
user@pe2# set routing-instances vpn-a protocols pim interface lo0.1 version
    2
user@pe2# set routing-instances vpn-a protocols pim interface fe-0/1/0.0
    mode sparse
user@pe2# set routing-instances vpn-a protocols pim interface fe-0/1/0.0
    version 2
```

Use the show pim interfaces instance vpn-a command to verify that PIM sparse-mode is enabled on the lo0.1 interface and the customer-facing Fast Ethernet interface.

3. On the CE routers, enable the customer instance of PIM. In this example we configure all interfaces. Specify the mode as sparse and the version as 2.

```
user@ce1# set protocols pim interface all


user@ce2# set protocols pim interface all mode sparse
user@ce2# set protocols pim interface all version 2
```

Use the show pim interfaces command to verify that PIM sparse mode is enabled on all interfaces.

Configuring the Provider Tunnel

Step-by-Step Procedure

1. On the PE1 router, configure the provider tunnel. Specify the multicast address to be used.

   The provider-tunnel statement instructs the router to send multicast traffic across a tunnel. The pim-asm statement instructs the router to accept the multicast stream from any source.

   ```
   user@pe1# set routing-instances vpn-a provider-tunnel pim-asm group-address
       224.1.1.1
   ```

   Use the show configuration routing-instance vpn-a command to verify that the multicast group address is correct on Router PE1.

2. On the PE2 router, configure the provider tunnel. Specify the multicast address to be used.

   ```
   user@pe2# set routing-instances vpn-a provider-tunnel pim-asm group-address
       224.1.1.1
   ```

   Use the show configuration routing-instance vpn-a command to verify that the multicast group address is correct on Router PE2.

Configuring the Rendezvous Point

Step-by-Step Procedure

1. Configure the PE1 router to be the rendezvous point for the provider instance of PIM. Specify the lo0.0 address of the PE1 router.

   ```
   user@pe1# set protocols pim rp local address 192.168.7.1
   ```

   Use the show pim rps command to verify that to correct local IP address is configured for the provider instance RP.

2. Configure the static rendezvous point on the P router and PE2 router for the provider instance of PIM. Specify the lo0.0 address of the PE1 router. Specify the version as 2.

   ```
   user@p# set protocols pim rp static address 192.168.7.1 version 2

   user@pe2# set protocols pim rp static address 192.168.7.1 version 2
   ```

   Use the show pim rps command to verify that the correct static IP address is configured for the provider instance RP.

3. Configure the PE1 router to be the rendezvous point for the customer instance of PIM. Specify the lo0.1 address of the PE1 router. Specify the multicast address to be used.

   ```
   user@pe1# set routing-instances vpn-a protocols pim rp local address
     10.10.47.101
   user@pe1# set routing-instances vpn-a protocols pim rp local group-ranges
     224.1.1.1/32
   ```

   Use the show pim rps instance vpn-a command to verify that the correct local IP address is configured for the customer instance RP.

4. On the PE2 router, configure the static rendezvous point for the customer instance of PIM. Specify the lo0.1 address of the PE1 router.

   ```
   user@pe2# set routing-instances vpn-a protocols pim rp static address
     10.10.47.101
   ```

   Use the show pim rps instance vpn-a command to verify that the correct static IP address is configured for the customer instance RP.

5. On the CE routers, configure the static rendezvous point for the customer instance of PIM. Specify the lo0.1 address of the PE1 router.

   ```
   user@ce1# set protocols pim rp static address 10.10.47.101 version 2

   user@ce2# set protocols pim rp static address 10.10.47.101 version 2
   ```

   Use the show pim rps command to verify that the correct static IP address is configured for the customer instance RP.

6. Use the commit check command to verify that the configuration can be successfully committed. If the configuration passes the check, commit the configuration.
7. Start the multicast sender device connected to CE1.
8. Start the multicast receiver device connected to CE2.
9. Verify that the receiver is receiving the multicast stream.
10. Use show commands to verify the routing, VPN, and multicast operation.

Results The configuration and verification parts of this example have been completed. The following section is for your reference.

The relevant sample configuration for the CE1 router follows.

Router CE1

```
interfaces {
    lo0 {
        unit 0 {
            family inet {
                address 192.168.6.1/32 {
                    primary;
                }
            }
        }
    }
    fe-0/1/0 {
        unit 0 {
            family inet {
                address 10.0.67.13/30;
            }
        }
    }
    fe-1/3/0 {
        unit 0 {
            family inet {
                address 10.10.12.1/24;
            }
        }
    }
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface fe-0/1/0.0;
            interface lo0.0;
            interface fe-1/3/0.0;
        }
    }
    pim {
        rp {
            static {
                address 10.10.47.101 {
                    version 2;
                }
```

```
        }
      }
      interface all;
    }
  }
```

The relevant sample configuration for the PE1 router follows.

Router PE1

```
interfaces {
  lo0 {
    unit 0 {
      family inet {
        address 192.168.7.1/32 {
          primary;
        }
      }
    }
  }
  fe-0/1/0 {
    unit 0 {
      family inet {
        address 10.0.67.14/30;
      }
    }
  }
  at-0/2/0 {
    atm-options {
      pic-type atm1;
      vpi 0 {
        maximum-vcs 256;
      }
    }
    unit 0 {
      vci 0.128;
      family inet {
        address 10.0.78.5/32 {
          destination 10.0.78.6;
        }
      }
      family mpls;
    }
  }
  lo0 {
    unit 1 {
      family inet {
        address 10.10.47.101/32;
      }
    }
  }
}
routing-options {
  autonomous-system 0.65010;
}
protocols {
  rsvp {
```

```
        interface fe-0/1/0.0;
        interface at-0/2/0.0;
    }
    mpls {
        label-switched-path to-pe2 {
            to 192.168.9.1;
        }
        interface fe-0/1/0.0;
        interface at-0/2/0.0;
        interface lo0.0;
    }
    bgp {
        group group-mvpn {
            type internal;
            local-address 192.168.7.1;
            family inet-vpn {
                unicast;
            }
            family inet-mvpn {
                signaling;
            }
            neighbor 192.168.9.1;
            neighbor 192.168.8.1;
        }
    }
    ospf {
        traffic-engineering {
            shortcuts;
        }
        area 0.0.0.0 {
            interface at-0/2/0.0;
            interface lo0.0;
        }
    }
    pim {
        rp {
            local {
                address 192.168.7.1;
            }
        }
        interface at-0/2/0.0 {
            mode sparse;
            version 2;
        }
        interface lo0.0 {
            mode sparse;
            version 2;
        }
    }
}
policy-options {
    policy-statement bgp-to-ospf {
        from protocol bgp;
        then accept;
    }
}
```

```
routing-instances {
   vpn-a {
      instance-type vrf;
      interface lo0.1;
      interface fe-0/1/0.0;
      route-distinguisher 65010:1;
      provider-tunnel {
         pim-asm {
            group-address 224.1.1.1;
         }
      }
      vrf-target target:2:1;
      protocols {
         ospf {
            export bgp-to-ospf;
            area 0.0.0.0 {
               interface all;
            }
         }
         pim {
            rp {
               local {
                  address 10.10.47.101;
                  group-ranges {
                     224.1.1.1/32;
                  }
               }
            }
            interface lo0.1 {
               mode sparse;
               version 2;
            }
            interface fe-0/1/0.0 {
               mode sparse;
               version 2;
            }
         }
         mvpn;
      }
   }
}
```

The relevant sample configuration for the P router follows.

Router P

```
interfaces {
   lo0 {
      unit 0 {
         family inet {
            address 192.168.8.1/32 {
               primary;
            }
         }
      }
   }
   at-0/2/0 {
```

```
        atm-options {
            pic-type atm1;
            vpi 0 {
                maximum-vcs 256;
            }
        }
        unit 0 {
            vci 0.128;
            family inet {
                address 10.0.78.6/32 {
                    destination 10.0.78.5;
                }
            }
            family mpls;
        }
    }
    at-0/2/1 {
        atm-options {
            pic-type atm1;
            vpi 0 {
                maximum-vcs 256;
            }
        }
        unit 0 {
            vci 0.128;
            family inet {
                address 10.0.89.5/32 {
                    destination 10.0.89.6;
                }
            }
            family mpls;
        }
    }
}
routing-options {
    autonomous-system 0.65010;
}
protocols {
    rsvp {
        interface at-0/2/0.0;
        interface at-0/2/1.0;
    }
    mpls {
        interface at-0/2/0.0;
        interface at-0/2/1.0;
    }
    bgp {
        group group-mvpn {
            type internal;
            local-address 192.168.8.1;
            family inet {
                unicast;
            }
            family inet-mvpn {
                signaling;
            }
```

```
            neighbor 192.168.9.1;
            neighbor 192.168.7.1;
        }
    }
    ospf {
        traffic-engineering {
            shortcuts;
        }
        area 0.0.0.0 {
            interface lo0.0;
            interface all;
            interface fxp0.0 {
                disable;
            }
        }
    }
    pim {
        rp {
            static {
                address 192.168.7.1 {
                    version 2;
                }
            }
        }
        interface at-0/2/0.0 {
            mode sparse;
            version 2;
        }
        interface at-0/2/1.0 {
            mode sparse;
            version 2;
        }
    }
}
```

The relevant sample configuration for the PE2 router follows.

Router PE2

```
interfaces {
    lo0 {
        unit 0 {
            family inet {
                address 192.168.9.1/32 {
                    primary;
                }
            }
        }
    }
    fe-0/1/0 {
        unit 0 {
            family inet {
                address 10.0.90.13/30;
            }
        }
    }
    at-0/2/1 {
```

```
        atm-options {
            pic-type atm1;
            vpi 0 {
                maximum-vcs 256;
            }
        }
        unit 0 {
            vci 0.128;
            family inet {
                address 10.0.89.6/32 {
                    destination 10.0.89.5;
                }
            }
            family mpls;
        }
    }
    lo0 {
        unit 1 {
            family inet {
                address 10.10.47.100/32;
            }
        }
    }
}
routing-options {
    autonomous-system 0.65010;
}
protocols {
    rsvp {
        interface fe-0/1/0.0;
        interface at-0/2/1.0;
    }
    mpls {
        label-switched-path to-pe1 {
            to 192.168.7.1;
        }
        interface lo0.0;
        interface fe-0/1/0.0;
        interface at-0/2/1.0;
    }
    bgp {
        group group-mvpn {
            type internal;
            local-address 192.168.9.1;
            family inet-vpn {
                unicast;
            }
            family inet-mvpn {
                signaling;
            }
            neighbor 192.168.7.1;
            neighbor 192.168.8.1;
        }
    }
    ospf {
        traffic-engineering {
```

```
            shortcuts;
        }
        area 0.0.0.0 {
            interface lo0.0;
            interface at-0/2/1.0;
        }
    }
    pim {
        rp {
            static {
                address 192.168.7.1 {
                    version 2;
                }
            }
        }
        interface lo0.0 {
            mode sparse;
            version 2;
        }
        interface at-0/2/1.0 {
            mode sparse;
            version 2;
        }
    }
}
policy-options {
    policy-statement bgp-to-ospf {
        from protocol bgp;
        then accept;
    }
}
routing-instances {
    vpn-a {
        instance-type vrf;
        interface fe-0/1/0.0;
        interface lo0.1;
        route-distinguisher 65010:2;
        provider-tunnel {
            pim-asm {
                group-address 224.1.1.1;
            }
        }
        vrf-target target:2:1;
        protocols {
            ospf {
                export bgp-to-ospf;
                area 0.0.0.0 {
                    interface all;
                }
            }
            pim {
                rp {
                    static {
                        address 10.10.47.101;
                    }
                }
```

```
            interface fe-0/1/0.0 {
               mode sparse;
               version 2;
            }
            interface lo0.1 {
               mode sparse;
               version 2;
            }
         }
         mvpn;
      }
   }
}
```

The relevant sample configuration for the CE2 router follows.

Router CE2

```
interfaces {
   lo0 {
      unit 0 {
         family inet {
            address 192.168.0.1/32 {
               primary;
            }
         }
      }
   }
   fe-0/1/0 {
      unit 0 {
         family inet {
            address 10.0.90.14/30;
         }
      }
   }
   fe-1/3/0 {
      unit 0 {
         family inet {
            address 10.10.11.1/24;
         }
         family inet6 {
            address fe80::205:85ff:fe88:ccdb/64;
         }
      }
   }
}
protocols {
   ospf {
      area 0.0.0.0 {
         interface fe-0/1/0.0;
         interface lo0.0;
         interface fe-1/3/0.0;
      }
   }
   pim {
      rp {
         static {
```

```
            address 10.10.47.101 {
              version 2;
            }
          }
        }
        interface all {
          mode sparse;
          version 2;
        }
      }
    }
```

Example: Configuring MBGP Multicast VPN Topology Variations

This section describes how to configure multicast virtual private networks (MVPNs) using multiprotocol BGP (MBGP) (next-generation MVPNs).

- Requirements on page 385
- Overview and Topology on page 385
- Configuring Full Mesh MBGP MVPNs on page 387
- Configuring Sender-Only and Receiver-Only Sites Using PIM ASM Provider Tunnels on page 389
- Configuring Sender-Only, Receiver-Only, and Sender-Receiver MVPN Sites on page 391
- Configuring Hub-and-Spoke MVPNs on page 394

Requirements

To implement multiprotocol BGP-based multicast VPNs, auto-RP, bootstrap router (BSR) RP, and PIM dense mode you need JUNOS Release 9.2 or later.

To implement multiprotocol BGP-based multicast VPNs, sender-only sites, and receiver-only sites you need JUNOS Release 8.4 or later.

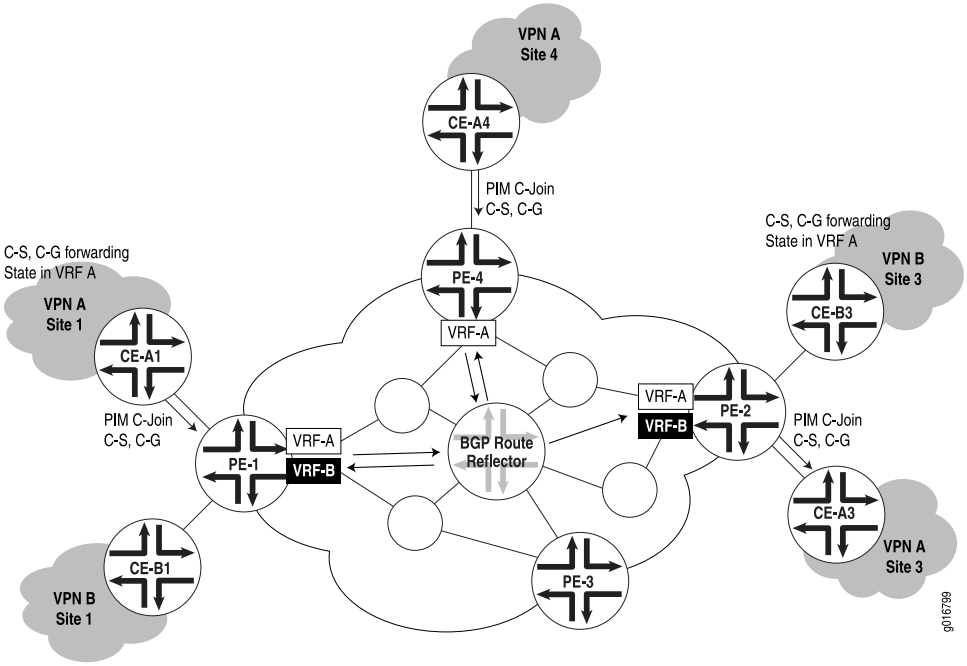
Overview and Topology

You can configure PIM auto-RP, bootstrap router (BSR) RP, PIM dense mode, and mtrace for next generation multicast VPN networks. Auto-RP uses PIM dense mode to propagate control messages and establish RP mapping. You can configure an auto-RP node in one of three different modes: discovery mode, announce mode, and mapping mode. BSR is the IETF standard for RP establishment. A selected router in a network acts as a BSR, which selects a unique RP for different group ranges. BSR messages are flooded using the data tunnel between PE routers. When you enable PIM dense mode, data packets are forwarded to all interfaces except the incoming interface. Unlike PIM sparse mode, where explicit joins are required for data packets to be transmitted downstream, data packets are flooded to all routers in the routing instance in PIM dense mode.

This section shows you how to configure a MVPN using MBGP. If you have multicast VPNs based on draft-rosen, they will continue to work as before and are not affected by the configuration of MVPNs using MBGP.

The network configuration used for most of the examples in this section is shown in Figure 32 on page 386.

Figure 32: MBGP MVPN Topology Variations Diagram



In the figure, two VPNs, VPN A and VPN B, are serviced by the same provider at several sites, two of which have CE routers for both VPN A and VPN B (site 2 is not shown). The PE routers are shown with VRF tables for the VPN CEs for which they have routing information. It is important to note that no multicast protocols are required between the PE routers on the network. The multicast routing information is carried by MBGP between the PE routers. There may be one or more BGP route reflectors in the network. Both VPNs operate independently and are configured separately.

Both the PE and CE routers run PIM sparse mode and maintain forwarding state information about customer source (C-S) and customer group (C-G) multicast components. CE routers still send a customer's PIM join messages (PIM C-Join) from CE to PE, and from PE to CE, as shown in the figure. But on the provider's backbone network, all multicast information is carried by MBGP. The only addition over and above the unicast VPN configuration normally used is the use of a special provider tunnel (provider-tunnel) for carrying PIM sparse mode message content between provider nodes on the network.

There are several scenarios for MVPN configuration using MBGP, depending on whether a customer site has senders (sources) of multicast traffic, has receivers of multicast traffic, or a mixture of senders and receivers. MVPNs can be:

- A full mesh (each MVPN site has both senders and receivers)
- A mixture of sender-only and receiver-only sites
- A mixture of sender-only, receiver-only, and sender-receiver sites
- A hub and spoke (two interfaces between hub PE and hub CE, and all spokes are sender-receiver sites)

Each type of MVPN differs more in the configuration VPN statements than the provider tunnel configuration. For information about configuring VPNs, see the *JUNOS VPNs Configuration Guide*.

Configuring Full Mesh MBGP MVPNs

This example describes how to configure a full mesh MBGP MVPN:

Configuration Steps

Step-by-Step Procedure

In this example, PE-1 connects to VPN A and VPN B at site 1, PE-4 connects to VPN A at site 4, and PE-2 connects to VPN B at site 3. To configure a full mesh MVPN for VPN A and VPN B, perform the following steps:

1. Configure PE-1 (both VPN A and VPN B at site 1):

```
[edit]
routing-instances {
   VPN-A {
      instance-type vrf;
      interface so-6/0/0.0;
      interface so-6/0/1.0;
      provider-tunnel {
         pim-asm {
            group-address 224.1.1.1;
         }
      }
      protocols {
         mvpn;
      }
      route-distinguisher 65535:0;
      vrf-target target:1:1;
   }
   VPN-B {
      instance-type vrf;
      interface ge-0/3/0.0;
      provider-tunnel {
         pim-asm {
            group-address 224.1.1.2;
         }
      }
      protocols {
```

```
        mvpn;
      }
      route-distinguisher 65535:1;
      vrf-target target:1:2;
    }
```

2. Configure PE-4 (VPN A at site 4):

```
[edit]
routing-instances {
  VPN-A {
    instance-type vrf;
    interface so-1/0/0.0;
    provider-tunnel {
      pim-asm {
        group-address 224.1.1.1;
      }
    }
    protocols {
      mvpn;
    }
    route-distinguisher 65535:4;
    vrf-target target:1:1;
  }
```

3. Configure PE-2 (VPN B at site 3):

```
[edit]
routing-instances {
  VPN-B {
    instance-type vrf;
    interface ge-1/3/0.0;
    provider-tunnel {
      pim-asm {
        group-address 224.1.1.2;
      }
    }
    protocols {
      mvpn;
    }
    route-distinguisher 65535:3;
    vrf-target target:1:2;
  }
```

Configuring Sender-Only and Receiver-Only Sites Using PIM ASM Provider Tunnels

This example describes how to configure an MBGP MVPN with a mixture of sender-only and receiver-only sites using PIM-ASM provider tunnels.

Configuration Steps

Step-by-Step Procedure

In this example, PE-1 connects to VPN A (sender-only) and VPN B (receiver-only) at site 1, PE-4 connects to VPN A (receiver-only) at site 4, and PE-2 connects to VPN A (receiver-only) and VPN B (sender-only) at site 3. To configure an MVPN for a mixture of sender-only and receiver-only sites on VPN A and VPN B, perform the following steps:

1. Configure PE-1 (VPN A sender-only and VPN B receiver-only at site 1):

```
[edit]
routing-instances {
   VPN-A {
      instance-type vrf;
      interface so-6/0/0.0;
      interface so-6/0/1.0;
      provider-tunnel {
         pim-asm {
            group-address 224.1.1.1;
         }
      }
      protocols {
         mvpn {
            sender-site;
            route-target {
               export-target unicast;
               import-target target target:1:4;
            }
         }
      }
      route-distinguisher 65535:0;
      vrf-target target:1:1;
      routing-options {
         auto-export;
      }
   }
   VPN-B {
      instance-type vrf;
      interface ge-0/3/0.0;
      provider-tunnel {
         pim-asm {
            group-address 224.1.1.2;
         }
      }
      protocols {
         mvpn {
            receiver-site;
            route-target {
```

```
                export-target target target:1:5;
                import-target unicast;
            }
        }
    }
    route-distinguisher 65535:1;
    vrf-target target:1:2;
    routing-options {
        auto-export;
    }
}
```

2. Configure PE-4 (VPN A receiver-only at site 4):

```
[edit]
routing-instances {
    VPN-A {
        instance-type vrf;
        interface so-1/0/0.0;
        provider-tunnel {
            pim-asm {
                group-address 224.1.1.1;
            }
        }
        protocols {
            mvpn {
                receiver-site;
                route-target {
                    export-target target target:1:4;
                    import-target unicast;
                }
            }
        }
        route-distinguisher 65535:2;
        vrf-target target:1:1;
        routing-options {
            auto-export;
        }
    }
```

3. Configure PE-2 (VPN A receiver-only and VPN B sender-only at site 3):

```
[edit]
routing-instances {
    VPN-A {
        instance-type vrf;
        interface so-2/0/1.0;
        provider-tunnel {
            pim-asm {
                group-address 224.1.1.1;
            }
        }
        protocols {
            mvpn {
                receiver-site;
```

```
            route-target {
               export-target target target:1:4;
               import-target unicast;
            }
         }
      }
      route-distinguisher 65535:3;
      vrf-target target:1:1;
      routing-options {
         auto-export;
      }
   }
   VPN-B {
      instance-type vrf;
      interface ge–1/3/0.0;
      provider-tunnel {
         pim-asm {
            group-address 224.1.1.2;
         }
      }
      protocols {
         mvpn {
            sender-site;
            route-target {
               export-target unicast
               import-target target target:1:5;
            }
         }
      }
      route-distinguisher 65535:4;
      vrf-target target:1:2;
      routing-options {
         auto-export;
      }
   }
```

Configuring Sender-Only, Receiver-Only, and Sender-Receiver MVPN Sites

This example describes how to configure an MBGP MVPN with a mixture of sender-only, receiver-only, and sender-receiver sites.

Configuration Steps

Step-by-Step Procedure

In this example, PE-1 connects to VPN A (sender-receiver) and VPN B (receiver-only) at site 1, PE-4 connects to VPN A (receiver-only) at site 4, and PE-2 connects to VPN A (sender-only) and VPN B (sender-only) at site 3. To configure an MVPN for a mixture of sender-only, receiver-only, and sender-receiver sites for VPN A and VPN B, perform the following steps:

1. Configure PE-1 (VPN A sender-receiver and VPN B receiver-only at site 1):

```
[edit]
routing-instances {
   VPN-A {
      instance-type vrf;
      interface so-6/0/0.0;
      interface so-6/0/1.0;
      provider-tunnel {
         pim-asm {
            group-address 224.1.1.1;
         }
      }
      protocols {
         mvpn {
            route-target {
               export-target unicast target target:1:4;
               import-target unicast target target:1:4 receiver;
            }
         }
      }
      route-distinguisher 65535:0;
      vrf-target target:1:1;
      routing-options {
         auto-export;
      }
   }
   VPN-B {
      instance-type vrf;
      interface ge-0/3/0.0;
      provider-tunnel {
         pim-asm {
            group-address 224.1.1.2;
         }
      }
      protocols {
         mvpn {
            receiver-site;
            route-target {
               export-target target target:1:5;
               import-target unicast;
            }
         }
      }
      route-distinguisher 65535:1;
      vrf-target target:1:2;
      routing-options {
         auto-export;
      }
   }
```

2. Configure PE-4 (VPN A receiver-only at site 4):

```
[edit]
routing-instances {
   VPN-A {
      instance-type vrf;
```

```
    interface so-1/0/0.0;
    provider-tunnel {
      pim-asm {
        group-address 224.1.1.1;
      }
    }
    protocols {
      mvpn {
        receiver-site;
          route-target {
          export-target target target:1:4;
          import-target unicast;
        }
      }
    }
    route-distinguisher 65535:2;
    vrf-target target:1:1;
    routing-options {
      auto-export;
    }
  }
```

3. Configure PE-2 (VPN-A sender-only and VPN-B sender-only at site 3):

```
[edit]
routing-instances {
  VPN-A {
    instance-type vrf;
    interface so-2/0/1.0;
    provider-tunnel {
      pim-asm {
        group-address 224.1.1.1;
      }
    }
    protocols {
      mvpn {
        receiver-site;
          route-target {
          export-target target target:1:4;
          import-target unicast;
        }
      }
    }
    route-distinguisher 65535:3;
    vrf-target target:1:1;
    routing-options {
      auto-export;
    }
  }
  VPN-B {
    instance-type vrf;
    interface ge-1/3/0.0;
    provider-tunnel {
      pim-asm {
        group-address 224.1.1.2;
```

```
        }
    }
    protocols {
        mvpn {
            sender-site;
            route-target {
                export-target unicast;
                import-target target target:1:5;
            }
        }
    }
    route-distinguisher 65535:4;
    vrf-target target:1:2;
    routing-options {
        auto-export;
    }
}
```

Configuring Hub-and-Spoke MVPNs

This example describes how to configure an MBGP MVPN in a hub and spoke topology.

Configuration Steps

Step-by-Step Procedure

In this example, which only configures VPN A, PE-1 connects to VPN A (spoke site) at site 1, PE-4 connects to VPN A (hub site) at site 4, and PE-2 connects to VPN A (spoke site) at site 3. Current support is limited to the case where there are two interfaces between the hub site CE and PE. To configure a hub-and-spoke MVPN for VPN A, perform the following steps:

1. Configure PE-1 for VPN A (spoke site):

```
[edit]
routing-instances {
    VPN-A {
        instance-type vrf;
        interface so-6/0/0.0;
        interface so-6/0/1.0;
        provider-tunnel {
            rsvp-te {
                label-switched-path-template {
                    default-template;
                }
            }
        }
        protocols {
            mvpn {
                route-target {
                    export-target unicast;
                    import-target unicast target target:1:4;
                }
            }
```

```
    }
    route-distinguisher 65535:0;
    vrf-target {
      import target:1:1;
      export target:1:3;
    }
    routing-options {
      auto-export;
    }
  }
```

2. Configure PE-4 for VPN A (hub site):

```
[edit]
routing-instances {
  VPN-A-spoke-to-hub {
    instance-type vrf;
    interface so-1/0/0.0; #receives data and joins from the CE
    protocols {
      mvpn {
        receiver-site;
        route-target {
          export-target target target:1:4;
          import-target unicast;
        }
      }
      ospf {
        export redistribute-vpn; #redistributes VPN routes to CE
        area 0.0.0.0 {
          interface so-1/0/0;
        }
      }
    }
    route-distinguisher 65535:2;
    vrf-target {
      import target:1:3;
    }
    routing-options {
      auto-export;
    }
  }
  VPN-A-hub-to-spoke {
    instance-type vrf;
    interface so-2/0/0.0;  #receives data and joins from the CE
    provider-tunnel {
      rsvp-te {
        label-switched-path-template {
          default-template;
        }
      }
    }
    protocols {
      mvpn {
        sender-site;
        route-target {
```

```
            import-target target target:1:3;
            export-target unicast;
          }
        }
        ospf {
          export redistribute-vpn;  #redistributes VPN routes to CE
          area 0.0.0.0 {
            interface so-2/0/0;
          }
        }
      }
      route-distinguisher 65535:2;
      vrf-target {
        import target:1:1;
      }
      routing-options {
        auto-export;
      }
    }
```

3. Configure PE-2 for VPN A (spoke site):

```
[edit]
routing-instances {
  VPN-A {
    instance-type vrf;
    interface so-2/0/1.0;
    provider-tunnel {
      rsvp-te {
        label-switched-path-template {
          default-template;
        }
      }
    }
    protocols {
      mvpn {
        route–target {
          import-target target target:1:4;
          export-target unicast;
        }
      }
    }
    route-distinguisher 65535:3;
    vrf-target {
      import target:1:1;
      export target:1:3;
    }
    routing-options {
      auto-export;
    }
  }
```

MBGP Multicast VPN Extranets Overview

A multicast VPN (MVPN) extranet enables service providers to forward IP multicast traffic originating in one VPN routing and forwarding (VRF) instance to receivers in a different VRF instance. This capability is also know as overlapping MVPNs.

The MVPN extranet feature supports the following traffic flows:

- A receiver in one VFR can receive multicast traffic from a source connected to a different router in a different VRF.
- A receiver in one VFR can receive multicast traffic from a source connected to the same router in a different VRF.
- A receiver in one VFR can receive multicast traffic from a source connected to a different router in the same VRF.
- A receiver in one VFR can be prevented from receiving multicast traffic from a specific source in a different VRF.

MBGP Multicast VPN Extranets Application

An MVPN extranet is useful in the following applications:

Mergers and Data Sharing

An MVPN extranet is useful when there are business partnerships between different enterprise VPN customers that require them to be able to communicate with one another. For example, a wholesale company might want to broadcast inventory to its contractors and resellers. An MVPN extranet is also useful when companies merge and one set of VPN sites needs to receive content from another VPN. The enterprises involved in the merger are different VPN customers from the service provider point of view. The MVPN extranet makes the connectivity possible.

Video Distribution

Another use for MVPN extranets is video multicast distribution from a video headend to receiving sites. Sites within a given multicast VPN might be in different organizations. The receivers can subscribe to content from a specific content provider.

The PE routers on the MVPN provider network learn about the sources and receivers using MVPN mechanisms. These PE routers can use selective trees as the multicast distribution mechanism in the backbone. The network carries traffic belonging only to a specified set of one or more multicast groups, from one or more multicast VPNs. As a result, this model facilitates the distribution of content from multiple providers on a selective basis if desired.

Financial Services

A third use for MVPN extranets is enterprise and financial services infrastructures. The delivery of financial data, such as financial market updates, stock ticker values, and financial TV channels, is an example of an application that must deliver the same data stream to hundreds and potentially thousands of end users. The content

distribution mechanisms largely rely on multicast within the financial provider network. In this case, there could also be an extensive multicast topology within brokerage firms and banks networks to enable further distribution of content and for trading applications. Financial service providers require traffic separation between customers accessing the content, and MVPN extranets provide this separation.

MBGP Multicast VPN Extranets Configuration Guidelines

When configuring MVPN extranets, remember the following considerations:

- If there is more than one VRF routing instance on a provider edge (PE) router that has receivers interested in receiving multicast traffic from the same source, virtual tunnel (VT) interfaces must be configured on all instances.
- For auto-RP operation, the mapping agent must be configured on at least two PEs in the extranet network.
- For asymmetrically configured extranets using auto-RP, when one VRF instance is the only instance that imports routes from all other extranet instances, the mapping agent must be configured in the VRF that can receive all RP discovery messages from all VRF instances and mapping-agent election should be disabled.
- For bootstrap router (BSR) operation, the elected BSR must be on a PE router that is configured with one extranet instance and that instance must import routes from all VRF instances in the extranet.
- RSVP-TE point-to-multipoint LSPs must be used for the provider tunnels.
- PIM dense mode is not supported in the MVPN extranets VRF instances.

Example: Configuring MBGP Multicast VPN Extranets

This example provides a step-by-step procedure to configure multicast VPN extranets using static rendezvous points. It is organized in the following sections:

- Requirements on page 398
- Overview and Topology on page 399
- Configuration on page 400

Requirements

This example uses the following hardware and software components:

- JUNOS Release 9.5 or later.
- Six M Series, T Series, TX Series, or MX Series Juniper routers.
- One adaptive services PIC or MultiServices PIC in each of the M Series or T Series routers acting as PE routers.
- One host systems capable of sending multicast traffic and supporting the Internet Group Management Protocol (IGMP).
- Three host systems capable of receiving multicast traffic and supporting IGMP.

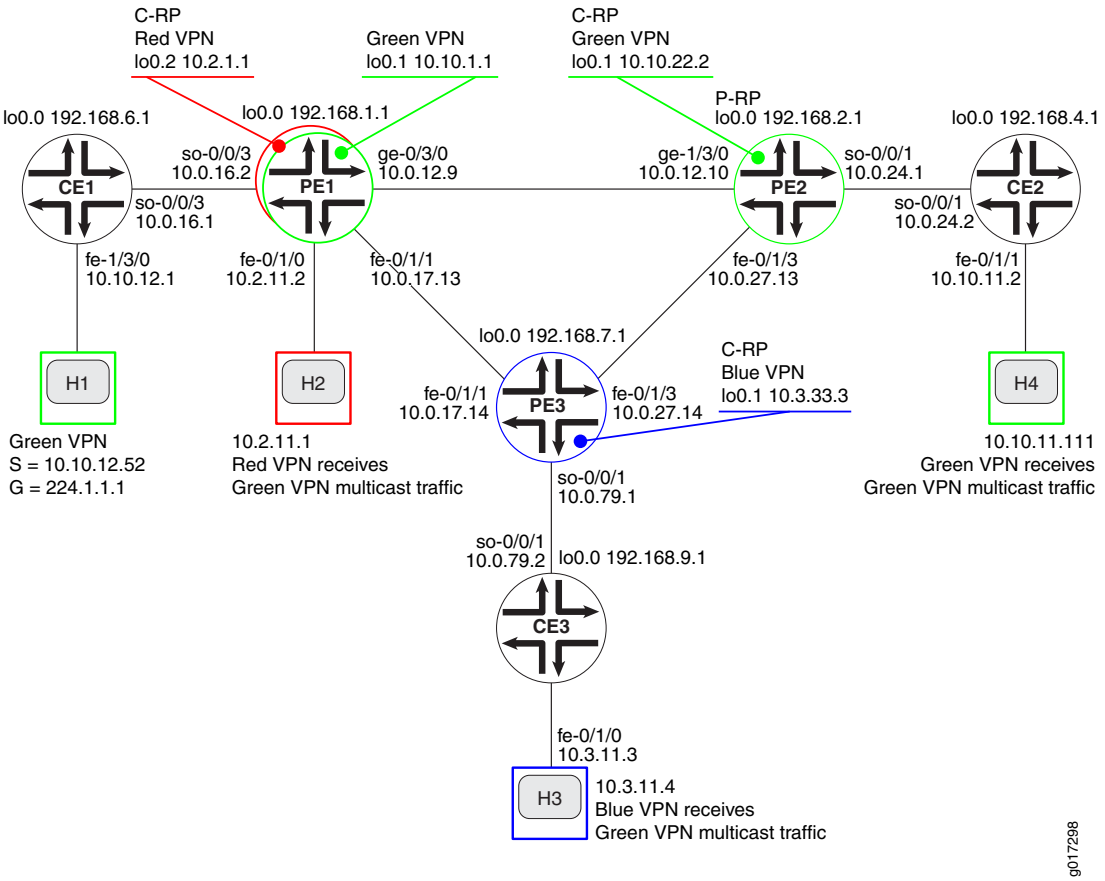
Overview and Topology

In the topology shown in Figure 33 on page 399:

- Host H1 is the source for group 244.1.1.1 in the green VPN.
- The multicast traffic originating at source H1 can be received by host H4 connected to router CE2 in the green VPN.
- The multicast traffic originating at source H1 can be received by host H3 connected to router CE3 in the blue VPN.
- The multicast traffic originating at source H1 can be received by host H2 directly connected to router PE1 in the red VPN.
- Any host can be a sender site or receiver site.

The topology of the network is shown in Figure 33 on page 399.

Figure 33: MVPN Extranets Topology Diagram



Configuration

NOTE: In any configuration session, it is good practice to verify periodically that the configuration can be committed using the commit check command.

In this example, the router being configured is identified using the following command prompts:

- ce1 identifies the customer edge 1 (CE1) router
- pe1 identifies the provider edge 1 (PE1) router
- ce2 identifies the customer edge 2 (CE2) router
- pe2 identifies the provider edge 2 (PE2) router
- ce3 identifies the customer edge 3 (CE3) router
- pe3 identifies the provider edge 3 (PE3) router

To configure multicast VPN extranets, perform the following tasks:

- Configuring Interfaces on page 400
- Configuring an IGP in the Core on page 402
- Configuring BGP in the Core on page 404
- Configuring LDP on page 405
- Configuring RSVP on page 406
- Configuring MPLS on page 407
- Configuring the VRF Routing Instances on page 408
- Configuring MVPN Extranet Policy on page 411
- Configuring CE-PE BGP on page 415
- Configuring PIM on the PE Routers on page 417
- Configuring PIM on the CE Routers on page 419
- Configuring the Rendezvous Points on page 419
- Testing MVPN Extranets on page 422

Configuring Interfaces

Step-by-Step Procedure

1. On each router, configure an IP address on the loopback logical interface 0 (lo0.0).

   user@ce1# **set interfaces lo0 unit 0 family inet address 192.168.6.1/32 primary**

   user@pe1# **set interfaces lo0 unit 0 family inet address 192.168.1.1/32 primary**

```
user@pe2# set interfaces lo0 unit 0 family inet address 192.168.2.1/32
    primary

user@ce2# set interfaces lo0 unit 0 family inet address 192.168.4.1/32 primary

user@pe3# set interfaces lo0 unit 0 family inet address 192.168.7.1/32
    primary

user@ce3# set interfaces lo0 unit 0 family inet address 192.168.9.1/32 primary
```

Use the show interfaces terse command to verify that the correct IP address is configured on the loopback interface.

2. On the PE and CE routers, configure the IP address and protocol family on the Fast Ethernet and Gigabit Ethernet interfaces. Specify the inet address family type.

```
user@ce1# set interfaces fe-1/3/0 unit 0 family inet address 10.10.12.1/24

user@pe1# set interfaces fe-0/1/0 unit 0 description "to H2"
user@pe1# set interfaces fe-0/1/0 unit 0 family inet address 10.2.11.2/30
user@pe1# set interfaces fe-0/1/1 unit 0 description "to PE3 fe-0/1/1.0"
user@pe1# set interfaces fe-0/1/1 unit 0 family inet address 10.0.17.13/30
user@pe1# set interfaces ge-0/3/0 unit 0 family inet address 10.0.12.9/30

user@pe2# set interfaces fe-0/1/3 unit 0 description "to PE3 fe-0/1/3.0"
user@pe2# set interfaces fe-0/1/3 unit 0 family inet address 10.0.27.13/30
user@pe2# set interfaces ge-1/3/0 unit 0 description "to PE1 ge-0/3/0.0"
user@pe2# set interfaces ge-1/3/0 unit 0 family inet address 10.0.12.10/30

user@ce2# set interfaces fe-0/1/1 unit 0 description "to H4"
user@ce2# set interfaces fe-0/1/1 unit 0 family inet address 10.10.11.2/24

user@pe3# set interfaces fe-0/1/1 unit 0 description "to PE1 fe-0/1/1.0"
user@pe3# set interfaces fe-0/1/1 unit 0 family inet address 10.0.17.14/30
user@pe3# set interfaces fe-0/1/3 unit 0 description "to PE2 fe-0/1/3.0"
user@pe3# set interfaces fe-0/1/3 unit 0 family inet address 10.0.27.14/30

user@ce3# set interfaces fe-0/1/0 unit 0 description "to H3"
user@ce3# set interfaces fe-0/1/0 unit 0 family inet address 10.3.11.3/24
```

Use the show interfaces terse command to verify that the correct IP address and address family type areconfigured on the interfaces.

3. On the PE and CE routers, configure the SONET interfaces. Specify the inet address family type, and local IP address.

```
user@ce1# set interfaces so-0/0/3 unit 0 description "to PE1 so-0/0/3.0;"
user@ce1# set interfaces so-0/0/3 unit 0 family inet address 10.0.16.1/30

user@pe1# set interfaces so-0/0/3 unit 0 description "to CE1 so-0/0/3.0"
user@pe1# set interfaces so-0/0/3 unit 0 family inet address 10.0.16.2/30
```

```
user@pe2# set interfaces so-0/0/1 unit 0 description "to CE2 so-0/0/1:0.0"
user@pe2# set interfaces so-0/0/1 unit 0 family inet address 10.0.24.1/30

user@ce2# set interfaces so-0/0/1 unit 0 description "to PE2 so-0/0/1"
user@ce2# set interfaces so-0/0/1 unit 0 family inet address 10.0.24.2/30

user@pe3# set interfaces so-0/0/1 unit 0 description "to CE3 so-0/0/1.0"
user@pe3# set interfaces so-0/0/1 unit 0 family inet address 10.0.79.1/30

user@ce3# set interfaces so-0/0/1 unit 0 description "to PE3 so-0/0/1"
user@ce3# set interfaces so-0/0/1 unit 0 family inet address 10.0.79.2/30
```

Use the show configuration interfaces command to verify that the correct IP address and address family type are configured on the interfaces.

4. On each router, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

5. Use the ping command to verify unicast connectivity between each:
   - CE router and the attached host
   - CE router and the directly attached interface on the PE router.
   - PE router and the directly attached interfaces on the other PE routers.

Configuring an IGP in the Core

Step-by-Step Procedure

On the PE routers, configure an interior gateway protocol such as OSPF or IS-IS. This example configures Open Shortest Path First (OSPF).

1. Specify the lo0.0 and SONET core-facing logical interfaces.

```
user@pe1# set protocols ospf area 0.0.0.0 interface ge-0/3/0.0 metric 100
user@pe1# set protocols ospf area 0.0.0.0 interface fe-0/1/1.0 metric 100
user@pe1# set protocols ospf area 0.0.0.0 interface lo0.0 passive
user@pe1# set protocols ospf area 0.0.0.0 interface fxp0.0 disable

user@pe2# set protocols ospf area 0.0.0.0 interface fe-0/1/3.0 metric 100
user@pe2# set protocols ospf area 0.0.0.0 interface ge-1/3/0.0 metric 100
user@pe2# set protocols ospf area 0.0.0.0 interface lo0.0 passive
user@pe2# set protocols ospf area 0.0.0.0 interface fxp0.0 disable
```

```
user@pe3# set protocols ospf area 0.0.0.0 interface lo0.0 passive
user@pe3# set protocols ospf area 0.0.0.0 interface fe-0/1/3.0 metric 100
user@pe3# set protocols ospf area 0.0.0.0 interface fe-0/1/1.0 metric 100
user@pe3# set protocols ospf area 0.0.0.0 interface fxp0.0 disable
```

2. On the PE routers, configure a router ID.

```
user@pe1# set routing-options router-id 192.168.1.1

user@pe2# set routing-options router-id 192.168.2.1

user@pe3# set routing-options router-id 192.168.7.1
```

Use the show ospf overview and show configuration protocols ospf commands to verify that the correct interfaces have been configured for the OSPF protocol.

3. On the PE routers, configure OSPF traffic engineering support. Enabling traffic engineering extensions supports the Constrained Shortest Path First algorithm, which is needed to support Resource Reservation Protocol - Traffic Engineering (RSVP-TE) point-to-multipoint label-switched paths (LSPs). If you are configuring IS-IS, traffic engineering is supported without any additional configuration.

```
user@pe1# set protocols ospf traffic-engineering

user@pe2# set protocols ospf traffic-engineering

user@pe3# set protocols ospf traffic-engineering
```

Use the show ospf overview and show configuration protocols ospf commands to verify that traffic engineering support is enabled for the OSPF protocol.

4. On the PE routers, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

5. On the PE routers, verify that the OSPF neighbors form adjacencies using the following command:

```
user@PE1> show ospf neighbors

Address          Interface              State     ID               Pri  Dead
10.0.17.14       fe-0/1/1.0             Full      192.168.7.1      128    32
10.0.12.10       ge-0/3/0.0             Full      192.168.2.1      128    33
```

Verify that the neighbor state with the other two PE routers is Full.

Configuring BGP in the Core

Step-by-Step Procedure

1. On the PE routers, configure BGP. Configure the BGP local autonomous system number.

```
user@pe1# set routing-options autonomous-system 65000

user@pe2# set routing-options autonomous-system 65000

user@pe3# set routing-options autonomous-system 65000
```

2. Configure the BGP peer groups. Configure the local address as the lo0.0 address on the router. The neighbor addresses are the lo0.0 addresses of the other PE routers.

   The unicast statement enables the router to use BGP to advertise network layer reachability information (NLRI). The signaling statement enables the router to use BGP as the signaling protocol for the VPN.

```
user@pe1# set protocols bgp group group-mvpn type internal
user@pe1# set protocols bgp group group-mvpn local-address 192.168.1.1
user@pe1# set protocols bgp group group-mvpn family inet-vpn unicast
user@pe1# set protocols bgp group group-mvpn family inet-mvpn signaling
user@pe1# set protocols bgp group group-mvpn neighbor 192.168.2.1
user@pe1# set protocols bgp group group-mvpn neighbor 192.168.7.1

user@pe2# set protocols bgp group group-mvpn type internal
user@pe2# set protocols bgp group group-mvpn local-address 192.168.2.1
user@pe2# set protocols bgp group group-mvpn family inet-vpn unicast
user@pe2# set protocols bgp group group-mvpn family inet-mvpn signaling
user@pe2# set protocols bgp group group-mvpn neighbor 192.168.1.1
user@pe2# set protocols bgp group group-mvpn neighbor 192.168.7.1

user@pe3# set protocols bgp group group-mvpn type internal
user@pe3# set protocols bgp group group-mvpn local-address 192.168.7.1
user@pe3# set protocols bgp group group-mvpn family inet-vpn unicast
user@pe3# set protocols bgp group group-mvpn family inet-mvpn signaling
user@pe3# set protocols bgp group group-mvpn neighbor 192.168.1.1
user@pe3# set protocols bgp group group-mvpn neighbor 192.168.2.1
```

3. On the PE routers, commit the configuration:

```
user@host> commit check

configuration check succeeds
```

```
user@host> commit

commit complete
```

4. On the PE routers, verify that the BGP neighbors form a peer session using the following command:

```
user@PE1> show bgp group

Group Type: Internal    AS: 65000                  Local AS: 65000
  Name: group-mvpn      Index: 0                   Flags: Export Eval
  Holdtime: 0
  Total peers: 2        Established: 2
  192.168.2.1+54883
  192.168.7.1+58933
  bgp.l3vpn.0: 0/0/0/0
  bgp.mvpn.0: 0/0/0/0

Groups: 1 Peers: 2   External: 0   Internal: 2   Down peers: 0  Flaps: 0
Table        Tot Paths Act Paths Suppressed  History Damp State  Pending
bgp.l3vpn.0          0          0         0        0         0       0
bgp.mvpn.0           0          0         0        0         0       0
```

Verify that the peer state for the other two PE routers is Established and that the lo0.0 addresses of the other PE routers are shown as peers.

Configuring LDP

Step-by-Step Procedure

1. On the PE routers, configure LDP to support unicast traffic. Specify the core-facing Fast Ethernet and Gigabit Ethernet interfaces between the PE routers. Also configure LDP specifying the lo0.0 interface. As a best practice, disable LDP on the fxp0 interface.

```
user@pe1# set protocols ldp deaggregate
user@pe1# set protocols ldp interface fe-0/1/1.0
user@pe1# set protocols ldp interface ge-0/3/0.0
user@pe1# set protocols ldp interface fxp0.0 disable
user@pe1# set protocols ldp interface lo0.0

user@pe2# set protocols ldp deaggregate
user@pe2# set protocols ldp interface fe-0/1/3.0
user@pe2# set protocols ldp interface ge-1/3/0.0
user@pe2# set protocols ldp interface fxp0.0 disable
user@pe2# set protocols ldp interface lo0.0

user@pe3# set protocols ldp deaggregate
user@pe3# set protocols ldp interface fe-0/1/1.0
user@pe3# set protocols ldp interface fe-0/1/3.0
user@pe3# set protocols ldp interface fxp0.0 disable
```

```
user@pe3# set protocols ldp interface lo0.0
```

2. On the PE routers, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

3. On the PE routers, use the show ldp route command to verify the LDP route.

```
user@PE1> show ldp route

Destination         Next-hop intf/lsp               Next-hop address
 10.0.12.8/30       ge-0/3/0.0
 10.0.12.9/32
 10.0.17.12/30      fe-0/1/1.0
 10.0.17.13/32
 10.0.27.12/30      fe-0/1/1.0                      10.0.17.14
                    ge-0/3/0.0                      10.0.12.10
 192.168.1.1/32     lo0.0
 192.168.2.1/32     ge-0/3/0.0                      10.0.12.10
 192.168.7.1/32     fe-0/1/1.0                      10.0.17.14
 224.0.0.5/32
 224.0.0.22/32
```

Verify that a next-hop interface and next-hop address have been established for each remote destination in the core network. Notice that local destinations do not have next-hop interfaces, and remote destinations outside the core do not have next-hop addresses.

Configuring RSVP

Step-by-Step Procedure

1. On the PE routers, configure RSVP. Specify the core-facing Fast Ethernet and Gigabit Ethernet interfaces that participate in the LSP. Also specify the lo0.0 interface. As a best practice, disable RSVP on the fxp0 interface.

```
user@pe1# set protocols rsvp interface ge-0/3/0.0
user@pe1# set protocols rsvp interface fe-0/1/1.0
user@pe1# set protocols rsvp interface lo0.0
user@pe1# set protocols rsvp interface fxp0.0 disable

user@pe2# set protocols rsvp interface fe-0/1/3.0
user@pe2# set protocols rsvp interface ge-1/3/0.0
user@pe2# set protocols rsvp interface lo0.0
```

```
user@pe2# set protocols rsvp interface fxp0.0 disable

user@pe3# set protocols rsvp interface fe-0/1/3.0
user@pe3# set protocols rsvp interface fe-0/1/1.0
user@pe3# set protocols rsvp interface lo0.0
user@pe3# set protocols rsvp interface fxp0.0 disable
```

2. On the PE routers, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

Verify these steps using the show configuration protocols rsvp command. You can verify the operation of RSVP only after the LSP is established in Step 10 of "Configuring MVPN Extranet Policy".

Configuring MPLS

Step-by-Step Procedure

1. On the PE routers, configure MPLS. Specify the core-facing Fast Ethernet and Gigabit Ethernet interfaces that participate in the LSP. As a best practice, disable MPLS on the fxp0 interface.

```
user@pe1# set protocols mpls interface ge-0/3/0.0
user@pe1# set protocols mpls interface fe-0/1/1.0
user@pe1# set protocols mpls interface fxp0.0 disable

user@pe2# set protocols mpls interface fe-0/1/3.0
user@pe2# set protocols mpls interface ge-1/3/0.0
user@pe2# set protocols mpls interface fxp0.0 disable

user@pe3# set protocols mpls interface fe-0/1/3.0
user@pe3# set protocols mpls interface fe-0/1/1.0
user@pe3# set protocols mpls interface fxp0.0 disable
```

Use the show configuration protocols mpls command to verify that the ore-facing Fast Ethernet and Gigabit Ethernet interfaces are configured for MPLS.

2. On the PE routers, configure the core-facing interfaces associated with the LSP. Specify the mpls address family type.

```
user@pe1# set interfaces fe-0/1/1 unit 0 family mpls
user@pe1# set interfaces ge-0/3/0 unit 0 family mpls
```

```
user@pe2# set interfaces fe-0/1/3 unit 0 family mpls
user@pe2# set interfaces ge-1/3/0 unit 0 family mpls

user@pe3# set interfaces fe-0/1/3 unit 0 family mpls
user@pe3# set interfaces fe-0/1/1 unit 0 family mpls
```

Use the show mpls interface command to verify that the core-facing interfaces have the MPLS address family configured.

3. On the PE routers, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

You can verify the operation of MPLS after the LSP is established.

Configuring the VRF Routing Instances

Step-by-Step Procedure

1. On the PE1 router, configure the routing instance for the green and red VPNs. Specify the vrf instance type and specify the customer-facing SONET interfaces.

   Configure a virtual tunnel (VT) interface on all MVPN routing instances on each PE where hosts in different instances need to receive multicast traffic from the same source.

```
user@pe1# set routing-instances green instance-type vrf
user@pe1# set routing-instances green interface so-0/0/3.0
user@pe1# set routing-instances green interface vt-1/2/0.1 multicast
user@pe1# set routing-instances green interface lo0.1

user@pe1# set routing-instances red instance-type vrf
user@pe1# set routing-instances red interface fe-0/1/0.0
user@pe1# set routing-instances red interface vt-1/2/0.2
user@pe1# set routing-instances red interface lo0.2
```

   Use the show configuration routing-instances green and show configuration routing-instances red commands to verify that the virtual tunnel interfaces have been correctly configured.

2. On the PE2 router, configure the routing instance for the green VPN. Specify the vrf instance type and specify the customer-facing SONET interfaces.

```
user@pe2# set routing-instances green instance-type vrf
user@pe2# set routing-instances green interface so-0/0/1.0
user@pe2# set routing-instances green interface vt-1/2/0.1
user@pe2# set routing-instances green interface lo0.1
```

Use the show configuration routing-instances green command.

3. On the PE3 router, configure the routing instance for the blue VPN. Specify the vrf instance type and specify the customer-facing SONET interfaces.

```
user@pe3# set routing-instances blue instance-type vrf
user@pe3# set routing-instances blue interface so-0/0/1.0
user@pe3# set routing-instances blue interface vt-1/2/0.3
user@pe3# set routing-instances blue interface lo0.1
```

Use the show configuration routing-instances blue command to verify that the instance type has been configured correctly and the correct interfaces have been configured in the routing instance.

4. On the PE1 router, configure a route distinguisher for the green and red routing instances. A route distinguisher allows the router to distinguish between two identical IP prefixes used as VPN routes.

TIP: To help in troubleshooting, this example configures the route distinguisher to match the router ID. This allows you to associate a route with the router that advertised it.

```
user@pe1# set routing-instances green route-distinguisher 192.168.1.1:1

user@pe1# set routing-instances red route-distinguisher 192.168.1.1:2
```

5. On the PE2 router, configure a route distinguisher for the green routing instance.

```
user@pe2# set routing-instances green route-distinguisher 192.168.2.1:1
```

6. On the PE3 router, configure a route distinguisher for the blue routing instance.

```
user@pe3# set routing-instances blue route-distinguisher 192.168.7.1:3
```

7. On the PE routers, configure the VPN routing instance for multicast support.

```
user@pe1# set routing-instances green protocols mvpn
user@pe1# set routing-instances red protocols mvpn

user@pe2# set routing-instances green protocols mvpn

user@pe3# set routing-instances blue protocols mvpn
```

Use the show configuration routing-instance command to verify that the route distinguisher is configured correctly and the MVPN protocol is enabled in the routing instance.

8. On the PE routers, configure an IP address on additional loopback logical interfaces. These logical interfaces are used as the loopback addresses for the VPNs.

```
user@pe1# set interfaces lo0 unit 1 description "green VRF loopback"
user@pe1# set interfaces lo0 unit 1 family inet address 10.10.1.1/32
user@pe1# set interfaces lo0 unit 2 description "red VRF loopback"
user@pe1# set interfaces lo0 unit 2 family inet address 10.2.1.1/32

user@pe2# set interfaces lo0 unit 1 description "green VRF loopback"
user@pe2# set interfaces lo0 unit 1 family inet address 10.10.22.2/32

user@pe3# set interfaces lo0 unit 1 description "blue VRF loopback"
user@pe3# set interfaces lo0 unit 1 family inet address 10.3.33.3/32
```

   Use the show interfaces terse command to verify that the loopback logical interfaces are correctly configured.

9. On the PE routers, configure virtual tunnel interfaces. These interfaces are used in VRF instances where multicast traffic arriving on a provider tunnel needs to be forwarded to multiple VPNs.

```
user@pe1# set interfaces vt-1/2/0 unit 1 description "green VRF multicast
   vt"
user@pe1# set interfaces vt-1/2/0 unit 1 family inet
user@pe1# set interfaces vt-1/2/0 unit 2 description "red VRF unicast and
   multicast vt"
user@pe1# set interfaces vt-1/2/0 unit 2 family inet
user@pe1# set interfaces vt-1/2/0 unit 3 description "blue VRF multicast vt"
user@pe1# set interfaces vt-1/2/0 unit 3 family inet

user@pe2# set interfaces vt-1/2/0 unit 1 description "green VRF unicast and
   multicast vt"
user@pe2# set interfaces vt-1/2/0 unit 1 family inet
user@pe2# set interfaces vt-1/2/0 unit 3 description "blue VRF unicast and
   multicast vt"
user@pe2# set interfaces vt-1/2/0 unit 3 family inet

user@pe3# set interfaces vt-1/2/0 unit 3 description "blue VRF unicast and
   multicast vt"
user@pe3# set interfaces vt-1/2/0 unit 3 family inet
```

   Use the show interfaces terse command to verify that the virtual tunnel interfaces have the correct address family type configured.

10. On the PE routers, configure the provider tunnel.

```
user@pe1# set routing-instances green provider-tunnel rsvp-te
   label-switched-path-template default-template
user@pe1# set routing-instances red provider-tunnel rsvp-te
   label-switched-path-template default-template
```

```
user@pe2# set routing-instances green provider-tunnel rsvp-te
  label-switched-path-template default-template

user@pe3# set routing-instances blue provider-tunnel rsvp-te
  label-switched-path-template default-template
```

Use the show configuration routing-instance command to verify that the provider tunnel is configured to use the default LSP template.

NOTE: You cannot commit the configuration for the VRF instance until you configure the VRF target in the next section.

Configuring MVPN Extranet Policy

Step-by-Step Procedure

1. On the PE routers, define the VPN community name for the route targets for each VPN. The community names are used in the VPN import and export policies.

```
user@pe1# set policy-options community green-com members target:65000:1
user@pe1# set policy-options community red-com members target:65000:2
user@pe1# set policy-options community blue-com members target:65000:3

user@pe2# set policy-options community green-com members target:65000:1
user@pe2# set policy-options community red-com members target:65000:2
user@pe2# set policy-options community blue-com members target:65000:3

user@pe3# set policy-options community green-com members target:65000:1
user@pe3# set policy-options community red-com members target:65000:2
user@pe3# set policy-options community blue-com members target:65000:3
```

   Use the show policy-options command to verify that the correct VPN community name and route target are configured.

2. On the PE routers, configure the VPN import policy. Include the community name of the route targets you want to accept. Do not include the community name of the route targets you do not want to accept. For example, omit the community name for routes from the VPN of a multicast sender from which you do not want to receive multicast traffic.

```
user@pe1# set policy-options policy-statement green-red-blue-import term t1
  from community green-com
user@pe1# set policy-options policy-statement green-red-blue-import term t1
  from community red-com
user@pe1# set policy-options policy-statement green-red-blue-import term t1
  from community blue-com
user@pe1# set policy-options policy-statement green-red-blue-import term t1
  then accept
```

```
user@pe1# set policy-options policy-statement green-red-blue-import term t2
  then reject

user@pe2# set policy-options policy-statement green-red-blue-import term t1
  from community green-com
user@pe2# set policy-options policy-statement green-red-blue-import term t1
  from community red-com
user@pe2# set policy-options policy-statement green-red-blue-import term t1
  from community blue-com
user@pe2# set policy-options policy-statement green-red-blue-import term t1
  then accept
user@pe2# set policy-options policy-statement green-red-blue-import term t2
  then reject

user@pe3# set policy-options policy-statement green-red-blue-import term t1
  from community green-com
user@pe3# set policy-options policy-statement green-red-blue-import term t1
  from community red-com
user@pe3# set policy-options policy-statement green-red-blue-import term t1
  from community blue-com
user@pe3# set policy-options policy-statement green-red-blue-import term t1
  then accept
user@pe3# set policy-options policy-statement green-red-blue-import term t2
  then reject
```

Use the show policy green-red-blue-import command to verify that the VPN import policy is correctly configured.

3. On the PE routers, apply the VRF import policy. In this example the policy is defined in a policy-statement policy, and target communities are defined under the [edit policy-options] hierarchy level.

```
user@pe1# set routing-instances green vrf-import green-red-blue-import
user@pe1# set routing-instances red vrf-import green-red-blue-import

user@pe2# set routing-instances green vrf-import green-red-blue-import

user@pe3# set routing-instances blue vrf-import green-red-blue-import
```

Use the show configuration routing-instances command to verify that the correct VRF import policy has been applied.

4. On the PE routers, configure VRF export targets. The vrf-target statement and export option cause the routes being advertised to be labeled with the target community.

For the PE3 router the vrf-target statement is included without specifying the export option. If you do not specify the import or export options, default VRF import and export policies are generated that accept imported routes and tag exported routes with the specified target community.

NOTE: You must configure the same route target on each PE router for a given VPN routing instance.

```
user@pe1# set routing-instances green vrf-target export target:65000:1
user@pe1# set routing-instances red vrf-target export target:65000:2

user@pe2# set routing-instances green vrf-target export target:65000:1

user@pe3# set routing-instances blue vrf-target target:65000:3
```

Use the show configuration routing-instances command to verify that the correct VRF export targets have been configured.

5. On the PE routers, configure automatic exporting of routes between VRF instances. When you include the auto-export statement, the vrf-import and vrf-export policies are compared across all VRF instances. If there is a common route target community between the instances, the routes are shared. In this example, the auto-export statement must be included under all instances that need to send traffic to and receive traffic from another instance located on the same router.

```
user@pe1# set routing-instances green routing-options auto-export
user@pe1# set routing-instances red routing-options auto-export

user@pe2# set routing-instances green routing-options auto-export

user@pe3# set routing-instances blue routing-options auto-export
```

6. On the PE routers, configure the load balance policy statement. Load balancing leads to better utilization of the available links but is not required for MVPN extranets. It is included here as a best practice.

```
user@pe1# set policy-options policy-statement load-balance then load-balance
  per-packet

user@pe2# set policy-options policy-statement load-balance then load-balance
  per-packet

user@pe3# set policy-options policy-statement load-balance then load-balance
  per-packet
```

Use the show policy-options command to verify that the load balance policy statement has been correctly configured.

7. On the PE routers, apply the load balance policy.

```
user@pe1# set routing-options forwarding-table export load-balance

user@pe2# set routing-options forwarding-table export load-balance
```

```
user@pe3# set routing-options forwarding-table export load-balance
```

8. On the PE routers, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

9. On the PE routers, use the show rsvp neighbor command to verify that the RSVP neighbors are established.

```
user@PE1> show rsvp neighbor

RSVP neighbor: 2 learned
Address            Idle Up/Dn LastChange HelloInt HelloTx/Rx MsgRcvd
10.0.17.14            5  1/0       43:52        9   293/293  247
10.0.12.10            0  1/0       50:15        9   336/336  140
```

Verify that the other PE routers are listed as RSVP neighbors.

10. On the PE routers, display the MPLS LSPs using the following commands:

```
user@PE1> show mpls lsp p2mp

Ingress LSP: 2 sessions
P2MP name: 192.168.1.1:1:mvpn:green, P2MP branch count: 2
To              From            State Rt P     ActivePath       LSPname
192.168.2.1     192.168.1.1     Up     0 *
192.168.2.1:192.168.1.1:1:mvpn:green
192.168.7.1     192.168.1.1     Up     0 *
192.168.7.1:192.168.1.1:1:mvpn:green
P2MP name: 192.168.1.1:2:mvpn:red, P2MP branch count: 2
To              From            State Rt P     ActivePath       LSPname
192.168.2.1     192.168.1.1     Up     0 *
192.168.2.1:192.168.1.1:2:mvpn:red
192.168.7.1     192.168.1.1     Up     0 *
192.168.7.1:192.168.1.1:2:mvpn:red
Total 4 displayed, Up 4, Down 0

Egress LSP: 2 sessions
P2MP name: 192.168.2.1:1:mvpn:green, P2MP branch count: 1
To              From            State   Rt Style Labelin Labelout LSPname

192.168.1.1     192.168.2.1     Up     0  1 SE  299888        3
192.168.1.1:192.168.2.1:1:mvpn:green
P2MP name: 192.168.7.1:3:mvpn:blue, P2MP branch count: 1
To              From            State   Rt Style Labelin Labelout LSPname

192.168.1.1     192.168.7.1     Up     0  1 SE  299872        3
```

```
192.168.1.1:192.168.7.1:3:mvpn:blue
Total 2 displayed, Up 2, Down 0

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

In this display from the PE1 router, notice that there are two ingress LSPs for the green VPN and two for the red VPN configured on this router. Verify that the state of each ingress LSP is up. Also notice that there is one egress LSP for each of the green and blue VPNs. Verify that the state of each egress LSP is up.

TIP: The LSP name displayed in the show mpls lsp p2mp command output can be used in the ping mpls rsvp <*lsp-name*> multipath command.

Configuring CE-PE BGP

Step-by-Step Procedure

1. On the PE routers, configure the BGP export policy. The BGP export policy is used to allow static routes and routes that originated from directly attached interfaces to be exported to BGP.

```
user@pe1# set policy-options policy-statement BGP-export term t1 from protocol
    direct
user@pe1# set policy-options policy-statement BGP-export term t1 then accept
user@pe1# set policy-options policy-statement BGP-export term t2 from protocol
    static
user@pe1# set policy-options policy-statement BGP-export term t2 then accept

user@pe2# set policy-options policy-statement BGP-export term t1 from protocol
    direct
user@pe2# set policy-options policy-statement BGP-export term t1 then accept
user@pe2# set policy-options policy-statement BGP-export term t2 from protocol
    static
user@pe2# set policy-options policy-statement BGP-export term t2 then accept

user@pe3# set policy-options policy-statement BGP-export term t1 from protocol
    direct
user@pe3# set policy-options policy-statement BGP-export term t1 then accept
user@pe3# set policy-options policy-statement BGP-export term t2 from protocol
    static
user@pe3# set policy-options policy-statement BGP-export term t2 then accept
```

   Use the show policy BGP-export command to verify that the BGP export policy is correctly configured.

2. On the PE routers, configure the CE to PE BGP session. Use the IP address of the SONET interface as the neighbor address. Specify the autonomous system number for the VPN network of the attached CE router.

```
user@pe1# set routing-instances green protocols bgp group PE-CE export
  BGP-export
user@pe1# set routing-instances green protocols bgp group PE-CE neighbor
  10.0.16.1 peer-as 65001

user@pe2# set routing-instances green protocols bgp group PE-CE export
  BGP-export
user@pe2# set routing-instances green protocols bgp group PE-CE neighbor
  10.0.24.2 peer-as 65009

user@pe3# set routing-instances blue protocols bgp group PE-CE export
  BGP-export
user@pe3# set routing-instances blue protocols bgp group PE-CE neighbor
  10.0.79.2 peer-as 65003
```

3. On the CE routers, configure the BGP local autonomous system number.

```
user@ce1# set routing-options autonomous-system 65001

user@ce2# set routing-options autonomous-system 65009

user@ce3# set routing-options autonomous-system 65003
```

4. On the CE routers, configure the BGP export policy. The BGP export policy is used to allow static routes and routes that originated from directly attached interfaces to be exported to BGP.

```
user@ce1# set policy-options policy-statement BGP-export term t1 from protocol
  direct
user@ce1# set policy-options policy-statement BGP-export term t1 then accept
user@ce1# set policy-options policy-statement BGP-export term t2 from protocol
  static
user@ce1# set policy-options policy-statement BGP-export term t2 then accept

user@ce2# set policy-options policy-statement BGP-export term t1 from protocol
  direct
user@ce2# set policy-options policy-statement BGP-export term t1 then accept
user@ce2# set policy-options policy-statement BGP-export term t2 from protocol
  static
user@ce2# set policy-options policy-statement BGP-export term t2 then accept

user@ce3# set policy-options policy-statement BGP-export term t1 from protocol
  direct
user@ce3# set policy-options policy-statement BGP-export term t1 then accept
user@ce3# set policy-options policy-statement BGP-export term t2 from protocol
  static
user@ce3# set policy-options policy-statement BGP-export term t2 then accept
```

Use the show policy BGP-export command to verify that the BGP export policy is correctly configured.

5. On the CE routers, configure the CE-to-PE BGP session. Use the IP address of the SONET interface as the neighbor address. Specify the autonomous system number of the core network. Apply the BGP export policy.

```
user@ce1# set protocols bgp group PE-CE export BGP-export
user@ce1# set protocols bgp group PE-CE neighbor 10.0.16.2 peer-as 65000

user@ce2# set protocols bgp group PE-CE export BGP-export
user@ce2# set protocols bgp group PE-CE neighbor 10.0.24.1 peer-as 65000

user@ce3# set protocols bgp group PE-CE export BGP-export
user@ce3# set protocols bgp group PE-CE neighbor 10.0.79.1 peer-as 65000
```

6. On the PE routers, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

7. On the PE routers, use the show bgp group pe-ce command to verify that the BGP neighbors form a peer session.

```
user@PE1> show bgp group pe-ce

Group Type: External                                Local AS: 65000
  Name: PE-CE           Index: 1                    Flags: <>
  Export: [ BGP-export ]
  Holdtime: 0
  Total peers: 1        Established: 1
  10.0.16.1+60500
  green.inet.0: 2/3/3/0
```

Verify that the peer state for the CE routers is Established and that the IP address configured on the peer SONET interface is shown as the peer.

Configuring PIM on the PE Routers

Step-by-Step Procedure

1. On the PE routers, enable an instance of PIM in each VPN. Configure the lo0.1, lo0.2, and customer-facing SONET and Fast Ethernet interfaces. Specify the mode as sparse.

```
user@pe1# set routing-instances green protocols pim interface lo0.1 mode
  sparse
```

```
user@pe1# set routing-instances green protocols pim interface so-0/0/3.0
  mode sparse
user@pe1# set routing-instances red protocols pim interface lo0.2 mode sparse
user@pe1# set routing-instances red protocols pim interface fe-0/1/0.0 mode
  sparse

user@pe2# set routing-instances green protocols pim interface lo0.1 mode
  sparse
user@pe2# set routing-instances green protocols pim interface so-0/0/1.0
  mode sparse

user@pe3# set routing-instances blue protocols pim interface lo0.1 mode
  sparse
user@pe3# set routing-instances blue protocols pim interface so-0/0/1.0 mode
  sparse
```

2. On the PE routers, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

3. On the PE routers, use the show pim interfaces instance green command and substitute the appropriate VRF instance name to verify that the PIM interfaces are up.

```
user@PE1> show pim interfaces instance green

Instance: PIM.green

Name               Stat Mode       IP V State NbrCnt JoinCnt DR address
lo0.1              Up   Sparse      4 2 DR         0       0 10.10.1.1
lsi.0              Up   SparseDense 4 2 P2P        0       0
pe-1/2/0.32769     Up   Sparse      4 2 P2P        0       0
so-0/0/3.0         Up   Sparse      4 2 P2P        1       2
vt-1/2/0.1         Up   SparseDense 4 2 P2P        0       0
lsi.0              Up   SparseDense 6 2 P2P        0       0
```

Also notice that the normal mode for the virtual tunnel interface and label-switched interface is SparseDense.

Configuring PIM on the CE Routers

Step-by-Step Procedure

1. On the CE routers, configure the customer-facing and core-facing interfaces for PIM. Specify the mode as sparse.

```
user@ce1# set protocols pim interface fe-1/3/0.0 mode sparse
user@ce1# set protocols pim interface so-0/0/3.0 mode sparse

user@ce2# set protocols pim interface fe-0/1/1.0 mode sparse
user@ce2# set protocols pim interface so-0/0/1.0 mode sparse

user@ce3# set protocols pim interface fe-0/1/0.0 mode sparse
user@ce3# set protocols pim interface so-0/0/1.0 mode sparse
```

   Use the show pim interfaces command to verify that the PIM interfaces have been configured to use sparse mode.

2. On the CE routers, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

3. On the CE routers, use the show pim interfaces command to verify that the PIM interface status is up.

```
user@CE1> show pim interfaces

Instance: PIM.master

Name               Stat Mode       IP V State NbrCnt JoinCnt DR address
fe-1/3/0.0         Up   Sparse      4 2 DR         0       0 10.10.12.1
pe-1/2/0.32769     Up   Sparse      4 2 P2P        0       0
so-0/0/3.0         Up   Sparse      4 2 P2P        1       1
```

Configuring the Rendezvous Points

Step-by-Step Procedure

1. Configure the PE1 router to be the rendezvous point for the red VPN instance of PIM. Specify the local lo0.2 address.

```
user@pe1# set routing-instances red protocols pim rp local address 10.2.1.1
```

2. Configure the PE2 router to be the rendezvous point for the green VPN instance of PIM. Specify the lo0.1 address of the PE2 router.

   user@pe2# **set routing-instances green protocols pim rp local address 10.10.22.2**

3. Configure the PE3 router to be the rendezvous point for the blue VPN instance of PIM. Specify the local lo0.1.

   user@pe2# **set routing-instances blue protocols pim rp local address 10.3.33.3**

4. On the PE1, CE1, and CE2 routers, configure the static rendezvous point for the green VPN instance of PIM. Specify the lo0.1 address of the PE2 router.

   user@pe1# **set routing-instances green protocols pim rp static address 10.10.22.2**

   user@ce1# **set protocols pim rp static address 10.10.22.2**

   user@ce2# **set protocols pim rp static address 10.10.22.2**

5. On the CE3 router, configure the static rendezvous point for the blue VPN instance of PIM. Specify the lo0.1 address of the PE3 router.

   user@ce3# **set protocols pim rp static address 10.3.33.3**

6. On the CE routers, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

7. On the PE routers, use the show pim rps instance <*instance-name*> command and substitute the appropriate VRF instance name to verify that the RPs have been correctly configured.

```
user@PE1> show pim rps instance <instance-name>

Instance: PIM.green
Address family INET
RP address              Type       Holdtime Timeout Groups Group prefixes
10.10.22.2               static             0    None      1 224.0.0.0/4

Address family INET6
```

   Verify that the correct IP address is shown as the RP.

8. On the CE routers, use the show pim rps command to verify that the RP has been correctly configured.

```
user@CE1> show pim rps

Instance: PIM.master
Address family INET
RP address              Type       Holdtime Timeout Groups Group prefixes
10.10.22.2               static             0    None      1 224.0.0.0/4

Address family INET6
```

Verify that the correct IP address is shown as the RP.

9. On the PE1 router, use the show route table green.mvpn.0 | find 1 command to verify that the type-1 routes have been received from the PE2 and PE3 routers.

```
user@CE1> show route table green.mvpn.0 | find 1

green.mvpn.0: 7 destinations, 9 routes (7 active, 1 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1:192.168.1.1:1:192.168.1.1/240
                   *[MVPN/70] 03:38:09, metric2 1
                      Indirect
1:192.168.1.1:2:192.168.1.1/240
                   *[MVPN/70] 03:38:05, metric2 1
                      Indirect
1:192.168.2.1:1:192.168.2.1/240
                   *[BGP/170] 03:12:18, localpref 100, from 192.168.2.1
                      AS path: I
                    > to 10.0.12.10 via ge-0/3/0.0
1:192.168.7.1:3:192.168.7.1/240
                   *[BGP/170] 03:12:18, localpref 100, from 192.168.7.1
                      AS path: I
                    > to 10.0.17.14 via fe-0/1/1.0
```

10. On the PE1 router, use the show route table green.mvpn.0 | find 5 command to verify that the type-5 routes have been received from the PE2 router.

```
user@PE1> show route table green.mvpn.0 | find 5

5:192.168.2.1:1:32:10.10.12.52:32:224.1.1.1/240
                   *[BGP/170] 03:12:18, localpref 100, from 192.168.2.1
                      AS path: I
                    > to 10.0.12.10 via ge-0/3/0.0
```

11. On the PE1 router, use the show route table green.mvpn.0 | find 7 command to verify that the type-7 routes have been received from the PE2 router.

```
user@PE1> show route table green.mvpn.0 | find 7

7:192.168.1.1:1:65000:32:10.10.12.52:32:224.1.1.1/240
                   *[MVPN/70] 03:22:47, metric2 1
```

```
  Multicast (IPv4)
[PIM/105] 03:34:18
  Multicast (IPv4)
[BGP/170] 03:12:18, localpref 100, from 192.168.2.1
  AS path: I
> to 10.0.12.10 via ge-0/3/0.0
```

12. On the PE1 router, use the show route advertising-protocol bgp 192.168.2.1 table green.mvpn.0 detail command to verify that the routes advertised by the PE2 router use the PMSI attribute set to RSVP-TE.

```
user@CE1> show route advertising-protocol bgp 192.168.2.1 table
green.mvpn.0 detail

green.mvpn.0: 7 destinations, 9 routes (7 active, 1 holddown, 0 hidden)
* 1:192.168.1.1:1:192.168.1.1/240 (1 entry, 1 announced)
 BGP group group-mvpn type Internal
     Route Distinguisher: 192.168.1.1:1
     Nexthop: Self
     Flags: Nexthop Change
     Localpref: 100
     AS path: [65000] I
     Communities: target:65000:1
     PMSI: Flags
0:RSVP-TE:label[0:0:0]:Session_13[192.168.1.1:0:56822:192.168.1.1]
```

Testing MVPN Extranets

Step-by-Step Procedure

1. Start the multicast receiver device connected to the CE2 router.
2. Start the multicast sender device connected to the CE1 router.
3. Verify that the receiver receives the multicast stream.
4. On the PE1 router, display the provider tunnel to multicast group mapping using the show mvpn c-multicast command.

```
user@PE1> show mvpn c-multicast

MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: green
  C-mcast IPv4 (S:G)            Ptnl                              St
  10.10.12.52/32:224.1.1.1/32   RSVP-TE P2MP:192.168.1.1, 56822,192.168.1.1
        RM
  0.0.0.0/0:239.255.255.250/32
MVPN instance:

Legend for provider tunnel
```

```
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: red
  C-mcast IPv4 (S:G)            Ptnl                              St
  10.10.12.52/32:224.1.1.1/32                                          DS

  0.0.0.0/0:224.1.1.1/32
```

5. On the PE2 router, use the show route table green.mvpn.0 | find 6 command to verify that the type-6 routes have been created as a result of receiving PIM join messages.

```
user@PE1> show route table green.mvpn.0 | find 6

6:192.168.2.1:1:65000:32:10.10.22.2:32:224.1.1.1/240
                   *[PIM/105] 04:01:23
                      Multicast (IPv4)
6:192.168.2.1:1:65000:32:10.10.22.2:32:239.255.255.250/240

                   *[PIM/105] 22:39:46
                      Multicast (IPv4)
```

NOTE: The multicast address 239.255.255.250 shown in the preceding step is not related to this example. This address is sent by some host machines.

6. Start the multicast receiver device connected to the CE3 router.
7. Verify that the receiver is receiving the multicast stream.
8. On the PE2 router, use the show route table green.mvpn.0 | find 6 command to verify that the type-6 routes have been created as a result of receiving PIM join messages from the multicast receiver device connected to the CE3 router.

```
user@PE1> show route table green.mvpn.0 | find 6

6:192.168.2.1:1:65000:32:10.10.22.2:32:239.255.255.250/240

                   *[PIM/105] 06:43:39
                      Multicast (IPv4)
```

9. Start the multicast receiver device directly connected to the PE1 router.
10. Verify that the receiver is receiving the multicast stream.
11. On the PE1 router, use the show route table green.mvpn.0 | find 6 command to verify that the type-6 routes have been created as a result of receiving PIM join messages from the directly connected multicast receiver device.

```
user@PE1> show route table green.mvpn.0 | find 6
```

```
6:192.168.1.1:2:65000:32:10.2.1.1:32:224.1.1.1/240
                    *[PIM/105] 00:02:32
                       Multicast (IPv4)
6:192.168.1.1:2:65000:32:10.2.1.1:32:239.255.255.250/240
                    *[PIM/105] 00:05:49
                       Multicast (IPv4)
```

NOTE: The multicast address 255.255.255.250 shown in the step above is not related to this example.

Results The configuration and verification parts of this example have been completed. The following section is for your reference.

The relevant sample configuration for the CE1 router follows.

Router CE1

```
interfaces {
   so-0/0/3 {
      unit 0 {
         description "to PE1 so-0/0/3.0";
         family inet {
            address 10.0.16.1/30;
         }
      }
   }
   fe-1/3/0 {
      unit 0 {
         family inet {
            address 10.10.12.1/24;
         }
      }
   }
   lo0 {
      unit 0 {
         description "CE1 Loopback";
         family inet {
            address 192.168.6.1/32 {
               primary;
            }
            address 127.0.0.1/32;
         }
      }
   }
}
routing-options {
   autonomous-system 65001;
   router-id 192.168.6.1;
   forwarding-table {
      export load-balance;
   }
}
protocols {
```

```
    bgp {
      group PE-CE {
        export BGP-export;
        neighbor 10.0.16.2 {
          peer-as 65000;
        }
      }
    }
    pim {
      rp {
        static {
          address 10.10.22.2;
        }
      }
      interface fe-1/3/0.0 {
        mode sparse;
      }
      interface so-0/0/3.0 {
        mode sparse;
      }
    }
  }
  policy-options {
    policy-statement BGP-export {
      term t1 {
        from protocol direct;
        then accept;
      }
      term t2 {
        from protocol static;
        then accept;
      }
    }
    policy-statement load-balance {
      then {
        load-balance per-packet;
      }
    }
  }
```

The relevant sample configuration for the PE1 router follows.

Router PE1

```
  interfaces {
    so-0/0/3 {
      unit 0 {
        description "to CE1 so-0/0/3.0";
        family inet {
          address 10.0.16.2/30;
        }
      }
    }
    fe-0/1/0 {
      unit 0 {
        description "to H2";
        family inet {
```

```
            address 10.2.11.2/30;
        }
    }
}
    fe-0/1/1 {
        unit 0 {
            description "to PE3 fe-0/1/1.0";
            family inet {
                address 10.0.17.13/30;
            }
            family mpls;
        }
    }
ge-0/3/0 {
    unit 0 {
        description "to PE2 ge-1/3/0.0";
        family inet {
            address 10.0.12.9/30;
        }
        family mpls;
    }
}
vt-1/2/0 {
    unit 1 {
        description "green VRF multicast vt";
        family inet;
    }
    unit 2 {
        description "red VRF unicast and multicast vt";
        family inet;
    }
    unit 3 {
        description "blue VRF multicast vt";
        family inet;
    }
}
lo0 {
    unit 0 {
        description "PE1 Loopback";
        family inet {
            address 192.168.1.1/32 {
                primary;
            }
            address 127.0.0.1/32;
        }
    }
    unit 1 {
        description "green VRF loopback";
        family inet {
            address 10.10.1.1/32;
        }
    }
    unit 2 {
        description "red VRF loopback";
        family inet {
            address 10.2.1.1/32;
```

```
        }
      }
    }
}
routing-options {
    autonomous-system 65000;
    router-id 192.168.1.1;
    forwarding-table {
        export load-balance;
    }
}
protocols {
    rsvp {
        interface ge-0/3/0.0;
        interface fe-0/1/1.0;
        interface lo0.0;
        interface fxp0.0 {
            disable;
        }
    }
    mpls {
        interface ge-0/3/0.0;
        interface fe-0/1/1.0;
        interface fxp0.0 {
            disable;
        }
    }
    bgp {
        group group-mvpn {
            type internal;
            local-address 192.168.1.1;
            family inet-vpn {
                unicast;
            }
            family inet-mvpn {
                signaling;
            }
            neighbor 192.168.2.1;
            neighbor 192.168.7.1;
        }
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface ge-0/3/0.0 {
                metric 100;
            }
            interface fe-0/1/1.0 {
                metric 100;
            }
            interface lo0.0 {
                passive;
            }
            interface fxp0.0 {
                disable;
            }
```

```
        }
    }
    ldp {
        deaggregate;
        interface ge-0/3/0.0;
        interface fe-0/1/1.0;
        interface fxp0.0 {
            disable;
        }
        interface lo0.0;
    }
}
policy-options {
    policy-statement BGP-export {
        term t1 {
            from protocol direct;
            then accept;
        }
        term t2 {
            from protocol static;
            then accept;
        }
    }
    policy-statement green-red-blue-import {
        term t1 {
            from community [ green-com red-com blue-com ];
            then accept;
        }
        term t2 {
            then reject;
        }
    }
    policy-statement load-balance {
        then {
            load-balance per-packet;
        }
    }
    community green-com members target:65000:1;
    community red-com members target:65000:2;
    community blue-com members target:65000:3;
}
routing-instances {
    green {
        instance-type vrf;
        interface so-0/0/3.0;
        interface vt-1/2/0.1 {
            multicast;
        }
        interface lo0.1;
        route-distinguisher 192.168.1.1:1;
        provider-tunnel {
            rsvp-te {
                label-switched-path-template {
                    default-template;
                }
            }
```

```
    }
    vrf-import green-red-blue-import;
    vrf-target export target:65000:1;
    vrf-table-label;
    routing-options {
        auto-export;
    }
    protocols {
        bgp {
            group PE-CE {
                export BGP-export;
                neighbor 10.0.16.1 {
                    peer-as 65001;
                }
            }
        }
        pim {
            rp {
                static {
                    address 10.10.22.2;
                }
            }
            interface so-0/0/3.0 {
                mode sparse;
            }
            interface lo0.1 {a
            mode sparse;
        }
    }
    mvpn;
}
red {
    instance-type vrf;
    interface fe-0/1/0.0;
    interface vt-1/2/0.2;
    interface lo0.2;
    route-distinguisher 192.168.1.1:2;
    provider-tunnel {
        rsvp-te {
            label-switched-path-template {
                default-template;
            }
        }
    }
    vrf-import green-red-blue-import;
    vrf-target export target:65000:2;
    routing-options {
        auto-export;
    }
    protocols {
        pim {
            rp {
                local {
                    address 10.2.1.1;
                }
            }
```

```
            interface fe-0/1/0.0 {
               mode sparse;
            }
            interface lo0.2 {
               mode sparse;
            }
         }
         mvpn;
      }
   }
}
```

The relevant sample configuration for the PE2 router follows.

Router PE2

```
interfaces {
   so-0/0/1 {
      unit 0 {
         description "to CE2 so-0/0/1:0.0";
         family inet {
            address 10.0.24.1/30;
         }
      }
   }
   fe-0/1/3 {
      unit 0 {
         description "to PE3 fe-0/1/3.0";
         family inet {
            address 10.0.27.13/30;
         }
         family mpls;
      }
      vt-1/2/0 {
         unit 1 {
            description "green VRF unicast and multicast vt";
            family inet;
         }
         unit 3 {
            description "blue VRF unicast and multicast vt";
            family inet;
         }
      }
   }
   ge-1/3/0 {
      unit 0 {
         description "to PE1 ge-0/3/0.0";
         family inet {
            address 10.0.12.10/30;
         }
         family mpls;
      }
   }
   lo0 {
      unit 0 {
         description "PE2 Loopback";
         family inet {
```

```
            address 192.168.2.1/32 {
               primary;
            }
            address 127.0.0.1/32;
         }
      }
      unit 1 {
         description "green VRF loopback";
         family inet {
            address 10.10.22.2/32;
         }
      }
   }
   routing-options {
      router-id 192.168.2.1;
      autonomous-system 65000;
      forwarding-table {
         export load-balance;
      }
   }
   protocols {
      rsvp {
         interface fe-0/1/3.0;
         interface ge-1/3/0.0;
         interface lo0.0;
         interface fxp0.0 {
            disable;
         }
      }
      mpls {
         interface fe-0/1/3.0;
         interface ge-1/3/0.0;
         interface fxp0.0 {
            disable;
         }
      }
      bgp {
         group group-mvpn {
            type internal;
            local-address 192.168.2.1;
            family inet-vpn {
               unicast;
            }
            family inet-mvpn {
               signaling;
            }
            neighbor 192.168.1.1;
            neighbor 192.168.7.1;
         }
      }
      ospf {
         traffic-engineering;
         area 0.0.0.0 {
            interface fe-0/1/3.0 {
               metric 100;
            }
```

```
            interface ge-1/3/0.0 {
                metric 100;
            }
            interface lo0.0 {
                passive;
            }
            interface fxp0.0 {
                disable;
            }
        }
    }
    ldp {
        deaggregate;
        interface fe-0/1/3.0;
        interface ge-1/3/0.0;
        interface fxp0.0 {
            disable;
        }
        interface lo0.0;
    }
}
policy-options {
    policy-statement BGP-export {
        term t1 {
            from protocol direct;
            then accept;
        }
        term t2 {
            from protocol static;
            then accept;
        }
    }
    policy-statement green-red-blue-import {
        term t1 {
            from community [ green-com red-com blue-com ];
            then accept;
        }
        term t2 {
            then reject;
        }
    }
    policy-statement load-balance {
        then {
            load-balance per-packet;
        }
    }
    community green-com members target:65000:1;
    community red-com members target:65000:2;
    community blue-com members target:65000:3;
}
routing-instances {
    green {
        instance-type vrf;
        interface so-0/0/1.0;
        interface vt-1/2/0.1;
        interface lo0.1;
```

```
            route-distinguisher 192.168.2.1:1;
            provider-tunnel {
               rsvp-te {
                  label-switched-path-template {
                     default-template;
                  }
               }
            }
            vrf-import green-red-blue-import;
            vrf-target export target:65000:1;
            routing-options {
               auto-export;
            }
            protocols {
               bgp {
                  group PE-CE {
                     export BGP-export;
                     neighbor 10.0.24.2 {
                        peer-as 65009;
                     }
                  }
               }
               pim {
                  rp {
                     local {
                        address 10.10.22.2;
                     }
                  }
                  interface so-0/0/1.0 {
                     mode sparse;
                  }
                  interface lo0.1 {
                     mode sparse;
                  }
               }
               mvpn;
            }
         }
      }
   }
```

The relevant sample configuration for the CE2 router follows.

Router CE2

```
interfaces {
   fe-0/1/1 {
      unit 0 {
         description "to H4";
         family inet {
            address 10.10.11.2/24;
         }
      }
   }
   so-0/0/1 {
      unit 0 {
         description "to PE2 so-0/0/1";
```

```
            family inet {
                address 10.0.24.2/30;
            }
        }
    }
    lo0 {
        unit 0 {
            description "CE2 Loopback";
            family inet {
                address 192.168.4.1/32 {
                    primary;
                }
                address 127.0.0.1/32;
            }
        }
    }
}
routing-options {
    router-id 192.168.4.1;
    autonomous-system 65009;
    forwarding-table {
        export load-balance;
    }
}
protocols {
    bgp {
        group PE-CE {
            export BGP-export;
            neighbor 10.0.24.1 {
                peer-as 65000;
            }
        }
    }
    pim {
        rp {
            static {
                address 10.10.22.2;
            }
        }
        interface so-0/0/1.0 {
            mode sparse;
        }
        interface fe-0/1/1.0 {
            mode sparse;
        }
    }
}
policy-options {
    policy-statement BGP-export {
        term t1 {
            from protocol direct;
            then accept;
        }
        term t2 {
            from protocol static;
            then accept;
```

```
    }
  }
  policy-statement load-balance {
    then {
      load-balance per-packet;
    }
  }
}
```

The relevant sample configuration for the PE3 router follows.

Router PE3

```
interfaces {
  so-0/0/1 {
    unit 0 {
      description "to CE3 so-0/0/1.0";
      family inet {
        address 10.0.79.1/30;
      }
    }
  }
  fe-0/1/1 {
    unit 0 {
      description "to PE1 fe-0/1/1.0";
      family inet {
        address 10.0.17.14/30;
      }
      family mpls;
    }
  }
  fe-0/1/3 {
    unit 0 {
      description "to PE2 fe-0/1/3.0";
      family inet {
        address 10.0.27.14/30;
      }
      family mpls;
    }
  }
  vt-1/2/0 {
    unit 3 {
      description "blue VRF unicast and multicast vt";
      family inet;
    }
  }
  lo0 {
    unit 0 {
      description "PE3 Loopback";
      family inet {
        address 192.168.7.1/32 {
          primary;
        }
        address 127.0.0.1/32;
      }
    }
    unit 1 {
```

```
            description "blue VRF loopback";
            family inet {
               address 10.3.33.3/32;
            }
         }
      }
   }
   routing-options {
      router-id 192.168.7.1;
      autonomous-system 65000;
      forwarding-table {
         export load-balance;
      }
   }
   protocols {
      rsvp {
         interface fe-0/1/3.0;
         interface fe-0/1/1.0;
         interface lo0.0;
         interface fxp0.0 {
            disable;
         }
      }
      mpls {
         interface fe-0/1/3.0;
         interface fe-0/1/1.0;
         interface fxp0.0 {
            disable;
         }
      }
      bgp {
         group group-mvpn {
            type internal;
            local-address 192.168.7.1;
            family inet-vpn {
               unicast;
            }
            family inet-mvpn {
               signaling;
            }
            neighbor 192.168.1.1;
            neighbor 192.168.2.1;
         }
      }
      ospf {
         traffic-engineering;
         area 0.0.0.0 {
            interface fe-0/1/3.0 {
               metric 100;
            }
            interface fe-0/1/1.0 {
               metric 100;
            }
            interface lo0.0 {
               passive;
            }
```

```
        interface fxp0.0 {
            disable;
        }
    }
  }
  ldp {
    deaggregate;
    interface fe-0/1/3.0;
    interface fe-0/1/1.0;
    interface fxp0.0 {
      disable;
    }
    interface lo0.0;
  }
}
policy-options {
  policy-statement BGP-export {
    term t1 {
      from protocol direct;
      then accept;
    }
    term t2 {
      from protocol static;
      then accept;
    }
  }
  policy-statement green-red-blue-import {
    term t1 {
      from community [ green-com red-com blue-com ];
      then accept;
    }
    term t2 {
      then reject;
    }
  }
  policy-statement load-balance {
    then {
      load-balance per-packet;
    }
  }
  community green-com members target:65000:1;
  community red-com members target:65000:2;
  community blue-com members target:65000:3;
}
routing-instances {
  blue {
    instance-type vrf;
    interface vt-1/2/0.3;
    interface so-0/0/1.0;
    interface lo0.1;
    route-distinguisher 192.168.7.1:3;
    provider-tunnel {
      rsvp-te {
        label-switched-path-template {
          default-template;
        }
```

```
            }
        }
        vrf-import green-red-blue-import;
        vrf-target target:65000:3;
        routing-options {
            auto-export;
        }
        protocols {
            bgp {
                group PE-CE {
                    export BGP-export;
                    neighbor 10.0.79.2 {
                        peer-as 65003;
                    }
                }
            }
            pim {
                rp {
                    local {
                        address 10.3.33.3;
                    }
                }
                interface so-0/0/1.0 {
                    mode sparse;
                }
                interface lo0.1 {
                    mode sparse;
                }
            }
            mvpn ;
        }
    }
}
```

The relevant sample configuration for the CE3 router follows.

Router CE3

```
interfaces {
    so-0/0/1 {
        unit 0 {
            description "to PE3";
            family inet {
                address 10.0.79.2/30;
            }
        }
    }
    fe-0/1/0 {
        unit 0 {
            description "to H3";
            family inet {
                address 10.3.11.3/24;
            }
        }
    }
    lo0 {
        unit 0 {
```

```
        description "CE3 loopback";
        family inet {
          address 192.168.9.1/32 {
            primary;
          }
          address 127.0.0.1/32;
        }
      }
    }
}
routing-options {
    router-id 192.168.9.1;
    autonomous-system 65003;
    forwarding-table {
      export load-balance;
    }
}
protocols {
    bgp {
      group PE-CE {
        export BGP-export;
        neighbor 10.0.79.1 {
          peer-as 65000;
        }
      }
    }
    pim {
      rp {
        static {
          address 10.3.33.3;
        }
      }
      interface so-0/0/1.0 {
        mode sparse;
      }
      interface fe-0/1/0.0 {
        mode sparse;
      }
    }
}
policy-options {
    policy-statement BGP-export {
      term t1 {
        from protocol direct;
        then accept;
      }
      term t2 {
        from protocol static;
        then accept;
      }
    }
    policy-statement load-balance {
      then {
        load-balance per-packet;
      }
    }
```

}

Dual PIM Draft-Rosen Multicast VPN Operation

The operation of draft-rosen multicast within a Layer 3 VPN domain with provider tunnels operating in any-source (ASM) multicast mode occurs in multiple stages, which are shown in Figure 34 on page 440 and described on the following pages.

Figure 34: Multicast Over Layer 3 VPN Operation

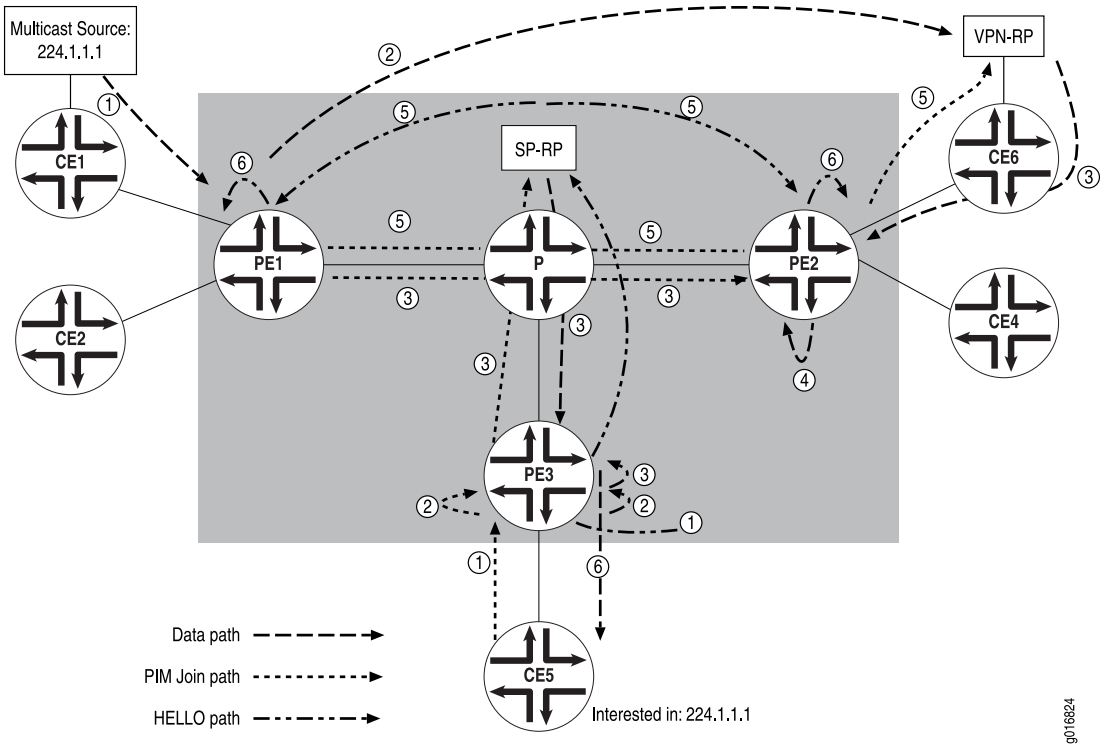


Figure 34 on page 440 shows the various stages that multicast packets pass through in a Layer 3 VPN environment.

- Stage 1: PIM hello messages

1. PIM is configured as part of a VPN routing instance and the configuration is committed. For Juniper Networks M Series Multiservice Edge Routers and Juniper Networks T Series Core Routers, a virtual multicast tunnel interface (mt-*fpc/pic/port.abcde*) is created if a Tunnel Services Physical Interface Card (PIC) is installed on the router. On Juniper Networks MX Series Ethernet Services Routers, you can create a virtual multicast tunnel interface by including the tunnel-services statement at the [edit chassis fpc *slot-number* pic *number*] hierarchy level. For more information about configuring tunnel interfaces on MX Series routers, see the *JUNOS System Basics Configuration Guide*. The virtual multicast tunnel interface is used to communicate between the PIM instance within the VRF and the master PIM instance.
2. A PIM hello message is sent from the VRF across the mt interface. When this happens, a GRE header is prepended to the PIM hello message with fields containing the VPN group address and the loopback address of the PE router.
3. A PIM register header is prepended to the hello message as the packet is looped through the pe (PIM encapsulation) interface. This header contains the destination address of the SP-RP and the loopback address of the PE router.
4. The packet is sent to the SP-RP.
5. The SP-RP de-encapsulates the top header off the packet as it travels through the pd (PIM de-encapsulation) interface and sends the remaining GRE encapsulated hello message to all of the PE routers.
6. The master PIM instance on the PE router handles the GRE encapsulated packet. Because the VPN group address is contained in the packet, the master PIM instance de-encapsulates the packet and sends the hello message over the mt interface to reach the desired VPN group address within the VRF.

- Stage 2: PIM join message

1. Router CE5 is interested in receiving from multicast source 224.1.1.1, so a PIM Join message is sent from Router CE5 to Router PE3.
2. The PIM Join message is sent through the mt interface and a GRE header is prepended to it. The GRE header contains the VPN group ID and the loopback address of Router PE3.
3. The GRE encapsulated Join message is sent to other PE routers.
4. Router PE2 receives the packet. Because the VPN C-RP is behind Router PE2, Router PE2 sends the packet through the mt interface, which strips off the GRE header.
5. The PIM Join message is now sent to the VPN C-RP.

- Stage 3: Multicast forwarding

1. The source behind Router CE1 is sending to group 224.1.1.1. The designated router (DR) behind the CE router encapsulates this packet into a PIM register.
2. Because the packet already has the PIM register header, it is forwarded to the VPN C-RP by unicast routing over the Layer 3 VPN.

3. The VPN C-RP de-encapsulates the data packet and sends it out the downstream interfaces (which include the return path interface leading to Router PE3). Router P also forwards the packet to Router PE3.
4. The data packet is sent through the mt interface on Router PE2. In the process, the GRE header is prepended to the packet.
5. The packet is sent to the PE routers with GRE header intact.
6. The "interested" PE routers strip the GRE header off the packet and forward it to the CE routers that requested the PIM join. If there are no PIM-join messages for this group at this site, the PE router drops the packet.

When PIM is configured within a routing instance, two mt interfaces are created:

- mt-*xxxxx* range is 32768 through 49151 for mt-encap
- mt-*yyyyy* range is 49152 through 65535 for mt-decap

PIM is run only on the mt-encap interface. The mt-decap interface is used to populate downstream interface information.

Configuring Draft-Rosen Multicast VPNs

Draft-rosen multicast VPN with provider tunnels operating in any-source multicast (ASM) mode is a legacy feature. We recommend that you implement multiprotocol BGP-based multicast VPNs for multicast VPNs. Use this section to maintain your already existing draft-rosen multicast VPNs with provider tunnels operating in ASM mode. To implement this type of draft-rosen IPv4 multicast for a Layer 3 VPN, configure the following:

- Configuring BGP, MPLS, RSVP, and an IGP on the PE and Core Routers for Draft Rosen VPNs on page 442
- Creating a Unique Logical Loopback Interface for the Routing Instance for Draft Rosen VPNs on page 443
- Configuring the Master PIM Instance on the PE Router in the Service Provider Network on page 443
- Configuring PIM and the VPN Group Address in a Routing Instance on page 444
- Option: Configuring PIM Sparse Mode Graceful Restart for a Layer 3 VPN on page 445
- Option: Configuring Multicast Distribution Trees for Data on page 445
- Option: Configuring MSDP Within a Layer 3 VPN on page 446

Configuring BGP, MPLS, RSVP, and an IGP on the PE and Core Routers for Draft Rosen VPNs

To send multicast traffic across a Layer 3 VPN, you must configure network protocols to handle *intradomain routing* (an interior gateway protocol [IGP], such as Open Shortest Path First [OSPF] or Intermediate System-to-Intermediate System [IS-IS]), *interdomain routing* (Border Gateway Protocol [BGP]), *label switching* (Multiprotocol

Label Switching [MPLS]), and *path signaling* (Resource Reservation Protocol [RSVP]). For more information about these protocols and examples of how to configure these protocols to support a Layer 3 VPN, see the *JUNOS VPNs Configuration Guide*.

NOTE: In multicast Layer 3 VPNs, the multicast PE routers must use the primary loopback address (or router ID) for sessions with their internal BGP peers. If the PE routers use a route reflector with next-hop self-configured, Layer 3 multicast over VPN does not work because PIM cannot transmit upstream interface information for multicast sources behind remote PE routers into the network core. Multicast Layer 3 VPNs require the BGP next-hop address of the VPN route to match the BGP next-hop address of the loopback VRF instance address.

Creating a Unique Logical Loopback Interface for the Routing Instance for Draft Rosen VPNs

To facilitate the PIM protocol within a Layer 3 VPN, configure a unique loopback interface for the routing instance at the [edit interfaces lo0 unit] hierarchy level:

```
[edit interfaces]
lo0 {
    unit 1 {
        family inet {
            address  ip-address;
        }
    }
}
```

Configuring the Master PIM Instance on the PE Router in the Service Provider Network

To configure the master PIM instance that communicates with other PIM neighbors and the SP-RP within the service provider network, include the pim statement at the [edit protocols] hierarchy level. The example shown enables PIM sparse mode.

```
[edit protocols]
pim {
    rp {
        static {
            address ip-address;
        }
    }
    interface all {
        mode sparse;
        version 2;
    }
}
```

Configuring PIM and the VPN Group Address in a Routing Instance

The configuration syntax for PIM in a Layer 3 instance is available at the [edit routing-instances protocols pim] hierarchy level. It is similar to the global PIM configuration syntax found at the [edit protocols pim] hierarchy level.

In JUNOS Release 5.3 and later, you can include the vpn-group-address statement at the [edit routing-instances *instance-name* protocols pim] hierarchy level. You include this statement within the routing instance and specify the multicast group address for a particular VPN. Only one vpn-group-address statement can be configured per VPN, and this address should be unique on a per-VPN basis. To review how the VPN group address is used within GRE packet headers, see Stage 2 in "Dual PIM Multicast VPNs: Draft Rosen" on page 341.

Keep in mind that each PE router contains two entries of PIM: one for the master instance of PIM that connects through the service provider network and a second for the routing instance that connects to the CE router. The RP listed within the routing instance is the VPN C-RP, whereas the RP in the master PIM instance is an SP-RP. The following sample configuration shows a PE router with PIM enabled for sparse-dense mode in the VPN instance.

```
[edit]
routing-instances {
   instance-name {
      .....
      protocols {
         .....
         pim {
            vpn-group-address group-address;
            rp {
               static {
                  address ip-address;
               }
            }
            interface interface-name {
               mode sparse-dense;
               version 2;
            }
            interface lo0.1 {
               mode sparse-dense;
               version 2;
            }
         }
      }
   }
}
```

NOTE: You cannot configure PIM within a nonforwarding instance. If you try to do so, the router displays a commit check error and does not complete the configuration commit process.

NOTE: In JUNOS Release 5.5 and later, you can configure PIM dense mode with the dense statement at the [edit routing-instances pim mode] hierarchy level. Sparse mode is available at this same hierarchy level in JUNOS Release 5.3 and later.

Option: Configuring PIM Sparse Mode Graceful Restart for a Layer 3 VPN

Graceful restart permits a routing platform to continue forwarding multicast traffic to neighbors while the routing protocol process restarts. To enable graceful restart for PIM sparse mode in a Layer 3 VPN, include the graceful-restart statement at both the [edit routing-options] and [edit routing-instances *instance-name* routing-options] hierarchy levels. To disable graceful restart in a Layer 3 VPN, include the disable statement at the [edit routing-instances *instance-name* protocols pim graceful-restart] hierarchy level.

```
[edit]
routing-options {
   graceful-restart {
      disable;
      restart-duration seconds;
   }
}
routing-instances {
   instance-name {
      .....
      protocols {
         pim {
            graceful-restart {
               disable;
               restart-duration seconds;
            }
         }
      }
      routing-options {
         graceful-restart {
            disable;
            restart-duration seconds;
         }
      }
   }
}
```

For more information about PIM sparse mode graceful restart in a Layer 3 VPN, see the *JUNOS High Availability Configuration Guide* or the *JUNOS Multicast Protocols Configuration Guide*.

Option: Configuring Multicast Distribution Trees for Data

By using date multicast distribution trees (MDTs) in a Layer 3 VPN, you can prevent multicast packets from being flooded unnecessarily to specified provider edge (PE) routers within a VPN group. This option is primarily useful for PE routers in your

Layer 3 VPN multicast network that have no receivers for the multicast traffic from a particular source.

When a PE router directly connected to the multicast source receives Layer 3 VPN multicast traffic exceeding a configured threshold, a new data MDT tunnel is established between the PE router connected to the site where the multicast source is and its remote PE router neighbors. Neighbors that do not have receivers for the multicast traffic ignore the new tunnel. Conversely, neighbors that do have receivers for the multicast traffic link to the data MDT tunnel, which is created when the site exceeds a traffic rate threshold. If the multicast traffic level drops back below the threshold, the data MDT is torn down automatically and traffic flows back across the default Layer 3 VPN PIM tunnel.

To specify when the PE router directly connected to the multicast source should create a new data MDT, you must configure the maximum threshold value by including the rate statement at the [edit routing-instances *instance-name* protocols pim mdt threshold group *group-address* source *source-address*] hierarchy level. The data rate is specified in kilobits per second (Kbps). To specify the maximum number of data MDTs that can be created for a single routing instance, include the tunnel-limit statement at the [edit routing-instances *instance-name* protocols pim mdt] hierarchy level. To specify the multicast group IP address range used when a new data MDT needs to be initiated on the PE router, include the group-range statement at the [edit routing-instances *instance-name* protocols pim mdt] hierarchy level.

```
[edit routing-instances instance-name protocols pim]
mdt {
   group-range multicast-prefix;
   threshold {
      group  group-address  {
         source  source-address  {
            rate  threshold-rate;
         }
      }
   }
   tunnel-limit  limit;
}
```

NOTE: Because MDTs applies to VPNs and VRF routing instances, you cannot configure MDT statements in the master routing instance. If you configure MDTs in the master routing instance, the configuration commit operation will fail.

For more information about MDT, see the *JUNOS Multicast Protocols Configuration Guide*.

Option: Configuring MSDP Within a Layer 3 VPN

MSDP, defined in RFC 3618, allows a PIM-enabled network to connect multicast routing domains. It typically runs on the same router as a PIM sparse-mode rendezvous point (RP). Each MSDP router establishes adjacencies with internal and external MSDP peers similar to adjacency establishment for BGP peers. MSDP peer

routers inform each other about active sources within the domain. When the peers detect active sources, they send explicit join messages to the active source.

You can configure MSDP in the master instance of a routing platform, or in the following types of routing instances:

- Forwarding
- No forwarding
- Virtual router
- VPLS
- VRF

To configure MSDP in a Layer 3 VPN, include the msdp statement at the [edit routing-instances *instance-name* protocols] hierarchy level and specify local and peer addresses. You must also configure PIM sparse mode in the routing instance and specify a rendezvous point.

```
[edit routing-instances instance-name protocols]
pim {
    rp {
        local {
            address ip-address;
        }
    }
    interface interface-name;
}
msdp {
    local-address local-ip-address;
    peer peer-ip-address;
}
```

To view information about the operation of MSDP within a Layer 3 VPN instance, issue the show msdp instance, show msdp statistics instance, show msdp source instance, and show msdp source-active instance commands. For more information about MSDP, see the *JUNOS Multicast Protocols Configuration Guide*.

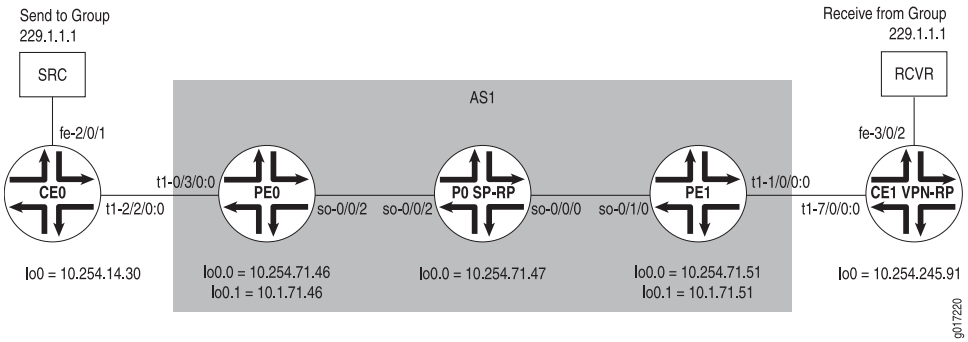
Draft-Rosen Multicast VPNs Examples

This section contains the following configuration examples and commands you can issue to verify your draft-rosen IPv4 multicast configuration:

- Example: Basic IPv4 Multicast over a Layer 3 VPN Configuration on page 448
- Example: IPv4 Multicast with Interprovider VPNs Configuration on page 461

Example: Basic IPv4 Multicast over a Layer 3 VPN Configuration

Figure 35: Basic IPv4 Multicast over a Layer 3 VPN Topology Diagram



In Figure 35 on page 448, the multicast source sends to group 229.1.1.1, and the receiver listens to the same group address. The VPN C-RP is located at Router CE1, whereas the SP-RP is located at Router P0. The routing instances are named VPN-A on both routers PE0 and PE1.

Router CE0

```
[edit]
protocols {
   pim {
      rp {
         dense-groups {
            229.0.0.0/8;
         }
         static {
            address 10.254.245.91;
         }
      }
      interface all {
         mode sparse-dense;
         version 2;
      }
      interface fxp0.0 {
         disable;
      }
   }
}
```

In this example, the interface all statement is configured. If the topology requires only a few interfaces to be configured for PIM, then loopback interface lo0 must also be one of the configured interfaces.

Router PE0

```
[edit]
protocols {
   pim {
      rp {
         static {
            address 10.254.71.47;
         }
```

```
        }
        interface all {
            mode sparse;
            version 2;
        }
        interface fxp0.0 {
            disable;
        }
    }
}
```

Router PE0 also requires a standard VPN configuration, along with the PIM instance configuration. The vpn-group-address command is the only new PIM statement with PIM used exclusively with a routing instance multicast configuration.

Router PE0

```
[edit]
routing-instances {
    VPN-A {
        instance-type vrf;
        interface t1-0/3/0:0.0;
        interface lo0.1
        route-distinguisher 10.254.71.46:100;
        vrf-import VPNA-import;
        vrf-export VPNA-export;
        protocols {
            ospf {
                export bgp-to-ospf;
                area 0.0.0.0 {
                    interface t1-0/3/0:0.0;
                    interface lo0.1;
                }
            }
            pim {
                dense-groups {
                    229.0.0.0/8;
                }
                vpn-group-address 239.1.1.1;
                rp {
                    static {
                        address 10.254.245.91;
                    }
                }
                interface t1-0/3/0:0.0 {
                    mode sparse-dense;
                    version 2;
                }
                interface lo0.1 {
                    mode sparse-dense;
                    version 2;
                }
            }
        }
    }
}
```

Router P0

```
[edit]
protocols {
    pim {
        rp {
            local {
                address 10.254.71.47;
            }
        }
        interface all {
            mode sparse;
            version 2;
        }
        interface fxp0.0 {
            disable;
        }
    }
}
```

Again, if the configuration calls for specific interfaces to be configured for PIM, loopback interface lo0 must be included as one of the interfaces running PIM.

Router PE1

```
[edit]
protocols {
    pim {
        rp {
            static {
                address 10.254.71.47;
            }
        }
        interface all {
            mode sparse;
            version 2;
        }
        interface fxp0.0 {
            disable;
        }
    }
}
routing-instances {
    VPN-A {
        instance-type vrf;
        interface t1-1/0/0:0.0;
        interface lo0.1;
        route-distinguisher 10.254.71.51:100;
        vrf-import VPNA-import;
        vrf-export VPNA-export;
        protocols {
            ospf {
                export bgp-to-ospf;
                area 0.0.0.0 {
                    interface t1-1/0/0:0.0;
                    interface lo0.1;
                }
            }
            pim {
```

```
            dense-groups {
               229.0.0.0/8;
            }
            vpn-group-address 239.1.1.1;
            rp {
               static {
                  address 10.254.245.91;
               }
            }
            interface t1-1/0/0:0.0 {
               mode sparse-dense;
               version 2;
            }
            interface lo0.1 {
               mode sparse-dense;
               version 2;
            }
         }
      }
   }
}
```

Router CE1

```
[edit]
protocols {
   pim {
      dense-groups {
         229.0.0.0/8;
      }
      rp {
         local {
            address 10.254.245.91;
         }
      }
      interface all {
         mode sparse-dense;
         version 2;
      }
      interface fxp0.0 {
         disable;
      }
   }
}
```

Verifying Your Work

To verify correct operation of basic IPv4 multicast over a Layer 3 VPN, use the following commands:

- show pim
- show pim rps
- show pim rps instance *instance-name*
- show pim join

- show pim join extensive
- show pim join extensive instance *instance-name*
- show multicast route extensive
- show multicast next-hops
- show interfaces mt-*fpc/pic/port* extensive

The following sections show the output of these commands used with the configuration example:

- RP Information on page 452
- PIM Information Before Multicast Transmission on page 452
- Successful PIM Join Verification on page 454

RP Information

You can view PIM information for the master instance with the show pim command. You can see information on the PIM routing instance with the show pim (rps | join extensive) instance *instance-name* command. Output verifying the SP-RP (10.254.71.47) as well as the VPN C-RP (10.254.245.91) follows.

```
user@PE0> show pim rps
Instance: PIM.master
Family: INET
RP address      Type      Holdtime Timeout Active groups Group prefixes
10.254.71.47    static           0    None             1 224.0.0.0/4
Family: INET6
RP address      Type      Holdtime Timeout Active groups Group prefixes
user@PE0>  show pim rps instance VPN-A
Instance: PIM.VPN-A
Family: INET
RP address      Type      Holdtime Timeout Active groups Group prefixes
10.254.245.91   static           0    None             0 224.0.0.0/4
Family: INET6
RP address      Type      Holdtime Timeout Active groups Group prefixes
```

PIM Information Before Multicast Transmission

With the configuration properly set, the backbone PIM sessions should be established before any traffic is forwarded. In the output below, the routers were configured, but the traffic source was not transmitting and the receiver was not requesting to be part of a group. Notice that there is no PIM join information for the routing instances yet.

Router PE0

```
user@PE0> show pim join extensive
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.47
    Flags: sparse,rptree,wildcard
    Upstream interface: so-0/0/2.0
    Upstream State: Join to RP
    Downstream Neighbors:
```

```
        Interface: mt-1/1/0.32769
            0.0.0.0 State: Join   Flags: SRW  Timeout: Infinity
Group: 239.1.1.1
    Source: 10.254.71.46
    Flags: sparse
    Upstream interface: local
    Upstream State: Local Source, Prune to RP
    Keepalive timeout: 166
    Downstream Neighbors:
        Interface: so-0/0/2.0
            192.168.296.70 State: Join   Flags: S    Timeout: 204
Group: 239.1.1.1
    Source: 10.254.71.51
    Flags: sparse,spt-pending
    Upstream interface: so-0/0/2.0
    Upstream State: Join to Source
    Keepalive timeout: 166
    Downstream Neighbors:
        Interface: mt-1/1/0.32769
            0.0.0.0 State: Join   Flags: S    Timeout: Infinity
user@PE0>  show pim join extensive instance VPN-A
Instance: PIM.VPN-A Family: INET
```

Router P0

```
user@P0> show pim join extensive
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.47
    Flags: sparse,rptree,wildcard
    Upstream interface: local
    Upstream State: Local RP
    Downstream Neighbors:
        Interface: so-0/0/0.0
            192.168.296.34 State: Join   Flags: SRW  Timeout: 186
        Interface: so-0/0/2.0
            192.168.296.69 State: Join   Flags: SRW  Timeout: 198
Group: 239.1.1.1
    Source: 10.254.71.46
    Flags: sparse,spt
    Upstream interface: so-0/0/2.0
    Upstream State: Local RP, Join to Source
    Keepalive timeout: 170
    Downstream Neighbors:
        Interface: so-0/0/0.0
            192.168.296.34 State: Join   Flags: S    Timeout: 186
        Interface: so-0/0/2.0             (pruned)
            192.168.296.69 State: Prune  Flags: SR   Timeout: 198
Group: 239.1.1.1
    Source: 10.254.71.51
    Flags: sparse,spt
    Upstream interface: so-0/0/0.0
    Upstream State: Local RP, Join to Source
    Keepalive timeout: 170
    Downstream Neighbors:
        Interface: so-0/0/0.0             (pruned)
            192.168.296.34 State: Prune  Flags: SR   Timeout: 186
        Interface: so-0/0/2.0
            192.168.296.69 State: Join   Flags: S    Timeout: 198
```

Router PE1

```
user@PE1> show pim join extensive
```

```
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.47
    Flags: sparse,rptree,wildcard
    Upstream interface: so-0/1/0.0
    Upstream State: Join to RP
    Downstream Neighbors:
        Interface: mt-1/1/0.32769
            0.0.0.0 State: Join   Flags: SRW  Timeout: Infinity
Group: 239.1.1.1
    Source: 10.254.71.46
    Flags: sparse,spt-pending
    Upstream interface: so-0/1/0.0
    Upstream State: Join to Source
    Keepalive timeout: 180
    Downstream Neighbors:
        Interface: mt-1/1/0.32769
            0.0.0.0 State: Join   Flags: S    Timeout: Infinity
Group: 239.1.1.1
    Source: 10.254.71.51
    Flags: sparse
    Upstream interface: local
    Upstream State: Local Source, Prune to RP
    Keepalive timeout: 180
    Downstream Neighbors:
        Interface: so-0/1/0.0
            192.168.296.33 State: Join   Flags: S    Timeout: 168
```

Successful PIM Join Verification

In the remaining output for this example, the show pim join output shows group participation. Also displayed is the output from the show multicast route extensive and show multicast next-hop commands. The join output for PIM within a VPN will reference the group 229.1.1.1, while the service provider side of the network will reference the join information for group 239.1.1.1 (which is the VPN group ID). In the show multicast route extensive output, you can view the group, sender, and upstream interface toward the sender.

Router CE0

```
user@CE0> show pim join
Instance: PIM.master Family: INET
Group: 229.1.1.1
    Source: 192.168.295.34
    Flags: dense
    Upstream interface: fe-2/0/1.0
Instance: PIM.master Family: INET6

user@CE0> show multicast route extensive
Family: INET
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout
229.1.1.1       192.168.295.34 /32 A   F  120   8010       0          360
    Upstream interface: fe-2/0/1.0
    Session name: Unknown
    Forwarding rate: 1 kbps (10 pps)
Family: INET6
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout

user@CE0> show multicast next-hops
```

```
Family: INET
ID      Refcount  KRefcount Downstream interface
120            2          1 t1-2/2/0:0.0
```

Router PE0

```
user@PE0> show pim join extensive
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.47
    Flags: sparse,rptree,wildcard
    Upstream interface: so-0/0/2.0
    Upstream State: Join to RP
    Downstream Neighbors:
        Interface: mt-1/1/0.32769
            10.1.71.46 State: Join   Flags: SRW  Timeout: Infinity
Group: 239.1.1.1
    Source: 10.254.71.46
    Flags: sparse
    Upstream interface: local
    Upstream State: Local Source, Prune to RP
    Keepalive timeout: 188
    Downstream Neighbors:
        Interface: so-0/0/2.0
            192.168.296.70 State: Join   Flags: S    Timeout: 180
Instance: PIM.master Family: INET6

user@PE0> show interfaces mt-1/1/0 extensive
Physical interface: mt-1/1/0, Enabled, Physical link is Up
  Interface index: 37, SNMP ifIndex: 45, Generation: 36
  Type: Multicast-GRE, Link-level type: GRE, MTU: Unlimited, Speed: 800mbps
  Hold-times     : Up 0 ms, Down 0 ms
  Device flags   : Present Running
  Interface flags: SNMP-Traps
  Statistics last cleared: Never
  Traffic statistics:
   Input  bytes  :              2887970                    0 bps
   Output bytes  :                    0                    0 bps
   Input  packets:                31896                    0 pps
   Output packets:                    0                    0 pps
  Logical interface mt-1/1/0.32769 (Index 43) (SNMP ifIndex 0) (Generation 46)
    Flags: Point-To-Point SNMP-Traps
    IP-Header 239.1.1.1:10.254.71.46:47:df:64:0000000800000000
    Encapsulation: GRE-NULL
    Traffic statistics:
     Input  bytes  :                   0
     Output bytes  :                2396
     Input  packets:                   0
     Output packets:                  34
    Local statistics:
     Input  bytes  :                   0
     Output bytes  :                2396
     Input  packets:                   0
     Output packets:                  34
    Transit statistics:
     Input  bytes  :                   0                    0 bps
     Output bytes  :                   0                    0 bps
     Input  packets:                   0                    0 pps
     Output packets:                   0                    0 pps
    Protocol inet, MTU: 4446, Generation: 79, Route table: 3
```

```
      Flags: None
  Logical interface mt-1/1/0.49154 (Index 44) (SNMP ifIndex 0) (Generation 47)
    Flags: Point-To-Point SNMP-Traps Encapsulation: GRE-NULL
    Traffic statistics:
     Input  bytes  :                1550
     Output bytes  :                   0
     Input  packets:                  33
     Output packets:                   0
    Local statistics:
     Input  bytes  :                1550
     Output bytes  :                   0
     Input  packets:                  33
     Output packets:                   0
    Transit statistics:
     Input  bytes  :                   0                    0 bps
     Output bytes  :                   0                    0 bps
     Input  packets:                   0                    0 pps
     Output packets:                   0                    0 pps
    Protocol inet, MTU: Unlimited, Generation: 80, Route table: 3
      Flags: None

user@PE0> show pim join extensive instance VPN-A
Instance: PIM.VPN-A Family: INET
Group: 229.1.1.1
    Source: 192.168.295.34
    Flags: dense
    Upstream interface: t1-0/3/0:0.0
    Downstream interfaces:
        mt-1/1/0.32769
Instance: PIM.VPN-A Family: INET6

user@PE0> show pim join
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.47
    Flags: sparse,rptree,wildcard
    Upstream interface: so-0/0/2.0
Group: 239.1.1.1
    Source: 10.254.71.46
    Flags: sparse
    Upstream interface: local
Instance: PIM.master Family: INET6
user@PE0>  show pim join instance VPN-A
Instance: PIM.VPN-A Family: INET
Group: 229.1.1.1
    Source: 192.168.295.34
    Flags: dense
    Upstream interface: t1-0/3/0:0.0
Instance: PIM.VPN-A Family: INET6

user@PE0> show multicast route extensive
Family: INET
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout
239.1.1.1       10.254.71.46   /32 A   F  86    9174       0          360
    Upstream interface: local
    Session name: Administratively Scoped
    Forwarding rate: 1 kBps (10 pps)
239.1.1.1       10.254.71.51   /32 A   F  96    36         0          360
    Upstream interface: so-0/0/2.0
    Session name: Administratively Scoped
```

```
    Forwarding rate: 0 kBps (0 pps)
Family: INET6
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout

user@PE0> show multicast route extensive instance VPN-A
Family: INET
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout
229.1.1.1       192.168.295.34 /32 A   F  85    9408       0          360
    Upstream interface: t1-0/3/0:0.0
    Session name: Unknown
    Forwarding rate: 1 kBps (10 pps)
Family: INET6
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout

user@PE0>  show multicast next-hops
Family: INET
ID      Refcount  KRefcount Downstream interface
86             2          1 so-0/0/2.0
85             2          1 mt-1/1/0.32769
96             2          1 mt-1/1/0.49154
Family: INET6
```

Router P0

```
user@P0> show pim join
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.47
    Flags: sparse,rptree,wildcard
    Upstream interface: local
Group: 239.1.1.1
    Source: 10.254.71.46
    Flags: sparse,spt
    Upstream interface: so-0/0/2.0
Group: 239.1.1.1
    Source: 10.254.71.51
    Flags: sparse,spt
    Upstream interface: so-0/0/0.0
Instance: PIM.master Family: INET6

user@P0> show multicast route extensive
Family: INET
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout
239.1.1.1       10.254.71.46   /32 A   F  127   9906       195        360
    Upstream interface: so-0/0/2.0
    Session name: Administratively Scoped
    Forwarding rate: 1 kBps (10 pps)
239.1.1.1       10.254.71.51   /32 A   F  126   135        23         359
    Upstream interface: so-0/0/0.0
    Session name: Administratively Scoped
    Forwarding rate: 0 kBps (0 pps)
Family: INET6
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout

user@P0> show multicast next-hops
Family: INET
ID      Refcount  KRefcount Downstream interface
127            2          1 so-0/0/0.0
```

```
126            2          1 so-0/0/2.0
Family: INET6
```

Router PE1

```
user@PE1> show pim join extensive
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.47
    Flags: sparse,rptree,wildcard
    Upstream interface: so-0/1/0.0
    Upstream State: Join to RP
    Downstream Neighbors:
        Interface: mt-1/1/0.32769
            10.1.71.51 State: Join   Flags: SRW  Timeout: Infinity
Group: 239.1.1.1
    Source: 10.254.71.46
    Flags: sparse,spt-pending
    Upstream interface: so-0/1/0.0
    Upstream State: Join to Source
    Keepalive timeout: 199
    Downstream Neighbors:
        Interface: mt-1/1/0.32769
            10.1.71.51 State: Join   Flags: S    Timeout: Infinity
Group: 239.1.1.1
    Source: 10.254.71.51
    Flags: sparse
    Upstream interface: local
    Upstream State: Local Source, Prune to RP
    Keepalive timeout: 79
    Downstream Neighbors:
        Interface: so-0/1/0.0
            192.168.296.33 State: Join   Flags: S    Timeout: 174
        Interface: register to RP 10.254.71.47 on pe-1/1/0.32769
Instance: PIM.master Family: INET6

user@PE1>  show pim join extensive instance VPN-A
Instance: PIM.VPN-A Family: INET
Group: 229.1.1.1
    Source: 192.168.295.34
    Flags: dense
    Upstream interface: mt-1/1/0.32769
    Downstream interfaces:
        t1-1/0/0:0.0
Instance: PIM.VPN-A Family: INET6

user@PE1> show interfaces mt-1/1/0 extensive
Physical interface: mt-1/1/0, Enabled, Physical link is Up
  Interface index: 38, SNMP ifIndex: 45, Generation: 37
  Type: Multicast-GRE, Link-level type: GRE, MTU: Unlimited, Speed: 800mbps
  Hold-times     : Up 0 ms, Down 0 ms
  Device flags   : Present Running
  Interface flags: SNMP-Traps
  Statistics last cleared: Never
  Traffic statistics:
   Input  bytes  :              2265256                 7568 bps
   Output bytes  :                    0                    0 bps
   Input  packets:                24981                   10 pps
   Output packets:                    0                    0 pps
  Logical interface mt-1/1/0.32769 (Index 45) (SNMP ifIndex 0) (Generation 46)
    Flags: Point-To-Point SNMP-Traps
```

```
    IP-Header 239.1.1.1:10.254.71.51:47:df:64:0000000800000000
    Encapsulation: GRE-NULL
    Traffic statistics:
     Input  bytes  :                   0
     Output bytes  :               10934
     Input  packets:                   0
     Output packets:                 153
    Local statistics:
     Input  bytes  :                   0
     Output bytes  :               10934
     Input  packets:                   0
     Output packets:                 153
    Transit statistics:
     Input  bytes  :                   0                    0 bps
     Output bytes  :                   0                    0 bps
     Input  packets:                   0                    0 pps
     Output packets:                   0                    0 pps
    Protocol inet, MTU: 4418, Generation: 77, Route table: 1
      Flags: None
  Logical interface mt-1/1/0.49154 (Index 46) (SNMP ifIndex 0) (Generation 47)
    Flags: Point-To-Point SNMP-Traps Encapsulation: GRE-NULL
    Traffic statistics:
     Input  bytes  :             1820512
     Output bytes  :                   0
     Input  packets:               19848
     Output packets:                   0
    Local statistics:
     Input  bytes  :                5536
     Output bytes  :                   0
     Input  packets:                 120
     Output packets:                   0
    Transit statistics:
     Input  bytes  :             1814976                 7568 bps
     Output bytes  :                   0                    0 bps
     Input  packets:               19728                   10 pps
     Output packets:                   0                    0 pps
    Protocol inet, MTU: Unlimited, Generation: 78, Route table: 1
      Flags: None

user@PE1> show multicast route extensive
Family: INET
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout
239.1.1.1       10.254.71.46   /32 A   F  76    11014      0          360
    Upstream interface: so-0/1/0.0
    Session name: Administratively Scoped
    Forwarding rate: 1 kBps (10 pps)
239.1.1.1       10.254.71.51   /32 A   F  103   1          0          360
    Upstream interface: local
    Session name: Administratively Scoped
    Forwarding rate: 0 kBps (0 pps)
Family: INET6
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout

user@PE1>  show multicast route extensive instance VPN-A
Family: INET
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout
229.1.1.1       192.168.295.34 /32 A   F  99    10976      4          360
    Upstream interface: mt-1/1/0.49154
    Session name: Unknown
    Forwarding rate: 1 kBps (10 pps)
Family: INET6
```

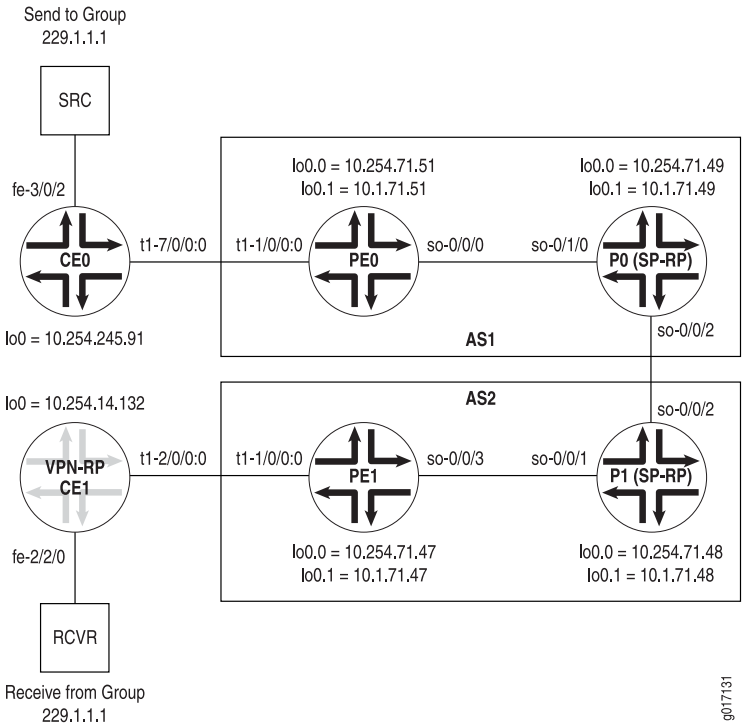
```
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout

user@PE1>  show multicast next-hops
Family: INET
ID      Refcount  KRefcount Downstream interface
75             2          1 so-0/1/0.0
99             2          1 t1-1/0/0:0.0
76             2          1 mt-1/1/0.49154
Family: INET6
```

Router CE1

```
user@CE1> show pim join
Instance: PIM.master Family: INET
Group: 229.1.1.1
    Source: 192.168.295.34
    Flags: dense
    Upstream interface: t1-7/0/0:0.0
Instance: PIM.master Family: INET6

user@CE1> show multicast route extensive
          2          1 fe-3/0/2.0

Family: INET
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout
229.1.1.1       192.168.295.34 /32 A   F  120   8010       0          360
    Upstream interface: t1-7/0/0:0.0
    Session name: Unknown
    Forwarding rate: 1 kBps (10 pps)
Family: INET6
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout

user@CE1> show multicast next-hops
Family: INET
ID      Refcount  KRefcount Downstream interface
120
```

Example: IPv4 Multicast with Interprovider VPNs Configuration

Figure 36: IPv4 Multicast with Interprovider VPNs Topology Diagram



Interprovider VPNs are also mentioned in RFC 4364. An example is shown in Figure 36 on page 461. The topology is slightly different; the main difference is the addition of MSDP between the two provider core transit (P) routers. In this limited topology, each P router is an SP-RP for the local autonomous system (AS), and Router CE1 is the VPN C-RP. VPN-A is the name of the routing instance on routers PE0 and PE1.

Router CE0

```
[edit]
protocols {
   pim {
      dense-groups {
         229.0.0.0/8;
      }
      rp {
         static {
            address 10.254.14.132;
         }
      }
      interface all {
         mode sparse-dense;
         version 2;
      }
      interface fxp0.0 {
         disable;
      }
```

```
    }
}
```

Router PE0

```
[edit]
protocols {
    pim {
        rp {
            static {
                address 10.254.71.49;
            }
        }
        interface all {
            mode sparse;
            version 2;
        }
        interface fxp0.0 {
            disable;
        }
    }
}
routing-instances {
    VPN-A {
        protocols {
            pim {
                dense-groups {
                    229.0.0.0/8;
                }
                vpn-group-address 239.1.1.1;
                rp {
                    static {
                        address 10.254.14.132;
                    }
                }
                interface t1-1/0/0:0.0 {
                    mode sparse-dense;
                    version 2;
                }
                interface lo0.1 {
                    mode sparse-dense;
                    version 2;
                }
            }
        }
    }
}
```

Router P0

```
[edit]
protocols {
    ...
    msdp {
        peer 10.254.71.48 {
            local-address 10.254.71.49;
        }
    }
    ...
```

```
    pim {
        rp {
            local {
                address 10.254.71.49;
            }
        }
        interface all {
            mode sparse;
            version 2;
        }
        interface fxp0.0 {
            disable;
        }
    }
}
```

Router P1

```
[edit]
protocols {
    ...
    msdp {
        peer 10.254.71.49 {
            local-address 10.254.71.48;
        }
    }
    ...
    pim {
        rp {
            local {
                address 10.254.71.48;
            }
        }
        interface all {
            mode sparse;
            version 2;
        }
        interface fxp0.0 {
            disable;
        }
    }
}
```

Router PE1

```
[edit]
protocols {
    pim {
        rp {
            static {
                address 10.254.71.48;
            }
        }
        interface all {
            mode sparse;
            version 2;
        }
        interface fxp0.0 {
            disable;
```

```
        }
      }
    }
    routing-instances {
      VPN-A {
        protocols {
          pim {
            dense-groups {
              229.0.0.0/8;
            }
            vpn-group-address 239.1.1.1;
            rp {
              static {
                address 10.254.14.132;
              }
            }
            interface t1-1/0/0:0.0 {
              mode sparse-dense;
              version 2;
            }
            interface lo0.1 {
              mode sparse-dense;
              version 2;
            }
          }
        }
      }
    }
```

Router CE1

```
[edit]
protocols {
  pim {
    dense-groups {
      229.0.0.0/8;
    }
    rp {
      local {
        address 10.254.14.132;
      }
    }
    interface all {
      mode sparse-dense;
      version 2;
    }
    interface fxp0.0 {
      disable;
    }
  }
}
```

Verifying Your Work

The show commands used to verify proper functionality of multicast in an interprovider environment are the same ones used with the first Layer 3 VPN multicast example (see "Verifying Your Work" on page 451).

The following output provides details for RP and the PIM join information:

- Router CE0 Status on page 465
- Router PE0 Status on page 465
- Router P0 Status on page 467
- Router P1 Status on page 468
- Router PE1 Status on page 469
- Router CE1 Status on page 470

Router CE0 Status

```
user@CE0> show pim rps extensive
Instance: PIM.master
Family: INET
RP: 10.254.14.132
Learned via: static configuration
Time Active: 00:21:35
Holdtime: 0
Device Index: 119
Subunit: 32769
Interface: pe-6/0/0.32769
Group Ranges:
        224.0.0.0/4
Active groups using RP:
Register State for RP:
Group           Source          FirstHop        RP Address      State    Timeout
Family: INET6

user@CE0> show pim join extensive
Instance: PIM.master Family: INET
Group: 229.1.1.1
    Source: 192.168.295.38
    Flags: dense
    Upstream interface: fe-3/0/2.0
    Downstream interfaces:
        t1-7/0/0:0.0
Instance: PIM.master Family: INET6
```

Router PE0 Status

```
user@PE0> show pim rps extensive
Instance: PIM.master
Family: INET
RP: 10.254.71.49
Learned via: static configuration
```

```
Time Active: 00:22:07
Holdtime: 0
Device Index: 34
Subunit: 32769
Interface: pe-1/1/0.32769
Group Ranges:
        224.0.0.0/4
Active groups using RP:
        239.1.1.1
        total 1 groups active
Register State for RP:
Group           Source          FirstHop        RP Address      State    Timeout
239.1.1.1       10.254.71.51    10.254.71.51    10.254.71.49    Suppress      20
Family: INET6

user@PE0> show pim rps extensive instance VPN-A
Instance: PIM.VPN-A
Family: INET
RP: 10.254.14.132
Learned via: static configuration
Time Active: 00:22:22
Holdtime: 0
Device Index: 34
Subunit: 32771
Interface: pe-1/1/0.32771
Group Ranges:
        224.0.0.0/4
Active groups using RP:
Register State for RP:
Group           Source          FirstHop        RP Address      State    Timeout
Family: INET6

user@PE0>  show pim join extensive
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.49
    Flags: sparse,rptree,wildcard
    Upstream interface: so-0/0/0.0
    Upstream State: Join to RP
    Downstream Neighbors:
        Interface: mt-1/1/0.32769
            0.0.0.0 State: Join   Flags: SRW  Timeout: Infinity
Group: 239.1.1.1
    Source: 10.254.71.47
    Flags: sparse,spt-pending
    Upstream interface: so-0/0/0.0
    Upstream State: Join to Source
    Keepalive timeout: 198
    Downstream Neighbors:
        Interface: mt-1/1/0.32769
            0.0.0.0 State: Join   Flags: S    Timeout: Infinity
Group: 239.1.1.1
    Source: 10.254.71.51
    Flags: sparse
    Upstream interface: local
    Upstream State: Local Source, Prune to RP
    Keepalive timeout: 198
    Downstream Neighbors:
        Interface: so-0/0/0.0
            192.168.296.42 State: Join   Flags: S    Timeout: 176
```

```
Instance: PIM.master Family: INET6

user@PE0> show pim join extensive instance VPN-A
Instance: PIM.VPN-A Family: INET
Group: 229.1.1.1
    Source: 192.168.295.38
    Flags: dense
    Upstream interface: t1-1/0/0:0.0
    Downstream interfaces:
        mt-1/1/0.32769
Instance: PIM.VPN-A Family: INET6
```

Router P0 Status

```
user@P0> show pim rps extensive
Instance: PIM.master
Family: INET
RP: 10.254.71.49
Learned via: static configuration
Time Active: 00:30:43
Holdtime: 0
Device Index: 33
Subunit: 32768
Interface: pd-1/1/0.32768
Group Ranges:
        224.0.0.0/4
Active groups using RP:
        239.1.1.1
        total 1 groups active
Register State for RP:
Group           Source          FirstHop        RP Address      State    Timeout
239.1.1.1       10.254.71.51    10.254.71.51    10.254.71.49    Receive
Family: INET6

user@P0> show pim join extensive
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.49
    Flags: sparse,rptree,wildcard
    Upstream interface: local
    Upstream State: Local RP
    Downstream Neighbors:
        Interface: so-0/1/0.0
            192.168.296.41 State: Join   Flags: SRW  Timeout: 184
Group: 239.1.1.1
    Source: 10.254.71.47
    Flags: sparse,spt-pending
    Upstream interface: so-0/0/2.0
    Upstream State: Local RP, Join to Source
    Keepalive timeout: 207
    Downstream Neighbors:
        Interface: so-0/1/0.0
            192.168.296.41 State: Join   Flags: S    Timeout: 184
Group: 239.1.1.1
    Source: 10.254.71.51
    Flags: sparse,spt
    Upstream interface: so-0/1/0.0
```

```
    Upstream State: Local RP, Join to Source
    Keepalive timeout: 207
    Downstream Neighbors:
        Interface: so-0/0/2.0
            192.168.296.73 State: Join   Flags: S    Timeout: 186
        Interface: so-0/1/0.0             (pruned)
            192.168.296.41 State: Prune  Flags: SR   Timeout: 184
Instance: PIM.master Family: INET6
```

Router P1 Status

```
user@P1> show pim rps extensive
Instance: PIM.master
Family: INET
RP: 10.254.71.48
Learned via: static configuration
Time Active: 06:26:56
Holdtime: 0
Device Index: 32
Subunit: 32768
Interface: pd-1/1/0.32768
Group Ranges:
        224.0.0.0/4
Active groups using RP:
        239.1.1.1
        total 1 groups active
Register State for RP:
Group           Source          FirstHop        RP Address      State    Timeout
239.1.1.1       10.254.71.47    10.254.71.47    10.254.71.48    Receive        0
Family: INET6

user@P1> show pim join extensive
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.48
    Flags: sparse,rptree,wildcard
    Upstream interface: local
    Upstream State: Local RP
    Downstream Neighbors:
        Interface: so-0/0/1.0
            192.168.296.50 State: Join   Flags: SRW  Timeout: 174
Group: 239.1.1.1
    Source: 10.254.71.47
    Flags: sparse,spt
    Upstream interface: so-0/0/1.0
    Upstream State: Local RP, Join to Source
    Keepalive timeout: 196
    Downstream Neighbors:
        Interface: so-0/0/1.0             (pruned)
            192.168.296.50 State: Prune  Flags: SR   Timeout: 174
        Interface: so-0/0/2.0
            192.168.296.74 State: Join   Flags: S    Timeout: 178
Group: 239.1.1.1
    Source: 10.254.71.51
    Flags: sparse,spt-pending
    Upstream interface: so-0/0/2.0
    Upstream State: Local RP, Join to Source
```

```
    Keepalive timeout: 196
    Downstream Neighbors:
        Interface: so-0/0/1.0
            192.168.296.50 State: Join   Flags: S    Timeout: 174
Instance: PIM.master Family: INET6
```

Router PE1 Status

```
user@PE1> show pim rps extensive
Instance: PIM.master
Family: INET
RP: 10.254.71.48
Learned via: static configuration
Time Active: 00:25:13
Holdtime: 0
Device Index: 34
Subunit: 32770
Interface: pe-1/1/0.32770
Group Ranges:
        224.0.0.0/4
Active groups using RP:
        239.1.1.1
        total 1 groups active
Register State for RP:
Group           Source          FirstHop        RP Address      State    Timeout
239.1.1.1       10.254.71.47    10.254.71.47    10.254.71.48    Suppress      42
Family: INET6

user@PE1> show pim rps extensive instance VPN-A
Instance: PIM.VPN-A
Family: INET
RP: 10.254.14.132
Learned via: static configuration
Time Active: 00:25:17
Holdtime: 0
Device Index: 34
Subunit: 32771
Interface: pe-1/1/0.32771
Group Ranges:
        224.0.0.0/4
Active groups using RP:
Register State for RP:
Group           Source          FirstHop        RP Address      State    Timeout
Family: INET6

user@PE1>  show pim join extensive
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.48
    Flags: sparse,rptree,wildcard
    Upstream interface: so-0/0/3.0
    Upstream State: Join to RP
    Downstream Neighbors:
        Interface: mt-1/1/0.32769
            0.0.0.0 State: Join   Flags: SRW  Timeout: Infinity
Group: 239.1.1.1
    Source: 10.254.71.47
```

```
    Flags: sparse
    Upstream interface: local
    Upstream State: Local Source, Prune to RP
    Keepalive timeout: 173
    Downstream Neighbors:
        Interface: so-0/0/3.0
            192.168.296.49 State: Join   Flags: S    Timeout: 199
Group: 239.1.1.1
    Source: 10.254.71.51
    Flags: sparse,spt-pending
    Upstream interface: so-0/0/3.0
    Upstream State: Join to Source
    Keepalive timeout: 173
    Downstream Neighbors:
        Interface: mt-1/1/0.32769
            0.0.0.0 State: Join   Flags: S    Timeout: Infinity
Instance: PIM.master Family: INET6

user@PE1>  show pim join extensive instance VPN-A
Instance: PIM.VPN-A Family: INET
Group: 229.1.1.1
    Source: 192.168.295.38
    Flags: dense
    Upstream interface: mt-1/1/0.32769
    Downstream interfaces:
        t1-1/0/0:0.0
Instance: PIM.VPN-A Family: INET6
```

Router CE1 Status

```
user@CE1> show pim rps extensive
Instance: PIM.master
Family: INET
RP: 10.254.14.132
Learned via: static configuration
Time Active: 00:28:22
Holdtime: 0
Device Index: 69
Subunit: 32768
Interface: pd-3/1/0.32768
Group Ranges:
        224.0.0.0/4
Active groups using RP:
Register State for RP:
Group           Source          FirstHop        RP Address      State    Timeout
Family: INET6

user@CE1>  show pim join extensive
Instance: PIM.master Family: INET
Group: 229.1.1.1
    Source: 192.168.295.38
    Flags: dense
    Upstream interface: t1-2/0/0:0.0
    Downstream interfaces:
        fe-2/2/0.0
Instance: PIM.master Family: INET6
```

For More Information

For additional information about multicast over Layer 3 VPNs, see the following resources:

- *JUNOS Multicast Protocols Configuration Guide*
- *JUNOS VPNs Configuration Guide*
- *RFC 2547, BGP/MPLS VPNs*
- *RFC 3618, Multicast Source Discovery Protocol (MSDP)*
- *RFC 4364, BGP/MPLS IP Virtual Private Networks (VPNs)*
- Internet draft draft-rosen-vpn-mcast-06.txt, *Multicast in MPLS/BGP VPNs* (expires April 2004)

Revision History

5 October 2009—Added Configuring MBGP MVPNs to Support IPv6 Multicast Traffic. 10.0R1 Release. Roy Spencer.

13 June 2009—Added MBGP Multicast VPN Extranets Overview, MBGP Multicast VPN Extranets Configuration Guidelines, Configuring MBGP Multicast VPN Extranets Example, and Configuring MBGP MVPN Senders and Receivers using PIM SM and RSVP-TE Provider Tunnels Example. 9.6R1 Release. Roy Spencer.

13 June 2009—9.6R1 Release. Roy Spencer.

13 April 2009—9.5R1 Release. Roy Spencer.

15 January 2009—9.4R1 Release. Roy Spencer.

10 October 2008—9.3R1 Release. Roy Spencer.

10 July 2008—Added the Configuring MBGP Multicast VPNs example. 9.2R1 Release. Roy Spencer.

10 April 2008—9.1R1 Release. Roy Spencer.

1 February 2008—9.0R1 Release. Fawn Damitio.

12 October 2007—Added point-to-multipoint traffic engineering provider tunnel support. 8.5R1 Release. Fawn Damitio.

29 June 2007—Added multiprotocol BGP multicast VPN: next-generation support. 8.4R1 Release. Fawn Damitio.

27 March 2007—8.3R1 Release. Fawn Damitio.

12 January 2007—Added support for MX960 Ethernet Services Routers. 8.2R1 Release. Fawn Damitio.

15 September 2006—Added RFC 4364, 8.1R1 Release. Richard Hendricks.

29 June 2006—8.0R1 Release. Richard Hendricks.

27 March 2006—7.6R1 Release. Richard Hendricks.

9 January 2006—7.5R1 Release. Richard Hendricks.

14 September 2005—7.4R1 Release. Richard Hendricks.

13 June 2005—7.3R1 Release. Richard Hendricks.

5 April 2005—Added support for MSDP in Layer 3 VPNs, 7.2R1 Release. Richard Hendricks.

2 February 2005—Added multicast distribution trees for data, 7.1R1 Release. Richard Hendricks.

6 October 2004—7.0R1 Release. Richard Hendricks.

6 July 2004—Added graceful restart information and mentioned that a PE router can now act as a VPN C-RP, 6.4R1 Release. Richard Hendricks.

5 April 2004—6.3R1 Release. Richard Hendricks.

22 December 2003—6.2R1 Release. Richard Hendricks.

22 September 2003—6.1R1 Release. Richard Hendricks.

30 June 2003—6.0R1 Release. Richard Hendricks.

2 April 2003—5.7R1 Release. Richard Hendricks.

27 December 2002—5.6R1 Release. Richard Hendricks.

30 September 2002—5.5R1 Release. Richard Hendricks.

27 August 2002—Reformatted the document in *Feature Guide* style. Richard Hendricks.

22 August 2002—Added PIM dense mode information. Bill Nowak.

8 February 2002—Initial 5.3 Quick Start Guide document. Bill Nowak.

Chapter 12

Translational Cross-Connect and Layer 2.5 VPNs

This chapter covers the following topics:

- Translational Cross-Connect and Layer 2.5 VPNs Overview on page 473
- Translational Cross-Connect and Layer 2.5 VPNs System Requirements on page 474
- Translational Cross-Connect and Layer 2.5 VPNs Terms and Acronyms on page 475
- Configuring TCC Interface Switching on page 476
- Defining the Encapsulation for Layer 2 TCC Switching on page 477
- Configuring ARP Functionality for Ethernet TCC Circuits on page 478
- Configuring ARP Functionality for Extended VLAN Circuits on page 478
- Configuring Static ARP For Ethernet and Extended VLAN TCCs on page 479
- Defining the Connection for Layer 2 TCC Switching on page 480
- Configuring MPLS on page 480
- TCC Configuration Examples on page 480
- Example: PPP to ATM TCC Configuration on page 481
- Example: Frame Relay to Fast Ethernet TCC Configuration on page 483
- Configuring Layer 2.5 VPNs on page 485
- Configuring the Encapsulation on Interfaces Participating in the Layer 2.5 VPN on page 486
- Configuring the Layer 2.5 VPN on page 487
- Option: Configuring ISO or MPLS Traffic on T Series and M320 Routers on page 487
- Example: Layer 2.5 VPN Configuration on page 488
- For More Information on page 497
- Revision History on page 498

Translational Cross-Connect and Layer 2.5 VPNs Overview

Translational cross-connect (TCC) is a switching concept that allows you to establish interconnections between a variety of Layer 2 protocols or circuits. It is similar to its predecessor, circuit cross-connect (CCC). However, while CCC requires the same Layer 2 encapsulations on both sides of a Juniper Networks router (such as PPP-to-PPP

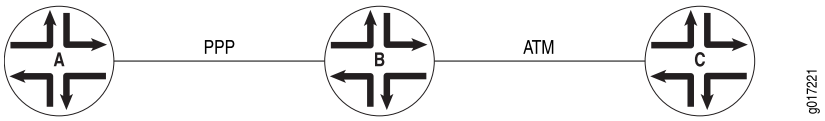
or Frame Relay-to-Frame Relay), TCC lets a network administrator connect different types of Layer 2 protocols interchangeably. With TCC, combinations such as PPP-to-ATM and Ethernet-to-Frame Relay cross-connections are possible. Also, TCC can be used to create Layer 2.5 VPNs and Layer 2.5 circuits.

The JUNOS Software makes interworking between unlike protocols possible. The software strips off the Layer 2 header when frames enter the router and adds a different Layer 2 header before the frames exit the router. TCC supports these Layer 2 protocols:

- ATM
- Cisco HDLC
- Ethernet
- Extended VLAN
- Frame Relay
- PPP

In Figure 37 on page 474, the PPP header is stripped from frames arriving at Router B and an ATM header is added before the frames are sent to Router C. All Layer 2 negotiations are terminated at the interconnecting router (Router B). Examples include Link Control Protocol (LCP) and Network Control Protocol (NCP) for PPP, keepalives for Cisco HDLC, and Local Management Interface (LMI) for Frame Relay.

Figure 37: TCC Concept Example



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.

This guide shows you how to use the Layer 2 interworking nature of TCC for interface switching and for constructing Layer 2.5 virtual private networks (VPNs).

Translational Cross-Connect and Layer 2.5 VPNs System Requirements

To implement TCC, your system must meet these minimum requirements:

- JUNOS Release 8.3 or later for support of IS-IS and MPLS traffic over Layer 2.5 VPNs.
- JUNOS Release 8.2 or later for support on M120 routers and MX Series routers
- JUNOS Release 6.2 or later for CCC, TCC, and Layer 2.5 VPN support on M320 routers

- JUNOS Release 6.0 or later for CCC, TCC, and Layer 2.5 VPN support on T Series routers
- JUNOS Release 5.6 or later for proxy Address Resolution Protocol (ARP) functionality in Ethernet TCC and extended VLAN TCC networks on M Series routers
- JUNOS Release 5.4 or later for Ethernet TCC and extended VLAN TCC interface encapsulation types on M Series routers
- JUNOS Release 5.2 or later for PPP TCC, Cisco HDLC TCC, ATM TCC, and Frame Relay TCC interface encapsulation types on M Series routers
- Ethernet-based PICs to support Ethernet TCC and extended VLAN TCC interface encapsulation types, with the following exceptions:
  - All 8-port, 12-port, and 48-port Fast Ethernet PICs do not support TCC.
  - 4-port Gigabit Ethernet PICs do not support extended VLAN TCC.
- Any combination of three Juniper Networks M Series or T Series routers for TCC or five routers for Layer 2.5 VPNs or Layer 2.5 circuits

Translational Cross-Connect and Layer 2.5 VPNs Terms and Acronyms

C

circuit cross-connect (CCC) A method of exchanging frames between two router interfaces running the same Layer 2 protocol, such as ATM, Cisco HDLC, Ethernet, extended VLAN, Frame Relay, and PPP. For more information about CCC, see the *JUNOS Network Interfaces Configuration Guide*.

customer edge router (CE) Any router that connects a customer site to a PE router.

L

Layer 2.5 circuits A method of privately connecting sites across the Internet. Layer 2.5 circuits function like Layer 2 circuits, but connect two sites that use different Layer 2 protocols. For more information about Layer 2 circuits, see “Layer 2 Circuits” on page 285 or the *JUNOS VPNs Configuration Guide*.

Layer 2.5 VPNs A method of privately connecting sites across the Internet. Layer 2.5 VPNs function like Layer 2 VPNs, but connect two sites that use different Layer 2 protocols. For more information about Layer 2 VPNs, see the *JUNOS VPNs Configuration Guide*.

P

provider core router (P) Any router that lies between PE routers in the provider core network.

provider edge router (PE) Any router that connects customer edge (CE) routers to the provider core network.

T

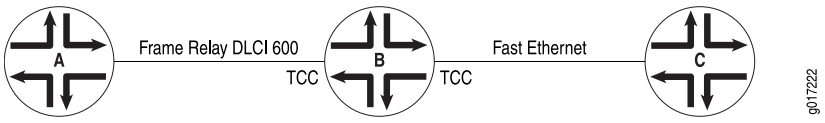
translational cross-connect (TCC) A Juniper Networks method of exchanging frames between two router interfaces running different Layer 2 protocols, such as ATM, Cisco HDLC, Ethernet, extended VLAN, Frame Relay, and PPP. For more information about TCC, see the *JUNOS Network Interfaces Configuration Guide*.

Configuring TCC Interface Switching

TCC joins disparate Layer 2 logical interfaces by means of a virtual Layer 2 switching technique.

Figure 38 on page 476 illustrates Layer 2 TCC switching. In this example, Router A is connected by Frame Relay to Router B and Router B is connected by Fast Ethernet to Router C. Router B is a Juniper Networks router. TCC allows you to configure Router B to act as a multiservice Layer 2 switch. To do this, you configure a connection from Router A to Router C that passes through Router B. This effectively establishes Router B as a Frame Relay-to-Fast Ethernet switch and allows the router to switch frames transparently between Router A and Router C without regard to the frames' contents or the Layer 3 protocols. Router B removes the Frame Relay header from frames arriving on data-link connection identifier (DLCI) 600 and adds an Ethernet header onto the frames before sending them to the Fast Ethernet link.

Figure 38: Layer 2 TCC Switching



You can configure Layer 2 TCC switching on ATM, Cisco HDLC, Ethernet, extended VLAN, Frame Relay, and PPP circuits. TCC enables unlike interfaces to be connected with a single cross-connect.

To configure Layer 2 TCC switching, perform the following tasks on the router that is acting as the switch (Router B in Figure 38 on page 476):

- Defining the Encapsulation for Layer 2 TCC Switching on page 477
- Configuring ARP Functionality for Ethernet TCC Circuits on page 478
- Configuring ARP Functionality for Extended VLAN Circuits on page 478
- Configuring Static ARP For Ethernet and Extended VLAN TCCs on page 479
- Defining the Connection for Layer 2 TCC Switching on page 480
- Configuring MPLS on page 480

Defining the Encapsulation for Layer 2 TCC Switching

To begin implementation of Layer 2 TCC switching, configure TCC encapsulation on the desired interfaces of the router that is acting as the switch (Router B in Figure 38 on page 476).

NOTE: You cannot configure standard protocol families on TCC or CCC interfaces. Only the CCC family is allowed on CCC-encapsulated interfaces. Likewise, only the TCC family is allowed on TCC-encapsulated interfaces.

For ATM connections, specify the encapsulation type when configuring ATM virtual circuits (VCs) at the [edit interfaces *interface-name* unit *unit-number*] hierarchy level. For each VC, you configure whether it is a circuit or a regular logical interface. The default interface type is point-to-point.

```
[edit]
interfaces {
   at-fpc/pic/port {
      atm-options {
         vpi vpi-identifier  maximum-vcs maximum-vcs;
      }
      unit logical-unit-number {
         point-to-point; # This is the default interface type.
         encapsulation (atm-tcc-vc-mux | atm-tcc-snap);
         vci vpi-identifier.vci-identifier;
      }
   }
}
```

For Cisco HDLC and PPP circuits, specify the encapsulation in the encapsulation statement at the [edit interfaces *interface-name*] hierarchy level. This statement configures the entire physical device. Also, you must configure the logical interface unit 0.

```
[edit]
interfaces {
   type-fpc/pic/port {
      encapsulation (cisco-hdlc-tcc | ppp-tcc);
      unit 0;
   }
}
```

You can specify the encapsulation for Frame Relay circuits at the [edit interfaces *interface-name*] hierarchy level and the [edit interfaces *interface-name* unit *unit-number*] hierarchy level. For TCC and CCC interfaces, the DLCI value must be configured in the range of 512 through 1022. The default interface type is point-to-point.

```
[edit]
interfaces {
   type-fpc/pic/port  {
      encapsulation frame-relay-tcc;
```

```
        unit logical-unit-number {
            point-to-point; # This is the default interface type.
            encapsulation frame-relay-tcc;
            dlci dlci-identifier;
        }
    }
}
```

Configuring ARP Functionality for Ethernet TCC Circuits

For Ethernet TCC circuits, include the encapsulation ethernet-tcc statement. This statement configures the encapsulation for the entire physical interface.

To provide Address Resolution Protocol (ARP) functionality for an Ethernet-based neighbor, configure the remote statement at the [edit interfaces *interface-name* unit *unit-number* family tcc] hierarchy level and specify either the MAC address or IP address of the TCC router's Ethernet neighbor. To complete the setup of ARP, configure the proxy statement at the [edit interfaces *interface-name* unit *unit-number* family tcc] hierarchy level and specify the IP address of the TCC router's non-Ethernet neighbor.

```
[edit]
interfaces
EthernetType-fpc/pic/port {
    encapsulation ethernet-tcc;
    unit 0 {
        family tcc {
            remote { # Addresses associated with the Ethernet TCC neighbor.
                mac-address mac-address; # Select a MAC or IP address.
                inet-address inet-address;
            }
            proxy { # Addresses belonging to the non-Ethernet TCC neighbor.
                inet-address inet-address;
            }
        }
    }
}
```

Configuring ARP Functionality for Extended VLAN Circuits

For extended VLAN circuits, include the encapsulation extended-vlan-tcc statement. This statement configures the encapsulation for the entire physical interface.

You must also enable VLAN tagging. Ethernet interfaces in VLAN mode can have multiple logical interfaces. For encapsulation type extended-vlan-tcc, all VLAN IDs from 0 through 4094 are valid, up to a maximum of 1024 VLANs.

To enable ARP functionality, configure the remote statement at the [edit interfaces *interface-name* unit *unit-number* family tcc] hierarchy level with either the MAC address or IP address of your Ethernet TCC neighbor. To complete the ARP setup, configure the proxy statement at the [edit interfaces *interface-name* unit *unit-number* family tcc] hierarchy level and specify the IP address of the non-Ethernet TCC neighbor.

```
[edit]
```

```
interfaces {
   EthernetType-fpc/pic/port {
      vlan-tagging;
      encapsulation extended-vlan-tcc;
      unit 0 {
         vlan-id 600;
         family tcc {
            remote { # Addresses associated with the Ethernet TCC neighbor.
               mac-address mac-address; # Select a MAC or IP address.
               inet-address inet-address;
            }
            proxy { # Addresses belonging to the non-Ethernet TCC neighbor.
               inet-address inet-address;
            }
         }
      }
   }
}
```

Configuring Static ARP For Ethernet and Extended VLAN TCCs

If you do not use the proxy statement to configure ARP functionality in an Ethernet TCC or extended VLAN TCC, you must use another method to allow ARP to function. To retain the functionality of ARP for Ethernet networks, you must configure static ARP on the Ethernet neighbor. Use of static ARP assumes that you have already configured the remote statement on the TCC router (see "Configuring ARP Functionality for Ethernet TCC Circuits" on page 478 and "Configuring ARP Functionality for Extended VLAN Circuits" on page 478).

You configure the arp statement on the Ethernet neighbor at the [edit interfaces *interface-number* unit *unit-number* family inet address *ip-address*] hierarchy level. Your static ARP statement must contain the IP address of the non-Ethernet neighbor on the opposite side of the TCC router and the Ethernet interface MAC address of the TCC router. This static ARP configuration enables return path ARP functionality and complements the remote statement previously set on the TCC router.

In Figure 40 on page 483, you would configure an ARP statement on the fe-0/0/0 interface of Router C. The ARP statement would contain the IP address for interface so-0/1/0.600 on Router A and the MAC address of the fe-1/0/0 interface of Router B.

Configure static ARP on an Ethernet neighbor at the [edit interfaces *interface-name* unit *unit-number* family inet address *ip-address*] hierarchy level.

```
[edit]
interfaces
EthernetType-fpc/pic/port {
   unit 0 {
      family inet {
         address ip-address { # The local IP address.
            arp ip-address mac mac-address; # IP address of the non-Ethernet
         }         # TCC neighbor and MAC address of the TCC
      }         # router's Ethernet interface.
   }
}
```

Defining the Connection for Layer 2 TCC Switching

The next step in configuring Layer 2 TCC switching is to define the connection between the two circuits. You configure this on the router acting as the TCC switch. 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 by the second interface, and those received on the second interface are transmitted by the first.

```
[edit]
protocols {
   connections {
      interface-switch connection-name {
         interface first-interface-name.unit-number;
         interface second-interface-name.unit-number;
      }
   }
}
```

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;
   }
}
```

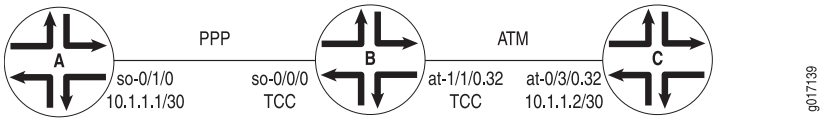
TCC Configuration Examples

This section contains configuration examples and commands you can issue to verify your Layer 2 TCC switching configuration:

- Example: PPP to ATM TCC Configuration on page 481
- Example: Frame Relay to Fast Ethernet TCC Configuration on page 483

Example: PPP to ATM TCC Configuration

Figure 39: TCC Interface Switching—PPP to ATM



In Figure 39 on page 481, Router A uses PPP to connect with Router B, while Router C connects with Router B through ATM. Router B acts as the Layer 2 virtual switch and transparently connects Router A to Router C.

On Router A, configure basic PPP encapsulation and any desired Layer 3 protocol families on the SONET/SDH interface.

Router A

```
[edit]
interfaces {
   so-0/1/0 {
      description "to Router B so-0/0/0";
      unit 0 {
         encapsulation ppp;
         family inet {
            address 10.1.1.1/30;
         }
      }
   }
}
```

Router B acts as the virtual Layer 2 switch. Here you configure the appropriate TCC encapsulations on the corresponding interfaces. In this case, encapsulation ppp-tcc is bound to physical interface so-0/0/0, and encapsulation atm-tcc-vc-mux is placed on VC 32 of interface at-1/1/0. Because the switching occurs at Layer 2, you cannot configure IP addresses or other Layer 3 family information on these interfaces.

You also need to configure MPLS and establish the cross-connect by adding the necessary interfaces to the interface-switch statement at the [edit protocols connections] hierarchy level.

Router B

```
[edit]
interfaces {
   so-0/0/0 {
      description "to Router A so-0/1/0";
      encapsulation ppp-tcc;
      unit 0 {
      }
      at-1/1/0 {
         description "to Router C at-0/3/0";
         atm-options {
            vpi 0 maximum-vcs 2000;
         }
         unit 32 {
            vci 32;
```

```
            encapsulation atm-tcc-vc-mux;
          }
        }
      }
      protocols {
        mpls {
          interface so-0/0/0.0;
          interface at-1/1/0.32;
        }
        connections {
          interface-switch PPP-to-ATM {
            interface so-0/0/0.0;
            interface at-1/1/0.32;
          }
        }
      }
    }
```

On Router C, the encapsulation option used to connect to the TCC-encapsulated ATM interface on Router B is atm-vc-mux. Since this ATM connection is switched at Layer 2 to reach the PPP link, it is transparent to Layer 3 addressing. As a result, the IP address must be configured in the same address space as Router A's so-0/1/0 interface.

Router C

```
[edit]
interfaces {
  at-0/3/0 {
    description "to Router B at-1/1/0";
    atm-options {
      vpi 0 maximum-vcs 2000;
    }
    unit 32 {
      vci 32;
      encapsulation atm-vc-mux;
      family inet {
        address 10.1.1.2/30;
      }
    }
  }
}
```

Verifying Your Work

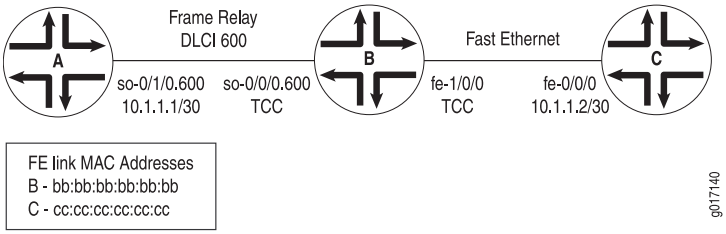
To verify your TCC connection, use the show connections command on Router B:

```
user@router_b> 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
  Dn -- down                          Legend for circuit types
  -> -- only outbound conn is up      intf -- interface
```

```
  <- -- only inbound  conn is up      tlsp -- transmit LSP
  Up -- operational                   rlsp -- receive LSP
Connection/Circuit                  Type    St  Time last up     # Up trans
PPP-to-ATM                          if-sw   Up  Nov 30 08:57:53           1
  at-1/1/0.32                         intf  Up
  so-0/0/0.0                          intf  Up
```

Example: Frame Relay to Fast Ethernet TCC Configuration

Figure 40: TCC Interface Switching—Frame Relay to Fast Ethernet



In the configuration example in Figure 40 on page 483, Router A uses Frame Relay to connect with Router B, while Router C connects to Router B by using Fast Ethernet. Router B acts as the Layer 2 virtual switch and transparently connects Router A to Router C.

You must enable Frame Relay encapsulation on Router A at the physical interface level.

Router A

```
[edit]
interfaces {
   so-0/1/0 {
      description "to Router B so-0/0/0";
      encapsulation frame-relay;
      unit 600 {
         point-to-point;
         dlci 600;
         family inet {
            address 10.1.1.1/30;
         }
      }
   }
}
```

Router B acts as the virtual switch. Enable the appropriate TCC encapsulations on the corresponding interfaces. In this case, configure the encapsulation frame-relay-tcc option on the logical and physical interfaces of so-0/0/0.600. Next, add the ethernet-tcc encapsulation type to the physical interface of fe-1/0/0. To enable ARP, configure the remote MAC address or IP address of Router C's Fast Ethernet interface with the remote statement at the [edit interfaces *interface-name* unit 0 family tcc] hierarchy level. To enable proxy ARP, include the proxy statement at the [edit interfaces *interface-name* unit 0 family tcc] hierarchy level and specify the IP address of Router A.

After configuring the correct interface encapsulations, complete your cross-connect by adding both interfaces into your MPLS configuration. Include the same interfaces in the interface-switch statement at the [edit protocols connections] hierarchy level.

Router B

```
[edit]
interfaces {
   so-0/0/0 {
      description "to Router A so-0/1/0";
      dce;
      encapsulation frame-relay-tcc;
      unit 600 {
         point-to-point;
         encapsulation frame-relay-tcc;
         dlci 600;
      }
   }
   fe-1/0/0 {
      description "to Router C fe-0/0/0";
      encapsulation ethernet-tcc;
      unit 0 {
         family tcc {
            protocol inet
            remote { # Addresses associated with the Ethernet TCC neighbor Router C.
               mac-address cc:cc:cc:cc:cc:cc; # Or, specify Router C's IP address here.
            }
            proxy { # Addresses associated with the other TCC neighbor—Router A.
               inet-address 10.1.1.1;
            }
         }
      }
   }
}
protocols {
   mpls {
      interface so-0/0/0.600;
      interface fe-1/0/0.0;
   }
   connections {
      interface-switch FR-to-Ether {
         interface so-0/0/0.600;
         interface fe-1/0/0.0;
      }
   }
}
```

Ethernet encapsulation is the default for Router C. Because the Fast Ethernet connection is switched at Layer 2 to reach the Frame Relay link, it is transparent to Layer 3 addressing. As a result, you must configure the IP address for the fe-0/0/0 interface in the same address space as Router A's so-0/1/0.600 interface.

Optionally, configure static ARP on the fe-0/0/0 interface if you omit the proxy statement on Router B. The arp statement must contain the IP address from interface so-0/1/0.600 on Router A and the MAC address of the Fast Ethernet interface on Router B.

Router C

```
[edit]
interfaces
fe-0/0/0 {
   description "to Router B fe-1/0/0";
   unit 0 {
      family inet {
         address 10.1.1.2/30 {
            arp 10.1.1.1 mac bb:bb:bb:bb:bb:bb; # Configure this only if you did not
         }       # enter a proxy statement on Router B.
      }
   }
}
```

Verifying Your Work

To verify the operational status of your TCC connection, use the show connections command on Router B:

```
user@router_b> 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
  Dn -- down                          Legend for circuit types
  -> -- only outbound conn is up      intf -- interface
  <- -- only inbound  conn is up      tlsp -- transmit LSP
  Up -- operational                   rlsp -- receive LSP
Connection/Circuit                  Type    St  Time last up     # Up trans
FR-to-Ether                         if-sw   Up  Dec 30 09:57:23           1
  so-0/0/0.600                      intf    Up
  fe-1/0/0.0                        intf    Up
```

Configuring Layer 2.5 VPNs

Layer 2.5 VPNs are an extension to Layer 2 VPNs. The main difference is that Layer 2.5 VPNs are not required to have the same media on both ends of the connection. In general, Layer 2.5 VPNs combine the capabilities of TCC with Layer 2 VPNs.

Layer 2.5 VPNs support the same media types as TCC: ATM, Cisco HDLC, Ethernet, extended VLAN, Frame Relay, and PPP. They support IPv4, IS-IS, and MPLS traffic types, but do not support IPv6 traffic.

Although not covered in this manual, you can also use TCC in conjunction with Layer 2 circuits to connect two CE sites that use different Layer 2 protocols. You can use these so-called Layer 2.5 circuits as an alternative to Layer 2.5 VPNs. For more information about Layer 2 circuits, see “Layer 2 Circuits” on page 285 or the *JUNOS VPNs Configuration Guide*.

To configure Layer 2.5 VPNs, complete the following tasks:

- Configuring the Encapsulation on Interfaces Participating in the Layer 2.5 VPN on page 486
- Configuring the Layer 2.5 VPN on page 487
- Option: Configuring ISO or MPLS Traffic on T Series and M320 Routers on page 487

Configuring the Encapsulation on Interfaces Participating in the Layer 2.5 VPN

The encapsulation types used for Layer 2.5 VPNs are parallel to CCC encapsulations and identical to the TCC encapsulations explained in "Configuring TCC Interface Switching" on page 476. The encapsulation types are:

- atm-tcc-vc-mux
- atm-tcc-snap
- cisco-hdlc-tcc
- ethernet-tcc
- extended-vlan-tcc
- frame-relay-tcc
- ppp-tcc

When you configure a TCC encapsulation type, some modifications are needed to handle VPN connections over unlike Layer 2 links and to terminate the Layer 2 protocol locally. The router performs the following media-specific changes:

- PPP TCC—Both Link Control Protocol (LCP) and Network Control Protocol (NCP) are terminated on the router. Internet Protocol Control Protocol (IPCP) IP address negotiation is not supported. The JUNOS Software strips all PPP encapsulation data from incoming frames before forwarding them. For frames destined to a PPP-connected neighbor, PPP encapsulation is added.
- Cisco HDLC TCC—Keepalive processing is terminated on the router. The JUNOS Software strips all Cisco HDLC encapsulation data from incoming frames before forwarding them. Cisco-HDLC encapsulation is added onto frames that are destined to a Cisco HDLC-connected neighbor.
- Frame Relay TCC—All Local Management Interface (LMI) processing is terminated on the router. The JUNOS Software strips all Frame Relay encapsulation data from incoming frames before forwarding them. For frames destined to a Frame Relay-connected neighbor, Frame Relay encapsulation is added.
- ATM—Operation, Administration, and Maintenance (OAM) and Interim Local Management Interface (ILMI) processing is terminated at the router. Cell relay is not supported. The JUNOS Software strips all ATM encapsulation data from incoming frames before forwarding them. ATM encapsulation is added onto frames that are destined to an ATM-connected neighbor.

Configuring the Layer 2.5 VPN

Layer 2.5 VPNs use essentially the same configuration format as Layer 2 VPNs. You should already have experience configuring Layer 2 VPNs before you attempt to configure a Layer 2.5 VPN. The major steps required to create a Layer 2 VPN are:

- Enable MPLS on interfaces pointing toward the core and the edge on your provider edge (PE) and provider core (P) routers.
- Configure Label Distribution Protocol (LDP) on all P and PE routers for traffic traveling from the PEs, through the core, and to the remote PEs.
- Establish an internal BGP (IBGP) Layer 2 VPN peering relationship between PE routers.
- Set up policies on your PE routers that will set a private community tag on outbound BGP traffic heading to the remote PEs and accept incoming traffic that matches similar community traffic from the remote PEs.
- Build VPN routing and forwarding (VRF) instances on your PE routers and apply the previously configured policies to deliver private traffic to the customer edge (CE) routers.
- In addition to the above configurations procedures, you can specify a Layer 3 protocol family for a Layer 2.5 VPN on T Series and M320 routers. See the "Option: Configuring ISO or MPLS Traffic on T Series and M320 Routers" on page 487 section for configuration information.

Option: Configuring ISO or MPLS Traffic on T Series and M320 Routers

Layer 2.5 VPNs on T Series and M320 routers support IPv4, IS-IS, and MPLS traffic types. Layer 2.5 VPNs running on M Series routers support only IPv4 traffic. Layer 2.5 VPNs do not support IPv6.

By default, IPv4 traffic runs on T Series and M320 routers. To configure IS-IS (ISO traffic) or MPLS traffic on Layer 2.5 VPNs, you must configure both ends of the VPN with the protocol configuration.

The same type of Layer 3 traffic must be configured on both ends of a TCC connection. By default, TCC connections carry IPv4 traffic. To specify which traffic can run over a TCC interface, include the [inet | mpls | iso] statement at the [edit interfaces *interface-name* encapsulation *encapsulation-type* logical-unit-number family tcc protocol] hierarchy level:

```
[edit]
interfaces {
   interface so-2/0/0  {
      encapsulation ppp-tcc;
         unit 0  {
         family tcc {
            protocols [iso | inet | mpls];
         }
      }
   }
```

```
}
```

When enabling ISO over a Layer 2.5 VPN that is configured on a CE Ethernet interface, include the [point-to-point] statement at the [edit protocols isis interface *interface-name*] hierarchy level:

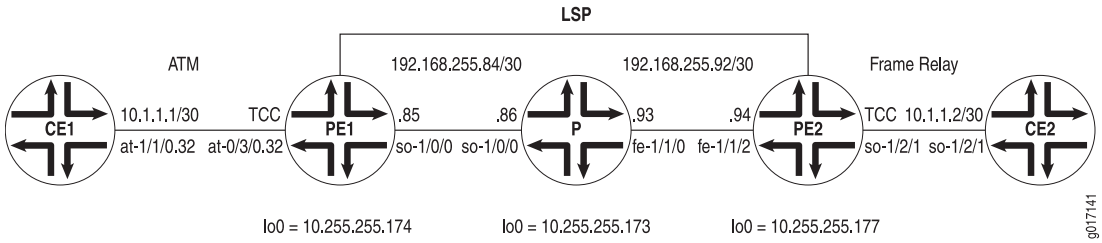
```
[edit]
protocols {
   isis {
      interface fe-1/0/0.0 {
         point-to-point;
      }
   }
}
```

For a full explanation of Layer 2 VPNs and configuration samples, see the *JUNOS VPNs Configuration Guide*.

Example: Layer 2.5 VPN Configuration

This section contains a configuration example and commands you can issue to verify your Layer 2.5 VPN configuration:

Figure 41: Layer 2.5 VPN Topology Diagram



In Figure 41 on page 488, ATM is configured between CE1 and PE1 and Frame Relay is configured between PE2 and CE2. To begin the Layer 2 VPN configuration, enable ATM and the corresponding encapsulation on CE1.

Router CE1

```
[edit]
interfaces
at-1/1/0 {
   description "to PE1 at-0/3/0";
   atm-options {
      vpi 0 maximum-vcs 2000;
   }
   unit 32 {
      vci 32;
      encapsulation atm-vc-mux;
      family inet {
         address 10.1.1.1/30;
      }
   }
}
```

The first provider edge (PE1) router uses ATM TCC encapsulation on the ATM VC connecting to CE1. After this, standard Layer 2 VPN design rules apply. You use MPLS on interfaces pointing toward the core and the edge, establish a Layer 2 VPN BGP peer relationship with PE2, use LDP or Resource Reservation Protocol (RSVP) for traffic traveling through the core, and configure the proper VRF instance. Finally, you create policies for PE1 that will set a private community tag on outbound BGP traffic heading to PE2 and accept incoming traffic that matches similar community traffic from PE2.

Router PE1

```
[edit]
interfaces {
    at-0/3/0 {
        description "to CE1 at-1/1/0";
        atm-options {
            vpi 0 maximum-vcs 2000;
        }
        unit 32 {
            encapsulation atm-tcc-vc-mux;
            vci 32;
        }
    }
    so-1/0/0 {
        description "to P so-1/0/0";
        unit 0 {
            family inet {
                address 192.168.255.86/30;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.255.174/32;
            }
        }
    }
}
protocols {
    mpls {
        interface at-0/3/0.32;
        interface so-1/0/0.0;
    }
    bgp {
        group my-internal-peers {
            type internal;
            local-address 10.255.255.174;
            family l2vpn {
                signaling;
            }
            neighbor 10.255.255.177;
        }
    }
    ldp {
        interface so-1/0/0.0;
    }
```

```
}
policy-options {
   policy-statement companyA-import {
      term T1 {
         from {
            protocol bgp;
            community companyA;
         }
         then accept;
      }
      term Final {
         then reject;
      }
   }
   policy-statement companyA-export {
      term T1 {
         then {
            community add companyA;
            accept;
         }
      }
      term Final {
         then reject;
      }
   }
   community companyA members target:100:1;
}
routing-instances {
   companyA {
      instance-type l2vpn;
      interface at-0/3/0.32;
      route-distinguisher 10.255.255.174:1;
      vrf-import companyA-import;
      vrf-export companyA-export;
      protocols {
         l2vpn {
            encapsulation-type interworking;
            site Denver {
               site-identifier 1;
               interface at-0/3/0.32 {
                  remote-site-id 2;
               }
            }
         }
      }
   }
}
```

On the provider core router (P), you need only enable MPLS and LDP on the interfaces that bridge the gap between the PE routers.

Router P

```
[edit]
interfaces {
   so-1/0/0 {
      description "to PE1 so-1/0/0";
```

```
        unit 0 {
            family inet {
                address 192.168.255.85/30;
            }
            family mpls;
        }
    }
    fe-1/1/0 {
        description "to PE2 fe-1/1/2";
        unit 0 {
            family inet {
                address 192.168.255.93/30;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.255.173/32;
            }
        }
    }
}
protocols {
    mpls {
        interface so-1/0/0.0;
        interface fe-1/1/0.0;
    }
    ldp {
        interface so-1/0/0.0;
        interface fe-1/1/0.0;
    }
}
```

The PE2 router uses Frame Relay TCC encapsulation on the Frame Relay DLCI connecting to CE2. To establish the Layer 2.5 VPN, follow the same steps you used to configure PE1. You use MPLS on interfaces pointing toward the core and the edge, establish a Layer 2 VPN BGP peer relationship with PE1, use LDP or RSVP for traffic traveling through the core, and configure the proper VRF instance. Finally, you create policies on PE2 that will set a private community tag on outbound BGP traffic heading to PE1 and accept incoming traffic that matches similar community traffic from PE1.

Router PE2

```
[edit]
interfaces {
    fe-1/1/2 {
        description "to P fe-1/1/0";
        unit 0 {
            family inet {
                address 192.168.255.94/30;
            }
            family mpls;
        }
    }
    so-1/2/1 {
```

```
        description "to CE2 so-1/2/1";
        dce;
        encapsulation frame-relay-tcc;
        unit 600 {
            encapsulation frame-relay-tcc;
            dlci 600;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.255.177/32;
            }
        }
    }
}
protocols {
    mpls {
        interface fe-1/1/2.0;
        interface so-1/2/1.600;
    }
    bgp {
        group my-internal-peers {
            type internal;
            local-address 10.255.255.177;
            family l2vpn {
                signaling;
            }
            neighbor 10.255.255.174;
        }
    }
}
ldp {
    interface fe-1/1/2.0;
}
policy-options {
    policy-statement companyA-import {
        term T1 {
            from {
                protocol bgp;
                community companyA;
            }
            then accept;
        }
        term Final {
            then reject;
        }
    }
    policy-statement companyA-export {
        term T1 {
            then {
                community add companyA;
                accept;
            }
        }
        term Final {
```

```
            then reject;
        }
    }
    community companyA members target:100:1;
}
routing-instances {
    companyA {
        instance-type l2vpn;
        interface so-1/2/1.600;
        route-distinguisher 10.255.255.177:1;
        vrf-import companyA-import;
        vrf-export companyA-export;
        protocols {
            l2vpn {
                encapsulation-type interworking;
                site NewYork {
                    site-identifier 2;
                    interface so-1/2/1.600 {
                        remote-site-id 1;
                    }
                }
            }
        }
    }
}
```

To complete the Layer 2.5 VPN configuration, enable Frame Relay encapsulation on CE2.

Router CE2

```
[edit]
interfaces
so-1/2/1 {
    description "to PE2 so-1/2/1";
    encapsulation frame-relay;
    unit 600 {
        dlci 600;
        family inet {
            address 10.1.1.2/30;
        }
    }
}
```

Verifying Your Work

To verify the operational status of your Layer 2.5 VPN, use the following commands:

- show route forwarding-table
- show ldp database
- show l2vpn connections
- show bgp summary
- show route

To view sample output of these commands as used with the configuration example, see the following:

- Router PE1 Status on page 494
- Router PE2 Status on page 495
- Router P Status on page 497

Router PE1 Status

```
user@PE1> show route forwarding-table
<snip>
Routing table:: ccc
MPLS:
Interface.Label    Type RtRef Nexthop           Type Index NhRef Netif
default            perm     0                  dscd    10     1
0                  user     0                  recv    12     2
1                  user     0                  recv    12     2
100128             user     0                  Pop           so-1/0/0.0
100128(S=0)        user     0                  Pop           so-1/0/0.0
100129             user     0                  Swap  100000  so-1/0/0.0
800001             user     0                  ucst   137     1 at-0/3/0.32
at-0/3/0. (CCC)    user     0                  indr   133     2
                                               Push 800000, Push 100000(top)
so-1/0/0.0
<snip>

user@PE1> show ldp database
Input label database, 10.255.255.174:0-10.255.255.173:0
  Label     Prefix
 100002     10.255.255.174/32
 100000     10.255.255.177/32
      3     10.255.255.173/32
Output label database, 10.255.255.174:0-10.255.255.173:0
  Label     Prefix
 100128     10.255.255.173/32
 100129     10.255.255.177/32
      3     10.255.255.174/32

user@PE1> show l2vpn connections
L2VPN Connections:
Legend for connection status (St)   Legend for interface status
OR -- out of range                  up -- operational
EI -- encapsulation invalid         Dn -- down
EM -- encapsulation mismatch        NP -- no present
CN -- circuit not present           DS -- disabled
OL -- no outgoing label             WE -- wrong encapsulation
Dn -- down                          UN -- uninitialized
VC-Dn -- Virtual circuit down
WE -- intf encaps != instance encaps
-> -- only outbound conn is up
<- -- only inbound  conn is up
UP -- operational
XX -- unknown
Instance: companyA
Local site: Denver (1)
    connection-site           Type  St     Time last up          # Up trans
    2                         rmt   Up     Nov 30 08:21:07 2001           1
      Local interface: at-0/3/0.32, Status: Up, Encapsulation: INTERWORKING
```

```
      Remote PE: 10.255.255.177
      Incoming label: 800001, Outgoing label: 800000

user@PE1>  show bgp summary
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 0          0          0          0          0          0
bgp.l2vpn.0            1          1          0          0          0          0
Peer               AS      InPkt     OutPkt    OutQ   Flaps Last
Up/DwnState|#Active/Received/Damped...
10.255.255.177      69         49         45       0       1       19:16 Establ
  bgp.l2vpn.0: 1/1/0
  companyA.l2vpn.0: 1/1/0

user@PE1> show route
<snip>
mpls.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
0                  *[MPLS/0] 1d 18:54:24, metric 1
                      Receive
1                  *[MPLS/0] 1d 18:54:24, metric 1
                      Receive
100128             *[LDP/9] 00:24:03, metric 1
                    > via so-1/0/0.0, Pop
100128(S=0)        *[LDP/9] 00:24:03, metric 1
                    > via so-1/0/0.0, Pop
100129             *[LDP/9] 00:24:03, metric 1
                    > via so-1/0/0.0, Swap 100000
800001             *[L2VPN/7] 00:10:35
                    > via at-0/3/0.32, Pop      [0]
at-0/3/0.32        *[L2VPN/7] 00:10:35
                    > via so-1/0/0.0, Push 800000, Push 100000(top)
companyA.l2vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
1:1:1:1        /96
                   *[L2VPN/7] 00:19:55
                      Discard
1:1:2:1        /96
                   *[BGP/170] 00:06:46, localpref 100, from 10.255.255.177
                      AS path: I
                    > via so-1/0/0.0, Push 100000
bgp.l2vpn.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
1:1:2:1        /96
                   *[BGP/170] 00:10:35, localpref 100, from 10.255.255.177
                      AS path: I
                    > via so-1/0/0.0, Push 100000
<snip>
```

Router PE2 Status

```
user@vpn07> show route forwarding-table
<snip>
Routing table:: ccc
MPLS:
Interface.Label    Type RtRef Nexthop           Type Index NhRef Netif
default            perm     0                   dscd     8     1
0                  user     0                   recv    10     2
1                  user     0                   recv    10     2
```

```
100002              user     0                    Pop           fe-1/1/2.0
100002(S=0)         user     0                    Pop           fe-1/1/2.0
100003              user     0                    Swap  100002  fe-1/1/2.0
800000              user     0                    ucst    60     1 so-1/2/1.0
so-1/2/1. (CCC)     user     0                    indr    59     2
<snip>

user@vpn07>  show ldp database
Input label database, 10.255.255.177:0-10.255.255.173:0
  Label     Prefix
 100000     10.255.255.177/32
      3     10.255.255.173/32
 100002     10.255.255.174/32
Output label database, 10.255.255.177:0-10.255.255.173:0
  Label     Prefix
 100002     10.255.255.173/32
      3     10.255.255.177/32
 100003     10.255.255.174/32

user@vpn07>  show l2vpn connections
L2VPN Connections:
Legend for connection status (St)   Legend for interface status
OR -- out of range                  up -- operational
EI -- encapsulation invalid         Dn -- down
EM -- encapsulation mismatch        NP -- no present
CN -- circuit not present           DS -- disabled
OL -- no outgoing label             WE -- wrong encapsulation
Dn -- down                          UN -- uninitialized
VC-Dn -- Virtual circuit down
WE -- intf encaps != instance encaps
-> -- only outbound conn is up
<- -- only inbound  conn is up
UP -- operational
XX -- unknown
Instance: companyA
Local site: NewYork (2)
    connection-site           Type  St     Time last up          # Up trans
    1                         rmt   Up     Nov 30 08:21:01 2001           1
      Local interface: so-1/2/1.0, Status: Up, Encapsulation: INTERWORKING
      Remote PE: 10.255.255.174
      Incoming label: 800000, Outgoing label: 800001

user@vpn07>  show bgp summary
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l2vpn.0            1          1          0          0          0          0
inet.0                 0          0          0          0          0          0
Peer               AS      InPkt     OutPkt    OutQ   Flaps Last
Up/DwnState|#Active/Received/Damped...
10.255.255.174      69         45         52       0       0       20:20 Establ
  bgp.l2vpn.0: 1/1/0
  companyA.l2vpn.0: 1/1/0

user@vpn07>  show route
<snip>
mpls.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
0                  *[MPLS/0] 02:34:04, metric 1
                      Receive
1                  *[MPLS/0] 02:34:04, metric 1
                      Receive
```

```
100002             *[LDP/9] 00:25:39, metric 1
                    > via fe-1/1/2.0, Pop
100002(S=0)        *[LDP/9] 00:25:39, metric 1
                    > via fe-1/1/2.0, Pop
100003             *[LDP/9] 00:25:01, metric 1
                    > via fe-1/1/2.0, Swap 100002
800000             *[L2VPN/7] 00:07:50
                    > via so-1/2/1.0, Pop      [0]
so-1/2/1.0         *[L2VPN/7] 00:07:50
                    > via fe-1/1/2.0, Push 800001, Push 100002(top)
companyA.l2vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
1:1:1:1        /96
                   *[BGP/170] 00:04:59, localpref 100, from 10.255.255.174
                     AS path: I
                    > via fe-1/1/2.0, Push 100002
1:1:2:1        /96
                   *[L2VPN/7] 00:11:34
                     Discard
bgp.l2vpn.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
1:1:1:1        /96
                   *[BGP/170] 00:11:38, localpref 100, from 10.255.255.174
                     AS path: I
                    > via fe-1/1/2.0, Push 100002
<snip>
```

Router P Status

```
user@P> show ldp database
Input label database, 10.255.255.173:0-10.255.255.174:0
  Label     Prefix
 100128     10.255.255.173/32
 100129     10.255.255.177/32
      3     10.255.255.174/32
Output label database, 10.255.255.173:0-10.255.255.174:0
  Label     Prefix
      3     10.255.255.173/32
 100000     10.255.255.177/32
 100002     10.255.255.174/32
Input label database, 10.255.255.173:0-10.255.255.177:0
  Label     Prefix
      3     10.255.255.177/32
 100002     10.255.255.173/32
 100003     10.255.255.174/32
Output label database, 10.255.255.173:0-10.255.255.177:0
  Label     Prefix
      3     10.255.255.173/32
 100000     10.255.255.177/32
 100002     10.255.255.174/32
```

For More Information

For additional information about TCC or Layer 2.5 VPNs, see the following documents:

- *JUNOS VPNs Configuration Guide*

- *JUNOS MPLS Applications Configuration Guide*
- *JUNOS Network Interfaces Configuration Guide*

Revision History

5 October 2009—10.0R1 Release. Roy Spencer.

13 June 2009—9.6R1 Release. Roy Spencer.

13 April 2009—9.5R1 Release. Roy Spencer.

15 January 2009—9.4R1 Release. Roy Spencer.

10 October 2008—9.3R1 Release. Roy Spencer.

10 July 2008—9.2R1 Release. Roy Spencer.

10 April 2008—9.1R1 Release. Roy Spencer.

1 February 2008—9.0R1 Release. Fawn Damitio.

29 June 2007—Added support for MX Series routers. 8.4R1 Release. Fawn Damitio.

27 March 2007—Added support for MPLS and ISIS traffic over Layer 2.5 VPN. 8.3R1 Release. Fawn Damitio.

12 January 2007—Added support for M120 router and MX960 router. 8.2R1 Release. Fawn Damitio.

15 September 2006—8.1R1 Release. Richard Hendricks.

29 June 2006—8.0R1 Release. Richard Hendricks.

27 March 2006—7.6R1 Release. Richard Hendricks.

9 January 2006—7.5R1 Release. Richard Hendricks.

14 September 2005—7.4R1 Release. Richard Hendricks.

13 June 2005—7.3R1 Release. Richard Hendricks.

5 April 2005—7.2R1 Release. Richard Hendricks.

2 February 2005—7.1R1 Release. Richard Hendricks.

6 October 2004—7.0R1 Release. Richard Hendricks.

6 July 2004—6.4R1 Release. Richard Hendricks.

5 April 2004—6.3R1 Release. Richard Hendricks.

22 December 2003—6.2R1 Release. Richard Hendricks.

22 September 2003—Mentioned Layer 2.5 circuits, 6.1R1 Release. Richard Hendricks.

30 June 2003—Added T Series support for TCC and Layer 2.5 VPNs, 6.0R1 Release. Elizabeth Lichtenberg.

2 April 2003—5.7R1 Release. Richard Hendricks.

27 December 2002—Added proxy ARP, 5.6R1 Release. Richard Hendricks.

30 September 2002—5.5R1 Release. Richard Hendricks.

19 July 2002—5.4R1 Release. Richard Hendricks.

6 June 2002—Document reformatted, added Ethernet TCC. Richard Hendricks.

21 November 2001—Initial document created. Gary Matthews.

Chapter 13

Virtual Private LAN Service

This chapter covers the following topics:

- Virtual Private LAN Service Overview on page 502
- Virtual Private LAN Service System Requirements on page 505
- Virtual Private LAN Service Terms and Acronyms on page 506
- Configuring VPLS on page 507
- Configuring Routing Protocols on the PE and Core Routers on page 508
- Configuring VPLS Encapsulation on CE-Facing Interfaces on page 508
- Configuring LDP Signaling for VPLS on page 510
- Configuring a VPLS Instance with BGP Signaling on page 511
- Configuring Interworking Between BGP Signaling and LDP Signaling in VPLS Instances on page 512
- Configuring Multihoming on a VPLS Border Router on page 514
- Option: Selecting an LSP for the VPLS Routing Instance to Traverse on page 515
- Option: Configuring VPLS Multihoming with BGP Signaling on page 516
- Option: Configuring VPLS Traffic Flooding over a Point-to-Multipoint LSP on page 519
- Option: Configuring Automatic Site Selection on page 521
- Option: Configuring VPLS to Use LSI Interfaces on page 522
- Option: Configuring Tunnel Services on MX Series Routers on page 523
- Configuring Integrated Routing and Bridging in a VPLS Instance (MX Series Routers Only) on page 523
- Configuring VLAN IDs in a VPLS Instance (MX Series Routers Only) on page 524
- Defining a VPLS Firewall Policer on page 525
- Defining a VPLS Firewall Filter on page 526
- Restricting Broadcast Packets in VPLS on page 527
- Option: Enabling VPLS Class of Service on page 528
- Option: Enabling VPLS Graceful Restart on page 528
- Configuring the VPLS MAC Address Timeout on page 529
- Option: Configuring VPLS Interinstance Bridging and Routing on page 530
- Option: Selecting Interfaces to Process VPLS Traffic on page 531

- Option: Limiting the Number of MAC Addresses Learned on a VPLS Interface on page 532
- Option: Optimizing VPLS Traffic Flows on page 533
- Option: Aggregated Interfaces for VPLS on page 533
- Configuration Steps for VPLS Graceful Routing Engine Switchover and Nonstop Active Routing on page 534
- Configuring VPLS Graceful Routing Engine Switchover and Nonstop Active Routing on page 534
- Synchronizing the Routing Engine Configuration on page 535
- Verifying VPLS Nonstop Active Routing Operation on page 536
- Tracing VPLS Nonstop Active Routing Synchronization Events on page 536
- Option: Configuring the Spanning Tree Protocol and VPLS on MX Series Routers on page 536
- Filtering Layer 2 Packets in a VPLS Instance (MX Series Routers Only) on page 537
- VPLS Configuration Examples on page 537
- Example: VPLS Configuration (BGP Signaling) on page 538
- Example: VPLS Configuration (BGP and LDP Interworking) on page 549
- Example: Configuring VPLS Nonstop Active Routing on page 563
- Example: Configuring Inter-AS VPLS with MAC Processing at the ASBR on page 564
- For More Information on page 591
- Revision History on page 591

Virtual Private LAN Service Overview

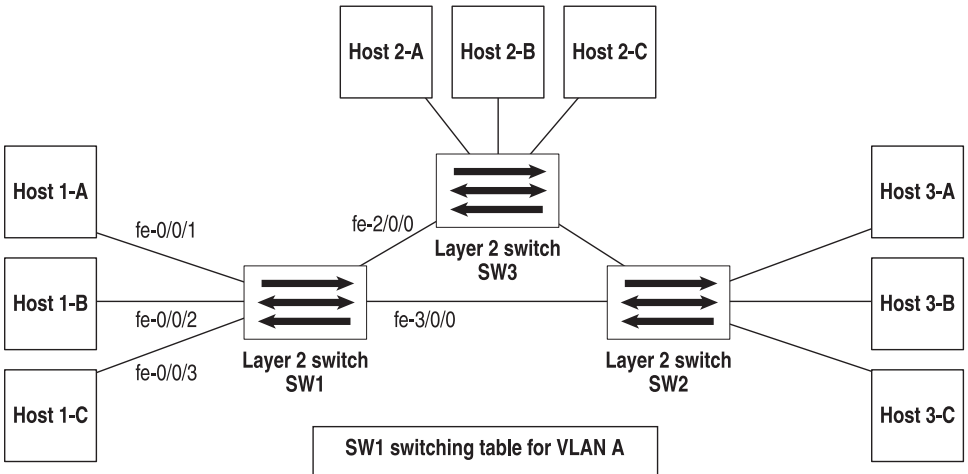
Ethernet is an increasingly important component of a service provider's slate of service offerings. Many customers are requesting the ability to connect local area network (LAN) locations across the country and around the world. To fulfill customer desire, service providers have had to set up complex point-to-point Layer 2 virtual private networks (VPNs) or connect expensive Layer 2 switches to handle traffic.

Virtual private LAN service (VPLS) meets the growing Ethernet needs of service providers and their customers. VPLS is an Ethernet-based multipoint-to-multipoint Layer 2 VPN. With VPLS, multiple Ethernet LAN sites can be connected to each other across an MPLS backbone. To the customer, all sites interconnected by VPLS appear to be on the same Ethernet LAN (even though traffic travels across a service provider network).

This guide explains the background knowledge you need to understand VPLS and provides detailed steps for you to follow to implement it in your network.

Before VPLS, the only way you could connect Ethernet LAN sites together was to set up a non-VPLS Layer 2 VPN or install multiple Layer 2 Ethernet switches. Figure 42 on page 503 shows how three switches can be connected to each other.

Figure 42: Ethernet Switching Example



SW1 switching table for VLAN A

Port	MAC Address
fe-0/0/1	1a1a.1a1a.1a1a
fe-2/0/0	2a2a.2a2a.2a2a
fe-3/0/0	3a3a.3a3a.3a3a

SW1 switching table for VLAN B

Port	MAC Address
fe-0/0/2	1b1b.1b1b.1b1b
fe-2/0/0	2b2b.2b2b.2b2b
fe-3/0/0	3b3b.3b3b.3b3b

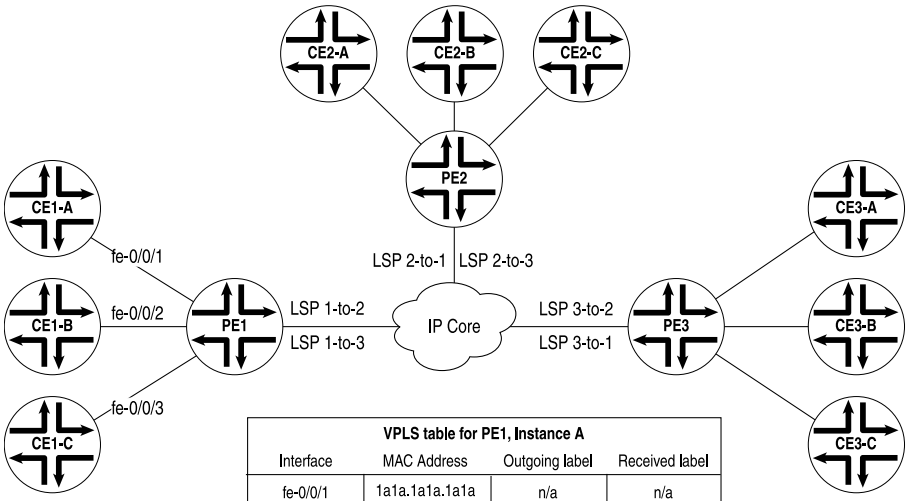
SW1 switching table for VLAN C

Port	MAC Address
fe-0/0/3	1c1c.1c1c.1c1c
fe-2/0/0	2c2c.2c2c.2c2c
fe-3/0/0	3c3c.3c3c.3c3c

A typical switch builds its Layer 2 switching table with MAC address and interface information learned from traffic received from other switches. If a switch does not know how to reach a particular destination, it floods traffic for that destination to all ports except the one where the traffic originated. When information about a previously unknown destination is received, this information is added to the switching table. If a destination is known, the switch sends the traffic directly to the intended recipient through the associated port listed in the switching table.

Figure 43 on page 504 shows a VPLS network comparable to the switch example and explains how VPLS functions similarly to Ethernet switches (assuming a Spanning Tree Protocol is configured).

Figure 43: VPLS Introductory Example



VPLS table for PE1, Instance A

Interface	MAC Address	Outgoing label	Received label
fe-0/0/1	1a1a.1a1a.1a1a	n/a	n/a
vt-0/3/0.32770	2a2a.2a2a.2a2a	800000	800002
vt-0/3/0.32773	3a3a.3a3a.3a3a	800012	800014

VPLS table for PE1, Instance B

Interface	MAC Address	Outgoing label	Received label
fe-0/0/2	1b1b.1b1b.1b1b	n/a	n/a
vt-0/3/0.32771	2b2b.2b2b.2b2b	800004	800006
vt-0/3/0.32774	3b3b.3b3b.3b3b	800016	800018

VPLS table for PE1, Instance C

Interface	MAC Address	Outgoing label	Received label
fe-0/0/3	1c1c.1c1c.1c1c	n/a	n/a
vt-0/3/0.32772	2c2c.2c2c.2c2c	800008	800010
vt-0/3/0.32775	3c3c.3c3c.3c3c	800020	800022

Notice that Layer 2 information gathered by a switch (for example, MAC addresses and interface ports) is included in the VPLS instance table. However, instead of requiring all VPLS interfaces to be physical switch ports, the router allows remote traffic for a VPLS instance to be delivered across an MPLS label-switched path (LSP) and arrive on a virtual port. The virtual port emulates a local, physical port. Traffic can be learned, forwarded, or flooded to the virtual port almost identically to the way traffic is sent to a local port.

The VPLS table learns MAC address and interface information for both physical and virtual ports. If no activity is seen for a particular MAC address, it is purged from the table over time.

As shown in Figure 43 on page 504, the main difference between a physical port and a virtual port is that the router captures additional information from a virtual port—an outgoing MPLS label used to reach the remote site and an incoming MPLS label for VPLS traffic received from the remote site.

When you configure VPLS on a routing platform, a virtual port is generated as a logical interface on a virtual loopback tunnel (vt) interface or a label-switched interface (LSI). On Juniper Networks M Series Multiservice Edge Routers and Juniper Networks T Series Core Routers, virtual ports are created dynamically on vt interfaces if you install a Physical Interface Card (PIC) that supports virtual tunnels. With VPLS, you must install at least one Tunnel Services, Link Services, or Adaptive Services PIC in each VPLS provider edge (PE) router. On Juniper Networks MX Series Ethernet Services

Routers, virtual ports are created dynamically on vt interfaces if you configure tunnel services on one of the four Packet Forwarding Engines (PFEs) included in each DPC. If your routing platform does not offer tunnel services through a PIC or PFE, you can configure VPLS to create virtual ports on LSI logical interfaces.

One property of flooding behavior in VPLS is that traffic received from remote PE routers is never forwarded to other PE routers. This restriction helps prevent loops in the core network. If a customer edge (CE) Ethernet switch has two connections or more to the same PE router, you must enable the Spanning Tree Protocol to prevent loops. For more information on configuring the Spanning Tree Protocol, see the *JUNOS Routing Protocols Configuration Guide*.

The paths carrying VPLS traffic between each PE router participating in a routing instance are called pseudowires. The pseudowires are signaled using either BGP or LDP.

Virtual Private LAN Service System Requirements

To implement VPLS, your system must meet these minimum requirements:

- JUNOS Release 9.1 or later for nonstop active routing (NSR), VPLS ping on M120, M320, and MX Series routers, and automatic site selection for BGP-signaled VPLS.
- JUNOS Release 9.0 or later for Virtual Spanning Tree Protocol (VSTP) support, 802.1p classification in Bridged Ethernet over ATM mode support, interworking between LDP and BGP signaling in a VPLS instance, and Layer 2 VPLS filters for MX960 routers.
- JUNOS Release 8.4 or later for VPLS with LDP signaling. Also, integrated routing and bridging (IRB) is supported starting in this release.
- JUNOS Release 8.3 or later for point-to-multipoint LSP support on VPLS.
- JUNOS Release 8.2 or later for VPLS support on MX Series routing platform, VPLS graceful Routing Engine switchover (GRES) support, and VPLS support on Gigabit Ethernet IQ2 aggregated Ethernet interfaces.
- JUNOS Release 7.6 or later for VPLS support on LSI logical interfaces.
- JUNOS Release 7.5 or later for multihoming a CE router to multiple PE routers.
- JUNOS Release 7.3 or later for VPLS per-packet load balancing, support for limiting MAC address learning per interface in a VPLS domain, and migration to the VPLS and Layer 2 VPN signaling statement at the [edit protocols bgp groups *group-name* family l2vpn] hierarchy level.
- JUNOS Release 6.4 or later to implement Ethernet VPLS over ATM LLC interface encapsulation on T Series and M320 routers, to select the tunnel-enabled PICs that provide virtual ports for VPLS operation, and to issue the show vpls statistics command.
- JUNOS Release 6.3 or later to clear MAC addresses from the VPLS table and to modify VPLS table timeout intervals.
- JUNOS Release 6.2 or later for VPLS class of service (CoS), VPLS graceful restart, VPLS interinstance bridging and routing, VPLS source and destination MAC address accounting, VPLS virtual port support on the Adaptive Services PIC for M Series routers, and general VPLS support for T Series and M320 routers.

- JUNOS Release 6.1 or later for VPLS policers and filters.
- JUNOS Release 6.0 or later for Ethernet VPLS over ATM LLC interface encapsulation on M Series routers.
- JUNOS Release 5.7 or later for VPLS with BGP signaling and Ethernet VPLS, VLAN VPLS, and extended VLAN VPLS interface encapsulations.
- Two Juniper Networks M Series (except the M160 router), MX Series, T Series, or TX Matrix routing platforms for the provider edge (PE).
- On M Series and T Series routers, one Adaptive Services PIC, Link Services PIC, or Tunnel Services PIC per routing platform to create VPLS virtual ports on vt interfaces.
- On M Series and T Series routers, one Fast Ethernet or Gigabit Ethernet PIC per routing platform (from this list):
  - 4-port Fast Ethernet PIC with 10/100 BASE-TX interfaces
  - 1-port, 2-port, or 10-port Gigabit Ethernet PIC
  - 4-port, quad-wide Gigabit Ethernet PIC
  - 1- and 2-port Gigabit Ethernet Intelligent Queuing (IQ) PIC
  - 4- and 8-port Gigabit Ethernet IQ2 PIC with small form-factor pluggable transceivers (SFPs)
  - 1-, 2-, and 4-port Gigabit Ethernet PIC with SFPs
  - 1-port 10-Gigabit Ethernet PIC

Virtual Private LAN Service Terms and Acronyms

V

virtual port A special logical interface that is generated dynamically when you configure VPLS on a PE router. Virtual ports send and receive VPLS traffic for remote PE routers as if the remote VPLS sites had Ethernet-based interfaces directly connected to the local PE router. To generate virtual ports, VPLS PE routing platforms use logical interfaces on a vt interface (that is generated by the Tunnel Services PIC, Link Services PIC, Adaptive Services PIC, an LSI interface, or a tunnel services interface configured on MX Series routers).

virtual private LAN service (VPLS) An Ethernet-based multipoint-to-multipoint Layer 2 VPN service used for interconnecting multiple Ethernet LANs across an MPLS backbone. BGP-based VPLS is based on the Internet Engineering Task Force (IETF) Internet draft draft-ietf-l2vpn-vpls-bgp-08.txt, *Virtual Private LAN Service (VPLS) Using BGP for Auto-discovery and Signaling* (expires December 2006). LDP-based VPLS is specified in the IETF draft *Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling*. For more information about VPLS, see the *JUNOS VPNs Configuration Guide*.

Configuring VPLS

The following sections show the configuration steps necessary to implement VPLS:

- Configuring Routing Protocols on the PE and Core Routers on page 508
- Configuring VPLS Encapsulation on CE-Facing Interfaces on page 508
- Configuring LDP Signaling for VPLS on page 510
- Configuring a VPLS Instance with BGP Signaling on page 511
- Configuring Interworking Between BGP Signaling and LDP Signaling in VPLS Instances on page 512
- Configuring Multihoming on a VPLS Border Router on page 514
- Option: Selecting an LSP for the VPLS Routing Instance to Traverse on page 515
- Option: Configuring VPLS Multihoming with BGP Signaling on page 516
- Option: Configuring VPLS Traffic Flooding over a Point-to-Multipoint LSP on page 519
- Option: Configuring Automatic Site Selection on page 521
- Option: Configuring VPLS to Use LSI Interfaces on page 522
- Option: Configuring Tunnel Services on MX Series Routers on page 523
- Configuring Integrated Routing and Bridging in a VPLS Instance (MX Series Routers Only) on page 523
- Configuring VLAN IDs in a VPLS Instance (MX Series Routers Only) on page 524
- Defining a VPLS Firewall Policer on page 525
- Defining a VPLS Firewall Filter on page 526
- Restricting Broadcast Packets in VPLS on page 527
- Option: Enabling VPLS Graceful Restart on page 528
- Configuring the VPLS MAC Address Timeout on page 529
- Option: Configuring VPLS Interinstance Bridging and Routing on page 530
- Option: Selecting Interfaces to Process VPLS Traffic on page 531
- Option: Limiting the Number of MAC Addresses Learned on a VPLS Interface on page 532
- Option: Optimizing VPLS Traffic Flows on page 533
- Option: Aggregated Interfaces for VPLS on page 533
- Configuring VPLS Graceful Routing Engine Switchover and Nonstop Active Routing on page 534

Configuring Routing Protocols on the PE and Core Routers

At a fundamental level, VPLS is a type of Layer 2 VPN. All forms of Layer 2 VPNs require that you configure network protocols to handle the following:

- *intradomain routing* — An interior gateway protocol (IGP) such as Open Shortest Path First (OSPF) or Intermediate System-to-Intermediate System (IS-IS)
- *interdomain routing* — Border Gateway Protocol (BGP)
- *label switching* — Multiprotocol Label Switching (MPLS)
- *path signaling* — Resource Reservation Protocol (RSVP) or Label Distribution Protocol (LDP)

For more information about these protocols and examples of how to configure these protocols to support a Layer 2 VPN, see the *JUNOS VPNs Configuration Guide*.

NOTE: The 8-port, 12-port, and 48-port dense Fast Ethernet Physical Interface Cards (PICs) cannot push more than two labels onto an MPLS packet. Because of this, we do not recommend that you configure these PICs as core-facing or equivalent interfaces.

Configuring VPLS Encapsulation on CE-Facing Interfaces

There are five types of VPLS interface encapsulation: Ethernet VPLS, Ethernet VPLS over ATM LLC, VLAN VPLS, extended VLAN VPLS, and flexible Ethernet services. When one of these encapsulations is applied to an interface, a family type of VPLS is enabled by default. The encapsulation types are:

- ether-vpls-over-atm-llc—Use Ethernet VPLS over ATM LLC encapsulation on ATM2 IQ logical interfaces. Use this encapsulation type to support IEEE 802.1p classification binding on ATM VCs. This encapsulation type enables a VPLS instance to support bridging between Ethernet interfaces and ATM interfaces, as described in RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*. When you use this encapsulation type, you configure it on logical interfaces only and you cannot configure multipoint interfaces.
- extended-vlan-vpls—Use extended VLAN VPLS encapsulation on Ethernet interfaces that have VLAN 802.1Q tagging and VPLS enabled and that must accept packets carrying TPIDs 0x8100, 0x9100, and 0x9901.

NOTE: The built-in Gigabit Ethernet PIC on the M7i router does not support MPLS.

- ethernet-vpls—Use Ethernet VPLS encapsulation on Ethernet interfaces that have VPLS enabled and must accept packets carrying standard Tag Protocol ID (TPID) values.
- vlan-vpls—Use VLAN VPLS encapsulation on Ethernet interfaces with VLAN tagging enabled. VLAN VPLS encapsulation supports TPID 0x8100 only. You must

configure this encapsulation type on both the physical interface and the logical interface.

- flexible-ethernet-services—Use flexible Ethernet services encapsulation when you want to configure multiple per-unit Ethernet encapsulations. This encapsulation type allows you to configure any combination of route, TCC, CCC, and VPLS encapsulations on a single physical port. Aggregated Ethernet bundles cannot use this encapsulation type.

Use the following guidelines to configure a VPLS interface:

- For encapsulation type vlan-vpls, VLAN IDs 1 through 511 are reserved for normal Ethernet VLANs, IDs 512 through 1023 are reserved for VPLS VLANs on Fast Ethernet interfaces, and IDs 512 through 4094 are reserved for VPLS VLANs on Gigabit Ethernet interfaces.
- For encapsulation type extended-vlan-vpls, all VLAN IDs from 1 through 1023 are valid for VPLS VLANs on Fast Ethernet interfaces, and all VLAN IDs from 1 through 4094 are valid for VPLS VLANs on Gigabit Ethernet interfaces. VLAN ID 0 is reserved for priority tagging. For encapsulation type flexible-ethernet-services, VLAN IDs from 1 through 511 are no longer reserved for normal VLANs.
- For encapsulation type flexible-ethernet-services, VLAN IDs from 1 through 511 are no longer reserved for normal VLANs.

VPLS Interface Encapsulation for an Ethernet Interface

To configure VPLS interface encapsulation for an Ethernet interface, include the encapsulation statement at the [edit interfaces *interface-fpc/pic/port*] hierarchy level and select ethernet-vpls, vlan-vpls, extended-vlan-vpls, flexible-ethernet-services or ether-vpls-over-atm-llc as the encapsulation type. If you select the VLAN VPLS encapsulation, also include the vlan-vpls statement at the [edit interfaces *ethernet-interface-fpc/pic/port* unit *unit-number* encapsulation] logical interface hierarchy level. When using either VLAN VPLS or extended VLAN VPLS encapsulations, include the vlan-tagging statement at the [edit interfaces *ethernet-interface-fpc/pic/port*] hierarchy level.

VPLS Interface Encapsulation for an ATM2 IQ Interface

To configure VPLS interface encapsulation for an ATM2 IQ interface, include the encapsulation statement at the [edit interfaces at-*fpc/pic/port*] hierarchy level and select ether-vpls-over-atm-llc as the encapsulation type. To configure VPLS interface encapsulation for a Gigabit Ethernet IQ interface or Gigabit Ethernet PICs with small form-factor pluggable transceivers (SFPs), include the encapsulation statement at the [edit interfaces ge-*fe/pic/port*] hierarchy level and select flexible-ethernet-services as the encapsulation type.

```
[edit]
interfaces {
   ge-0/1/0 {
      vlan-tagging;
      encapsulation vlan-vpls;
      unit 0 {
         encapsulation vlan-vpls;
```

```
            vlan-id 600;
        }
    }
    at-0/2/0 {
        encapsulation ether-vpls-over-atm-llc;
    }
}
```

Configuring LDP Signaling for VPLS

Like other Layer 2 VPNs, you must enable a VPLS instance to isolate VPLS traffic from other network traffic. An important element of a VPLS instance is the signaling protocol. You can configure BGP signaling, LDP signaling, or both BGP and LDP signaling in a VPLS instance.

To configure LDP signaling, you must first enable a VPLS instance to isolate VPLS traffic from other network traffic. To enable a VPLS instance, include the instance-type vpls statement at the [edit routing-instances *instance-name*] hierarchy level. To configure LDP signaling within the instance, identify the virtual circuit with the vpls-id statement and specify the PE routers participating in the instance with the neighbor statement:

```
[edit]
routing-instances {
    instance-name {
        instance-type vpls;
        interface ge-0/1/0.0;
        protocols {
            vpls {
                vpls-id id-name;
                neighbor neighbor-id; # The neighbor-id should be the loopback address of
                    # the remote PE router.
            }
        }
    }
}
```

To fully enable LDP signaling on a PE in a VPLS instance, you must also enable LDP on the loopback interface of the router. To enable LDP on the loopback interface, include the interface lo0.0 statement at the [edit protocols ldp] hierarchy level:

```
[edit]
protocols {
    ldp {
        interface lo0.0;
    }
}
```

For LDP signaling within a VPLS routing instance, the JUNOS Software supports the following values only:

- FEC—FEC 128
- Control bit—0
- Ethernet pseudowire type—hexadecimal 0x0005

Configuring a VPLS Instance with BGP Signaling

Like other Layer 2 VPNs, you must enable a VPLS instance to isolate VPLS traffic from other network traffic. An important element of a VPLS instance is the signaling protocol. You can configure BGP signaling, LDP signaling, or both BGP and LDP signaling in a VPLS instance.

You must enable a VPLS instance to isolate VPLS traffic from other network traffic. To enable a VPLS instance, include the instance-type vpls statement at the [edit routing-instances *instance-name*] hierarchy level.

Within the instance, you can define the maximum number of sites that can participate in this VPLS instance, a local site name, and a local site identifier. To configure the maximum number of sites, include the site-range statement at the [edit routing-instances *instance-name* protocols vpls] hierarchy level. The maximum number of sites is 65,535.

NOTE: The site ID must be less than the site range. If you specify a site ID that is greater than the site range, your connections do not come up, even though the commit operation succeeds.

To configure a site name, include the site statement at the [edit routing-instances *instance-name* protocols vpls] hierarchy level. To configure the site ID, include the site-identifier statement at the [edit routing-instances *instance-name* protocols vpls site *name*] hierarchy level.

```
[edit]
routing-instances;
   instance-name {
      instance-type vpls;
      interface ge-0/1/0.0;
      route-distinguisher 10.245.14.218:1;
      vrf-target target:11111:1;
      protocols {
         vpls {
            site-range 10;
            site greenPE1 {
               site-identifier 1;
            }
         }
      }
   }
```

To complete the configuration, you must configure the Layer 2 VPN family for BGP by including the signaling statement at the [edit protocols bgp family l2vpn] hierarchy level:

```
[edit]
protocols {
   bgp {
      family l2vpn;
```

```
            signaling;
        }
    }
}
```

Configuring Interworking Between BGP Signaling and LDP Signaling in VPLS Instances

If you want to configure a VPLS instance with both BGP and LDP-signaled pseudowires, you must configure a VPLS border router. Without a VPLS border router, LDP-signaled PEs and BGP-signaled PEs will be unaware of one another and the VPLS instance will not be fully meshed.

NOTE: Interworking between BGP signaling and LDP signaling in VPLS instances is supported only on MX Series and M320 routers.

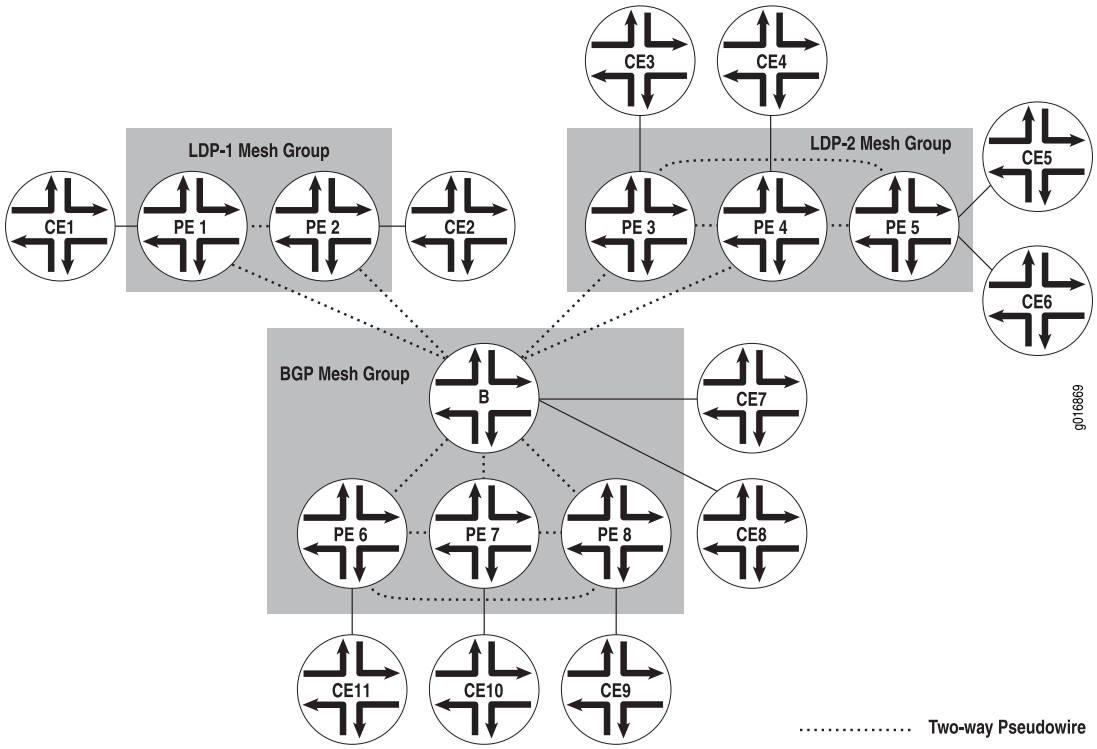
To enable interworking between BGP-signaled PE routers and LDP-signaled PE routers, you configure a border router to interconnect both sets of routers within the same VPLS routing instance. You also need to configure mesh groups on the border router to group the sets of PE routers that are fully meshed and which share the same signaling protocol, either BGP or LDP. You can configure multiple mesh groups to map each fully meshed LDP-signaled or BGP-signaled VPLS domain to a mesh group. In the data plane, the border router maintains a common MAC table used to forward traffic between the LDP-signaled and BGP-signaled mesh groups. When forwarding any VPLS traffic received over a PE router pseudowire, the border router ensures that traffic is not forwarded back to the PE routers, which are in same mesh group as the originating PE router.

There is always just one BGP mesh group in a VPLS instance, and it is created automatically when you configure BGP signaling for that instance. You can configure one or more LDP mesh groups. MX Series routers support up to 15 PE mesh groups (including the default BGP mesh group), and M Series and T Series routers support up to 127 PE mesh groups (including the default BGP mesh group).

In Figure 44 on page 513, Routers PE1 and PE2 are in the LDP-signaled mesh group “LDP-1”. Routers PE3, PE4, and PE5 are in the LDP-signaled mesh group “LDP-2”. Routers PE6, PE7, and PE8 are in the BGP-signaled mesh group. The border router acts as a traditional PE (by connecting to CEs) in addition to being a border router. Every router shown in the topology in Figure 44 on page 513 is in the same VPLS instance, bgp-ldp-mesh1.

When Router CE1 sends a frame whose destination MAC address is CE9, PE1 receives the frame and performs a MAC address lookup. The MAC address is not in the PE1 MAC table and so PE1 floods the frame to the other PEs in the LDP-1 mesh group (PE2) and also to Router B, which from the perspective of PE1, are the only members of the VPLS network. When Router B receives the data from PE1, it does not find the MAC address in its MAC table and so it floods the frame to PE3, PE4, PE5, PE6, PE7, and PE8, but not back to PE1 or PE2. The PE routers then perform a MAC table lookup and flood the data to their CE routers.

Figure 44: Topology for BGP/LDP Interworking in a VPLS Instance



In this topology, you configure routers PE6, PE7, and PE8 as you traditionally configure BGP-signaled VPLS routers. You configure routers PE1, PE2, PE3, PE4, and PE5 as you traditionally configure LDP-signaled VPLS routers. In addition, you create the mesh group LDP-1 for Routers PE1 and PE2 and mesh group LDP-2 for Routers PE3, PE4, and PE5 by including the mesh-group *mesh-group-name* statement at the [edit routing-instances *routing-instance-name* protocols vpls] hierarchy level.

NOTE: The border router can act as a normal PE in addition to being a border router and can support local CE-facing interfaces.

In the example below, the interfaces are CE interfaces. In this case, the router is acting as both a border router and a regular PE router.

To enable interworking between VPLS mesh groups, configure the border router by including the site *site-name* statement at the [edit routing instances *routing-instance-name* protocols] hierarchy level:

```
[edit]
routing-instances {
   bgp-ldp-mesh1 {
      instance-type vpls;
      route-distinguisher 10.245.14.218:1;
      interface fe-1/3/1.0;
      interface fe-1/3/2.0;
   vrf-target target:10:100;
```

```
}
protocols {
   vpls {
   site green {
      site-identifier 1;
   }
}
```

Configure LDP signaling with the vpls-id and neighbor *neighbor-id* statements. You can configure mesh groups LDP-1 and LDP-2 by including the mesh-group statement at the [edit routing instances *routing-instance-name* protocols vpls *vpls-id* and including the neighbor *neighbor-id* statement at the [edit routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name*] hierarchy level:

```
[edit routing-instances bgp-ldp-mesh1 protocols vpls]
vpls-id 100;
mesh-group LDP-1 {
   neighbor 10.1.1.1;
   neighbor 20.1.1.1;
}
mesh-group LDP-2 {
   neighbor 30.1.1.1;
   neighbor 40.1.1.1;
   neighbor 10.1.1.1;
}
```

NOTE: When you configure BGP signaling to interoperate with LDP signaling in a VPLS network, the following features are not supported:

- Point-to-multipoint VPLS
- Integrated routing and bridging

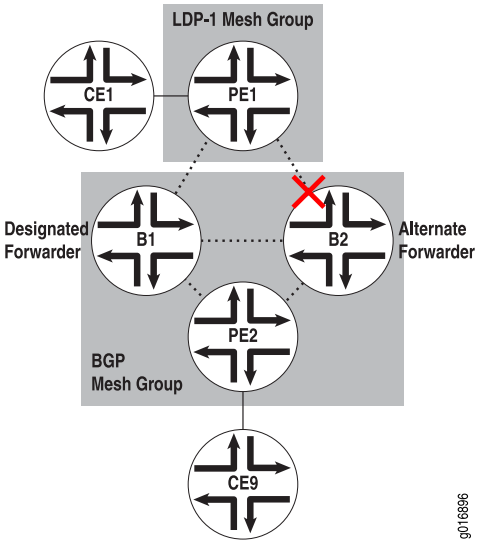
Configuring Multihoming on a VPLS Border Router

Configuring multihoming on VPLS border routers ensures that if one border router is unreachable, BGP/LDP PE connectivity is maintained through the other VPLS border router. With multihoming, one border router is chosen as the designated forwarder for each mesh group. The designated forwarder is chosen through either the BGP or VPLS path-selection procedure. If the designated forwarder loses connectivity with a mesh group, the alternate border router then takes over as designated forwarder for that mesh group. A VPLS instance must be configured with BGP signaling in order for multihoming to work.

Figure 45 on page 515 shows a simplified example of how multihoming works with VPLS border routers. In this example, B1 is the designated forwarder and B2 is the alternate forwarder. If CE1 wanted to send data to CE9, the data would travel from CE1 to PE1, which is part of the LDP-1 mesh group. PE1 would then flood the data to B1 (the designated forwarder), which would forward the data to PE2. It would not send the data to Router B2. PE2 would then send the data to its destination, CE9. If B1 lost connectivity with the LDP-1 mesh group, then B2 would become the

designated forwarder. In this case, PE1 would send the data through B2, not through B1.

Figure 45: Multihoming for Border Area Routers



You configure multihoming on border routers by including the site-identifier and multi-homing statements at the [edit routing-instances *routing-instance-name* protocols] hierarchy level. The designated forwarder and alternate forwarder must be configured with the same site identifier.

Router B1

```
[edit routing-instances example protocols]
vpls {
    site mult-home-ldp-1 {
        site-identifier 1;
        mesh-group ldp-1;
        multi-homing;
    }
```

Router B2

```
[edit routing-instances example protocols]
vpls {
    site mult-home-ldp-1 {
        site-identifier 1;
        mesh-group ldp-1;
        multi-homing;
    }
```

For more information on multihoming, see “Option: Configuring VPLS Multihoming with BGP Signaling” on page 516.

Option: Selecting an LSP for the VPLS Routing Instance to Traverse

If you have two or more equal-cost-path LSPs between your VPLS PE router sites, you can select an LSP over which the VPLS traffic will travel. To select an LSP for VPLS traffic, assign the VPLS instance to a BGP community, define a policy that

directs community traffic over a specified LSP, and then apply the policy to the forwarding table.

To configure a BGP community, include the community *community-name* statement at the [edit policy-options] hierarchy level. Be sure to specify the vrf-export or vrf-target values from the VPLS routing instance as community identifiers with the members *community-ids* statement at the [edit policy-options community *community-name*] hierarchy level.

To create a policy that sends community traffic over a specific LSP, include the community *community-name* statement at the [edit policy-options policy-statement *policy-name* term *term-name* from] hierarchy level and the install-nexthop lsp *lsp-name* statement at the [edit policy-options policy-statement *policy-name* term *term-name* then] hierarchy level. To apply the policy to the forwarding table, include the export *policy-name* statement at the [edit routing-options forwarding-table] hierarchy level.

```
[edit]
routing-options {
    autonomous-system 69;
    forwarding-table {
        export LSP-policy;
    }
    policy-options {
        policy-statement LSP-policy {
            term a {
                from community gold;
                then {
                    install-nexthop lsp pe1-to-pe2;
                    accept;
                }
            }
        }
        community gold members target:11111:1;
    }
}
```

Option: Configuring VPLS Multihoming with BGP Signaling

With VPLS multihoming, you can connect multiple PE router interfaces to one customer site. This feature provides VPLS redundancy should a PE router or PE router interface fail.

To configure multihoming, you must configure the same site IDs on all PE routers and router interfaces that are connected to the same customer site, You must also specify on each PE router which interfaces are connected to the customer site. We recommend that you configure distinct route distinguishers for each multihomed router. Configuring distinct route distinguishers helps with faster convergence when the connection to a primary router goes down. It also requires that the other PE routers maintain additional state information.

To configure a route distinguisher, include the route-distinguisher statement at the [edit routing-instances *instance-name*] hierarchy level. To assign a site ID, include the site-identifier statement at the [edit routing-instances *instance-name* protocols vpls site

name] hierarchy level. To specify the interfaces associated with a site, include the interface statement at the [edit routing-instances *instance-name* protocols vpls site *name*] hierarchy level.

To connect multiple PE routers to one customer site, you must configure multihoming on each PE router connected to that site. This will prevent routing loops should BGP connectivity fail. BGP automatically determines the primary and backup routers. Alternatively, you can statically configure a primary PE router and backup PE routers for a customer site by specifying the preference value. BGP uses preference values to determine routing paths.

NOTE: Multihoming relies on full BGP connectivity to all other PEs. Configure a dual router reflector topology to provide redundant PE-to-PE BGP connectivity.

To configure multihoming, include the multi-homing statement at the [edit routing-instances *instance-name* protocols vpls site *name*] hierarchy level. To configure preference value, include the preference-value statement at the [edit routing-instances *instance-name* protocols vpls *site name*] hierarchy level. You can configure the preference value as primary or backup, or you can specify a preference number. When specifying preference numbers, configure the primary interface with a preference value of 65,535 and any backup interfaces with a number from 1 to 65,534.

When multiple PE router interfaces on a single PE router are connected to one customer site, you must configure an active interface. All traffic will pass through the active interface unless this interface fails, in which case a backup interface will become the active interface.

To specify a multihomed interface as the primary interface for a site, include the active-interface statement at the [edit routing-instances *instance-name* protocols vpls site *name*] hierarchy level. The interface that you specify is called the primary interface. If the primary interface goes down, an alternate interface becomes the active interface. Once the primary interface comes back up, the primary interface becomes the active interface once again and the alternate interface becomes inactive.

If you do not want to specify a primary multihomed interface, you can use the any option. With the any option, the router dynamically chooses an active interface. If the active interface goes down, an alternate interface becomes the active interface. Once the down interface comes back up, it stays inactive.

If no active interfaces are configured at the site level, it is assumed that all traffic for a VPLS site travels through a single, nonmultihomed PE router.

NOTE: If you add a direct connection between CE devices that are multihomed to the same VPLS site on different PE routers, traffic loops and loss of connectivity might occur. We do not recommend this topology.

The following example shows a multihoming configuration with two PE routers that are connected to a single customer site. Note in the configuration that PE1 is the primary router and PE2 is the backup router.

Router PE1

```
[edit]
routing-instances {
   green {
      instance-type vpls;
      interface fe-0/1/3.0;
      route-distinguisher 10.255.14.218:1;
      vrf-target target:11111:1;
      protocols {
         vpls {
            site-range 10;
            site green4 {
               site-identifier 4;
               multi-homing; # Ensures that BGP is established before forwarding on the
                  # site member interfaces.
               preference value 65535;
               interface fe-1/1/3.0;
            }
         }
      }
   }
}
```

Router PE2

```
[edit]
routing-instances {
   green {
      instance-type vpls;
      interface fe-0/1/0.0;
      route-distinguisher 10.255.14.219:1;
      vrf-target target:11111:1;
      protocols {
         vpls {
            site-range 10;
            site green4 {
               site-identifier 4;
               multi-homing;
               preference value 1;
               interface fe-0/1/0.0;
            }
         }
      }
   }
}
```

The following example shows a multihoming configuration with one PE router with multiple interfaces that are connected to a single customer site.

Router PE3

```
[edit]
routing-instances {
   green {
      instance-type vpls;
      interface fe-1/1/0.0;
      interface fe-1/2/0.0;
      interface fe-1/3/0.0;
      route-distinguisher 10.255.14.218:1;
      vrf-target target:11111:1;
```

```
protocols {
   vpls {
      site-range 10;
      site green4 {
         site-identifier 4;
         active-interface any;
         interface fe-1/1/0.0;
         interface fe-1/2/0.0;
         interface fe-1/3/0.0;
      }
   }
}
}
}
```

For more information on VPLS multihoming, see the *JUNOS VPNs Configuration Guide*.

Option: Configuring VPLS Traffic Flooding over a Point-to-Multipoint LSP

In each VPLS routing instance, you can configure a dedicated point-to-multipoint LSP to carry all unknown unicast, broadcast, and multicast traffic. Enabling this feature increases the efficiency of your network, because duplicate copies of flooded traffic do not have to be created for each PE router in the VPLS routing instance. Figure 46 on page 519 shows how flooded traffic reaches PE routers in a VPLS routing instance when a point-to-multipoint LSP is not configured for flooding. Figure 47 on page 520 shows an example of a VPLS routing instance configured with point-to-multipoint LSP flooding.

NOTE: You cannot configure point-to-multipoint LSP flooding if your VPLS network is configured for interoperability between BGP and LDP signaling.

Figure 46: Traditional Flooding in a VPLS Routing Instance

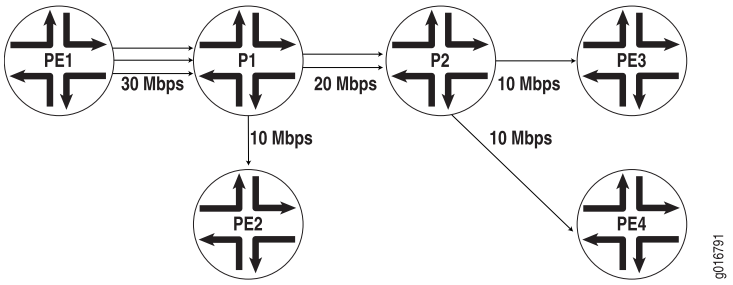
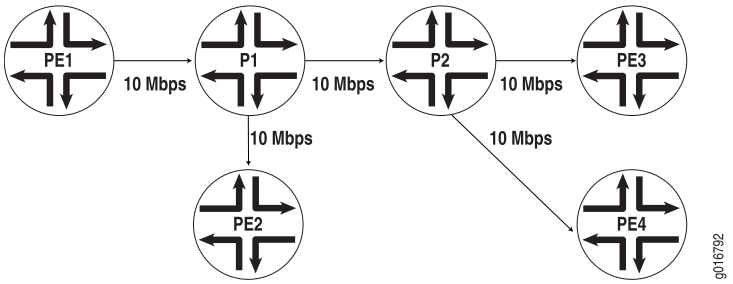


Figure 47: VPLS Routing Instance with Point-to-Multipoint LSP Flooding



You have three options when configuring a point-to-multipoint LSP for flooding:

- Static point-to-multipoint LSP—Configure this option to control which path each PE sub-LSP takes. When using this option, ensure that all PEs within the VPLS routing instance are part of the static point-to-multipoint LSP. When you add PEs to the VPLS routing instance, you must configure a sub-LSP for the new PE and add the sub-LSP to the static point-to-multipoint LSP. To configure a static point-to-multipoint LSP, include the label-switched-path *path-name* statement at the [edit protocols mpls] hierarchy level.
- Dynamic point-to-multipoint LSP with a preconfigured template—Configure this option to create a dynamic point-to-multipoint LSP with specific parameters such as link protection and optimized time. With this option, newly added PEs are automatically added to the point-to-multipoint LSP. To use the preconfigured template, include the template statement at the [edit protocols mpls label-switch-path *path-name*] hierarchy level.
- Dynamic point-to-multipoint LSP with a default template—Configure this option to automatically create a dynamic point-to-multipoint LSP with default parameters. With this option, newly added PEs are automatically added to the point-to-multipoint LSP. To use a default template, include the default-template statement at the [edit routing-instances *routing-instance-name* provider-tunnel rsvp-te label-switched-path-template] hierarchy level.

To define the parameters for a static point-to-multipoint LSP, include the label-switched-path *path-name* statement at the [edit protocols mpls] hierarchy level:

```
[edit]
protocols {
   mpls {
      label-switched-path vpls-bar-p2mp-s21_lsp_a {
         to 192.168.1.1
         p2mp vpls-bar-p2mp-lsp;
      }
      label-switched-path vpls-bar-p2mp-s21_lsp_b {
         to 192.168.1.2
         p2mp vpls-bar-p2mp-lsp;
      }
   }
}
```

To add a new PE router to the static point-to-multipoint LSP, include the label-switched-path *sub-path-name* statement at the [edit protocols mpls] hierarchy level:

```
[edit]
protocols {
    mpls {
        label-switched-path added-PE3 {
            to 1.1.1.1
            p2mp vpls-bar-p2mp-lsp;
        }
    }
}
```

For more information on configuring static and dynamic point-to-multipoint LSPs, see the *JUNOS MPLS Applications Configuration Guide*.

To enable this feature, configure either the static or label-switched-path-template options for the rsvp-te statement at the [edit routing-instance *routing-instance-name* provider-tunnel] hierarchy level:

```
[edit]
routing-instance foo {
    provider-tunnel {
        rsvp-te {
            static-lsp vpls-bar-p2mp-lsp;
        }
    }
}
```

To verify your work, enter the show vpls connection extensive command:

```
Router_1# show vpls connection extensive
....
status-vector: BF
connection-site Type St Time last up # Up trans
2 rmtUpJan 31 10:14:37 2007 1
Local interface: lsi.32768, Status: Up, Encapsulation: VPLS
Description: Intf -vpls VPLS-A local site 1 remote site 2
Remote PE: 10.255.164.2, Negotiated control-word: No
Incoming label: 262153, Outgoing label: 800000
RSVP-TE P2MP lsp:
Ingress branch LSP: 13:vpls:10.255.164.1:BPLS-A, State: Up
Egress branch LSP: 4:vpls:10.255.164.2:VPLS-A, Statue: Up
TimeEventInterface/Lb1/PE
Jan 31 10:14:37 2007 status update timer
Ingress RSVP-TE P2MP LSP: 11:vpls:10.255.164.1:VPLS-A, Flood next-hop ID: 476
```

Option: Configuring Automatic Site Selection

You can configure BGP-signaled VPLS instances to automatically specify the site IDs for the routers participating in the VPLS domain. Site IDs help to minimize label usage in VPLS instances with numerous PE routers.

The automatic-site-id statement includes the following options:

- startup-wait-time—Time to wait at startup to receive all VPLS information for configured route targets from other PE routers.
- new-site-wait-time—Time to wait to receive VPLS information from a newly configured routing instance or a new site. Effectively, it is the time to wait before a site makes an attempt to locate an unused site ID for its claim advertisement.
- collision-detect-time—Time to wait after issuing a claim advertisement before the PE router can start using the site ID if it does not receive a competing claim. If the PE router receives a competing claim within this time interval, it runs a collision resolution procedure. Explicitly configured site IDs always take precedence over automatically generated site IDs.
- reclaim-wait-time—Time to wait before attempting to claim a site ID after a collision. There are default values for all of these options, so they do not need to be explicitly configured.

To configure a VPLS automatic site ID, include the automatic-site-id statement at the [edit routing-instances *routing-instance-name* protocols vpls site *site-name*] hierarchy level:

```
[edit]
routing-instances {
    vpls instance 1 {
        protocols {
            vpls {
                site vpls instance 1 {
                    automatic-site-id;
                }
            }
        }
    }
}
```

Option: Configuring VPLS to Use LSI Interfaces

On M Series and T Series routers, VPLS uses tunnel-based PICs to create virtual ports on vt interfaces. If you do not have a tunnel-based PIC installed on your M Series or T Series router, you can still configure VPLS by using label-switched interfaces (LSIs) to support the virtual ports. Use of LSI interfaces requires the use of Ethernet-based PICs installed in an Enhanced FPC.

NOTE: On MX Series routers, when VPLS is used with an LSI interface for traffic flowing from the core to the egress CE, classification does not work on the egress PE.

To use LSI interfaces for VPLS instead of vt interfaces, include the no-tunnel-services statement at the [edit routing-instances *instance-name* protocols vpls] hierarchy level.

```
[edit routing-instances]
instance-name  {
```

```
    protocols {
        vpls {
            no-tunnel-services;
        }
    }
}
```

NOTE: The following interface types do not support the use of LSI interfaces with VPLS:

- Aggregated SONET/SDH interfaces (cannot be used as the core-facing interface)
- Channelized interfaces (cannot be used as the core-facing interface)
- ATM1 interfaces

Option: Configuring Tunnel Services on MX Series Routers

MX Series routers use Dense Port Concentrators (DPCs) with built-in physical ports, which means that you do not insert PICs on the router. Instead, you configure tunnel interfaces on one of the four Packet Forwarding Engines (PFEs) that are on each DPC.

To create tunnel interfaces on an MX Series router, include the tunnel-services statement at the [edit chassis fpc *slot-number* pic *number*] hierarchy level. To configure the bandwidth for a tunnel interface, include the bandwidth statement at the [edit chassis fpc *slot-number* pic *number*] hierarchy level.

The following example shows a tunnel interface with 1 Gbps of bandwidth configured on PFE 1 of the DPC installed in slot 4 of an MX Series router:

```
[edit chassis]
fpc 4;
pic 1 {
    tunnel services {
        bandwidth 1g;
    }
}
```

Once you have configured a tunnel interface on a PFE, you can treat this interface as a standard tunnel interface and proceed with a standard VPLS configuration. For more information, see the *JUNOS System Basics Configuration Guide*.

Configuring Integrated Routing and Bridging in a VPLS Instance (MX Series Routers Only)

Integrated routing and bridging (IRB) over VPLS cannot be used in conjunction with the vlan-id all statement. One or more Layer 2 logical interfaces must be configured inside the instance in order for IRB to function properly.

To configure IRB within a VPLS instance, include the routing-interface *irb-interface-name* statement at the [edit routing-instances *routing-instance-name* instance-type vpls] hierarchy level:

```
[edit]
routing-instances {
    marketing {
        instance-type vpls;
        route-distinguisher 11.11.11.11:10;
        vrf-target target:100:100;
        interface ae0.100;
        interface ae0.200;
        routing-interface irb.1234;
    }
}
```

Configuring VLAN IDs in a VPLS Instance (MX Series Routers Only)

You can configure VLAN identifiers for a VPLS instance in the following ways:

- By using the input-vlan-map and the output-vlan-map statements at the [edit interfaces] hierarchy level. For more information, see the *JUNOS Network Interfaces Configuration Guide* and *JUNOS Class of Service Configuration Guide*.
- By using the vlan-id or vlan-tags statements at the [edit routing-instances *routing-instance-name* instance-type vpls] hierarchy level.

The vlan-id and vlan-tags statements are used to perform the following functions:

- Translate, or normalize, the VLAN tags of received packets received into a learn VLAN identifier.
- Create multiple learning domains that each contain a VLAN identifier. A learning domain is a MAC address database to which MAC addresses are added based on the VLAN identifier.

For more information about how VLAN tags are processed and translated, see the *JUNOS MX Series Layer 2 Configuration Guide*.

To configure VLAN identifiers for a VPLS instance, include the vlan-id or vlan-tags statement at the [edit routing-instances *routing-instance-name* instance-type vpls] hierarchy level.

NOTE: You cannot configure VLAN mapping using the input-vlan-map and output-vlan-map statements if you configure a VLAN identifier for a VPLS instance using the vlan-id or vlan-tags statements.

```
[edit]
routing-instances {
    marketing {
        instance-type vpls;
        vlan-id 401;
```

```
        route-distinguisher 11.11.11.11:10;
        vrf-target target:100:100;
        interface ae0.100;
        interface ae0.200;
    }
}
```

Defining a VPLS Firewall Policer

You can configure filters, policers, and broadcast and unknown filters to determine which kind of traffic is allowed into and out of a VPLS domain. You can apply these filters and policers to CE-facing interfaces only.

To process traffic as it enters a VPLS domain, you can define a firewall policer and apply it to the input interface. To define policer characteristics for incoming VPLS traffic, include the bandwidth-limit and burst-size-limit statements at the [edit firewall policer *policer-name* if-exceeding] hierarchy level. Then, specify statements to implement the desired action (for example, discard) for the policed traffic at the [edit firewall policer *policer-name* then] hierarchy level. To apply the policer to a CE-facing interface, include the input or output statements and the name of the policer at the [edit interfaces *interface-name* unit *unit-number* family vpls policer] hierarchy level.

```
[edit]
interfaces {
    ge-2/1/0 {
        vlan-tagging;
        mtu 1544;
        encapsulation vlan-vpls;
        unit 0 {
            encapsulation vlan-vpls;
            vlan-id 600;
            family vpls {
                policer {
                    input vpls-policer;
                }
            }
        }
    }
}
firewall {
    policer {
        vpls-policer {
            if-exceeding {
                bandwidth-limit 5m;
                burst-size-limit 1m;
            }
            then discard;
        }
    }
}
```

Defining a VPLS Firewall Filter

You can configure filters, policers, and broadcast and unknown filters to determine which kind of traffic is allowed into and out of a VPLS domain. You can apply these filters and policers to CE-facing interfaces only.

To process traffic as it exits a VPLS domain, you can define a firewall filter and apply it to the output interface. To configure match conditions for a firewall filter, include the interface-group, source-mac-address, destination-mac-address, ethernet-type, or vlan-ethernet-type statements at the [edit firewall family vpls filter *filter-name* term *term-name* from] hierarchy level. Then, implement the desired action (for example, discard) for the traffic at the [edit firewall family vpls filter *filter-name* term *term-name* then] hierarchy level. To apply the filter to a CE-facing interface, include the input, output, or group statements at the [edit interfaces *interface-name* unit *unit-number* family vpls filter] hierarchy level.

```
[edit]
interfaces {
    fe-2/1/1 {
        vlan-tagging;
        mtu 1544;
        encapsulation vlan-vpls;
        unit 0 {
            encapsulation vlan-vpls;
            vlan-id 600;
            family vpls {
                filter {
                    output vpls-out-filter;
                }
            }
        }
    }
}
firewall {
    family vpls {
        filter vpls-out-filter {
            interface-specific;
            term 1 {
                from {
                    source-mac-address {
                        00.10.10.10.11.18/48;
                    }
                }
                then {
                    count count.ce2;
                    accept;
                }
            }
            term 2 {
                then accept;
            }
        }
    }
```

```
}
```

NOTE:

- Output filters do not work for broadcast, multicast, and unknown unicast traffic.
- If an IRB interface is configured as part of a VPLS routing instance, VPLS filters might not filter packets that are destined to the IRB interface. This can be configured by installing filters that match Layer 3 fields for the the IRB interface.
- If you apply a firewall filter to discard a source MAC address, the MAC address is not deleted from the MAC address table.

Restricting Broadcast Packets in VPLS

You can configure filters, policers, and broadcast and unknown filters to determine which kind of traffic is allowed into and out of a VPLS domain. You can apply these filters and policers to CE-facing interfaces only.

To restrict the flow of broadcast and unknown unicast packets into a VPLS domain, you must create a firewall filter and apply the filter to one of the forwarding tables of the VPLS routing instance. When you apply a filter in this way, the filter processes traffic from all interfaces in the instance, including vt interfaces. To configure match conditions for a VPLS-based firewall filter, include the source-mac-address, destination-mac-address, interface-group, ethernet-type, or vlan-ethernet-type statements at the [edit firewall family vpls filter *filter-name* term *term-name* from] hierarchy level. Then, specify statements to activate the desired action (for example, discard) for the matched packets at the [edit firewall family vpls filter *filter-name* term *term-name* then] hierarchy level.

To apply the filter to the broadcast and unknown unicast table of a VPLS routing instance, include the input statement and the name of the filter at the [edit routing-instances *instance-name* forwarding-options family vpls flood] hierarchy level. To apply the filter to the destination MAC address table of a VPLS routing instance, include the input statement and the name of the filter at the [edit routing-instances *instance-name* forwarding-options family vpls filter] hierarchy level.

```
[edit]
firewall {
   family vpls {
      filter vpls-flood {
         term 1 {
            from {
               destination-mac-address (broadcast | multicast | unknown-unicast) {
                  # The broadcast, multicast,
                  # and unknown-unicast options apply to MX Series
                  # routers only.
                  00.90.69.dc.95.3b/48;
               }
            }
            then discard;
         }
```

```
            term 2 {
                then accept;
            }
        }
    }
}
routing-instances {
    green {
        forwarding-options {
            family vpls {
                (flood | filter) {
                    input vpls-flood;
                }
            }
        }
    }
}
```

When you configure VPLS, a priority filter for Spanning Tree Protocol (STP) bridge protocol data units (BPDUs) is enabled by default. This BPDU filter matches on the well-known STP MAC address of 01:80:c2:00:00:00/24 and applies high priority to this traffic.

For more information on VPLS policers and filters, see the *JUNOS Policy Framework Configuration Guide* and the *JUNOS VPNs Configuration Guide.*

Option: Enabling VPLS Class of Service

For JUNOS Release 6.2 or later, you can configure class of service (CoS) for all interfaces in the VPLS domain. CoS information is sent across the MPLS backbone and is preserved for all VPLS traffic processed by local interfaces, virtual ports, and remote interfaces.

For more information on configuring CoS, see the *JUNOS Class of Service Configuration Guide*.

Option: Enabling VPLS Graceful Restart

VPLS graceful restart allows you to continue forwarding VPLS traffic across the core MPLS network even if one of the routers in the forwarding path restarts. Graceful restart for VPLS functions the same way as Layer 2 VPN graceful restart. To configure graceful restart for VPLS, include the graceful-restart statement at the [edit routing-options] hierarchy level on all PE and core routers.

```
[edit]
routing-options {
    graceful-restart;
}
```

For more information on graceful restart, see the *JUNOS High Availability Configuration Guide*.

Configuring the VPLS MAC Address Timeout

You can fine-tune your VPLS domain by clearing MAC address entries from the VPLS table or modifying the default timeout interval for the VPLS table.

NOTE: On MX Series routers running JUNOS Release 8.4 and later, you can set the expiration time of entries in the MAC table only for the entire router, not for specific VPLS routing instances. To set the expiration for the entire router, include the mac-table-aging-time seconds statement at the [edit protocols l2-learning] hierarchy level. Do not include the mac-table-aging-time statement at the [edit routing-instances routing-instance-name protocols vpls] hierarchy level on MX Series routers running JUNOS Release 8.4 and later.

To clear all MAC address entries from the VPLS table, issue the clear vpls mac-address command. Add the logical-system *logical-system-name* option to clear entries within a logical system and include the instance *instance-name* option to clear entries in a specific VPLS instance. Use the *mac-address* option to remove individual MAC addresses.

To configure the VPLS table timeout interval, include the mac-table-aging-time statement at the [edit routing-instances *instance-name* protocols vpls] hierarchy level. The default interval is 300 seconds, with a minimum of 10 seconds and a maximum of 1 million seconds. As a general rule, you can configure longer values for small, stable VPLS networks and shorter values for large, dynamic VPLS networks. If no traffic is received for a specific MAC address, M Series and T Series routers wait one additional interval before automatically clearing MAC address entries from the VPLS table. MX Series routers do not wait for this interval.

```
[edit]
routing-instances {
   instance-name {
      protocols {
         vpls {
            mac-table-aging-time seconds;
         }
      }
   }
}
```

Option: Configuring VPLS Interinstance Bridging and Routing

To deliver interinstance traffic between two or more VPLS instances, or between a VPLS instance and a Layer 3 VPN routing instance, you must use a logical tunnel interface. Originally designed to interconnect logical systems, the logical tunnel interface acts as a point-to-point connection between instances. A logical tunnel interface can be generated by a Tunnel Services PIC installed on an Enhanced FPC in your routing platform, an integrated Adaptive Services Module installed in an M7i router, or a tunnel services interface configured on MX Series routers. To configure a logical tunnel interface, include the lt-*fpc/pic*/0 statement at the [edit interfaces] hierarchy level. Keep in mind these rules when you connect instances:

- You need to configure both endpoints of the logical tunnel. Configure the first logical tunnel interface in the VPLS instance and the second within the instance you want to interconnect to the VPLS domain.
- Choose one of several interface encapsulation types for your logical tunnel interface peers. Your choices are Ethernet, Ethernet circuit cross-connect (CCC), Ethernet VPLS, Frame Relay, Frame Relay CCC, VLAN, VLAN CCC, and VLAN VPLS. Include one of these choices with the encapsulation statement at the [edit interfaces *lt-fpc/pic/*0 unit *unit-number*] hierarchy level.
- Depending on the encapsulation type you select, specify a corresponding data-link connection identifier (DLCI) number for Frame Relay or a VLAN identifier for VLAN encapsulations on your logical tunnel interface peers. To configure the DLCI or VLAN identifier, include the dlci or vlan-id statement at the [edit interfaces lt-*fpc/pic/*0 unit *unit-number*] hierarchy level.
- Your choice of protocol family for the logical tunnel interface also is determined by your selection of an encapsulation type. For Ethernet VPLS and VLAN VPLS, family vpls is assigned by default. For all other Ethernet and VLAN encapsulation types, include the mpls or inet statement at the [edit interfaces lt-*fpc/pic/*0 unit *unit-number* family] hierarchy level. For Frame Relay encapsulation types, you can configure any of the available protocol families: ccc, inet, inet6, iso, mpls, or tcc.
- Be sure to match the logical interface unit numbers of the peering logical tunnel interfaces. To configure, include the peer-unit statement at the [edit interfaces lt-*fpc/pic/*0 unit *unit-number*] hierarchy level.

```
[edit]
interfaces {
   lt-fpc/pic/0 {
      unit unit-number {
         encapsulation (ethernet | ethernet-ccc | ethernet-vpls | frame-relay |
            frame-relay-ccc | vlan | vlan-ccc | vlan-vpls);
         peer-unit number; # The logical unit number of the peering lt interface.
         dlci dlci-number;
         vlan-id vlan-number;
         family (ccc | inet | inet6 | iso | mpls | tcc);
      }
   }
}
routing-instances {
```

```
    vpls-instance-name {
        interface ge-fpc/pic/port.unit-number;
        interface lt-0/0/0.1;
        ...
        second-instance-name {
            interface at-fpc pic/port.unit-number;
            interface lt-0/0/0.2;
            ...
        }
    }
}
```

Option: Selecting Interfaces to Process VPLS Traffic

On M Series and T Series routers, the PICs that can create VPLS virtual ports dynamically from vt interfaces include the Tunnel Services PIC, the Link Services PIC, and the Adaptive Services PIC. On MX Series routers, logical tunnel interfaces configured by including the tunnel-services statement at the [edit chassis fpc *slot-number pic number*] hierarchy level can create VPLS virtual ports dynamically from vt interfaces.

By default, the JUNOS Software automatically and randomly selects vt interfaces to act as VPLS virtual ports in a round-robin fashion. However, if your routing platform contains two or more of these tunnel-enabled interfaces, you can manually select which interfaces process traffic for each VPLS domain.

You can select an interface to be the primary device responsible for VPLS traffic processing. You can also select a group of interfaces to share responsibility for VPLS traffic processing. When the primary interface is operating normally, it handles all VPLS-related tasks. If the primary device is not available, any interfaces included in the VPLS interface group assume responsibility.

To select an interface to be the primary device responsible for VPLS traffic processing, include the primary statement at the [edit routing-instances *instance-name* protocols vpls tunnel-services] hierarchy level. To select a group of interfaces to share responsibility for VPLS traffic processing, include the devices statement at the [edit routing-instances *instance-name* protocols vpls tunnel-services] hierarchy level.

```
[edit]
routing-instances {
    instance-name {
        protocols {
            vpls {
                tunnel-services {
                    devices [vt-0/0/0 vt-1/0/0 vt-2/0/0];
                    primary vt-0/0/0;
                }
            }
        }
    }
}
```

Option: Limiting the Number of MAC Addresses Learned on a VPLS Interface

There are three main levels where you can configure MAC address limits:

- interface-mac-limit—This statement allows you to specify a limit for MAC addresses at an interface level. For VPLS, you can include the interface-mac-limit statement at the [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls], [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls site *site-name* interfaces *interface-name*], [edit routing-instances *routing-instance-name* protocols vpls], or [edit routing-instances *routing-instance-name* protocols vpls site *site-name* interfaces *interface-name*] hierarchy level. For MX Series routers only, you can specify what the router does with additional MAC addresses once the MAC address limit is reached. The default behavior is for the router to flood the packet, but you can alternatively include the packet-action drop option to have the router drop the packets. The default MAC address table size for each interface is 1024 addresses.
- mac-table-size—This statement allows you to specify a limit for MAC addresses at a domain level. For VPLS, you can include the mac-table-size statement at the [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls] or [edit routing-instances *routing-instance-name* protocols vpls] hierarchy level. The default MAC address table size for each domain is 5120 addresses.
- global-mac-limit (MX Series routers only)—This statement allows you to specify a limit for MAC addresses for all interfaces and all domains for the entire router. You can include the global-mac-limit statement at the [edit protocols l2-learning] hierarchy level. The default MAC address table size for the entire system is 393,215 addresses.

NOTE: If you manually configure a MAC address limit, you must ensure that values for interface limits (such as the interface-mac-limit) are set lower than domain limits (such as mac-table-size), and the domain limits are set lower than global limits (such as global-mac-limit). If a value for a more specific limit is set higher than a more global limit, the commit operation fails.

The range of values for the interface-mac-limit statement is 16 through 65,536. The output of the show vpls statistics command displays the results of configuring interface-level MAC address limitations.

```
[edit]
routing-instances {
   instance-name {
      protocols {
         vpls {
            interface-mac-limit number;
            site site-name {
               interface interface-name {
                  interface-mac-limit number;
               }
            }
```

```
        }
      }
    }
  }
```

Option: Optimizing VPLS Traffic Flows

To improve the performance of VPLS traffic processing in your routing platform, you can implement the following features:

- To optimize VPLS traffic flows across multiple paths, you can enable per-packet load balancing. To enable per-packet load balancing, include the load-balance per-packet statement at the [edit policy-options policy-statement *policy-name* term *term-name* then] hierarchy level and apply the policy to the forwarding table with the export *policy-name* statement at the [edit routing-options forwarding-table] hierarchy level.
- To optimize hashing of source and destination MAC addresses within VPLS traffic flows, include the source-mac and destination-mac statements at the [edit forwarding-options hash-key family multiservice] hierarchy level.

For more information about load balancing and hash keys, see the *JUNOS Policy Framework Configuration Guide*.

Option: Aggregated Interfaces for VPLS

You can configure aggregated Ethernet interfaces between CE devices and PE routers for VPLS routing instances. Traffic is load-balanced across all of the links in the aggregated interface. If one or more links in the aggregated interface fails, the traffic is switched to the remaining links.

In the example below, 0 is the interface instance number that completes the link association. This number can be from 0 through 127, for a total of 128 aggregated interfaces. The VPLS encapsulation types supported on aggregated Ethernet interfaces are ethernet-vpls, vlan-vpls, or extended-vlan-vpls.

```
[edit]
interfaces ae0
vlan-tagging;
encapsulation vlan-vpls;
unit 0 {
   vlan-id 100;
}
```

The aggregated Ethernet interface must also be configured for a VPLS routing instance. Use the standard VPLS routing instance configuration on aggregated Ethernet interfaces.

For more information about how to configure aggregated Ethernet interfaces, see the *JUNOS Network Interfaces Configuration Guide*.

Configuration Steps for VPLS Graceful Routing Engine Switchover and Nonstop Active Routing

This section describe the following:

- Configuring VPLS Graceful Routing Engine Switchover and Nonstop Active Routing on page 534
- Synchronizing the Routing Engine Configuration on page 535
- Verifying VPLS Nonstop Active Routing Operation on page 536
- Tracing VPLS Nonstop Active Routing Synchronization Events on page 536

Configuring VPLS Graceful Routing Engine Switchover and Nonstop Active Routing

Nonstop active routing (NSR) enables a routing platform with redundant Routing Engines to switch from a primary Routing Engine to a backup Routing Engine without alerting peer nodes that a change has occurred. Nonstop active routing uses the same infrastructure as graceful Routing Engine switchover to preserve interface and kernel information. However, nonstop active routing also preserves routing information and protocol sessions by running the routing protocol process (rpd) on both Routing Engines. The logical interface, next-hop router, and both advertised and received labels are preserved. In addition, nonstop active routing preserves TCP connections maintained in the kernel.

Nonstop active routing requires you to configure graceful Routing Engine switchover (GRES). Graceful Routing Engine switchover (GRES) allows a routing platform with dual Routing Engines to switch over from a master Routing Engine to a backup Routing Engine without causing an interruption to packet forwarding. Graceful Routing Engine switchover is supported with VPLS, meaning that should the master Routing Engine go down, the router can switch to the backup Routing Engine without affecting VPLS traffic.

To configure graceful Routing Engine switchover, include the graceful-switchover statement at the [edit chassis redundancy] hierarchy level. To disable graceful Routing Engine switchover, remove the graceful-switchover statement at the [edit chassis redundancy] hierarchy level.

```
[edit]
chassis redundancy {
   graceful-switchover;
}
```

By default, nonstop active routing is disabled. To enable nonstop active routing, include the nonstop-routing statement at the [edit routing-options] hierarchy level:

```
[edit routing-options]
nonstop-routing;
```

To disable nonstop active routing, remove the nonstop-routing statement from the [edit routing-options] hierarchy level.

To enable the routing platform to switch over to the backup Routing Engine when the routing protocol process (rpd) fails rapidly three times in succession, include the other-routing-engine statement at the [edit system processes routing failover] hierarchy level.

The following example enables graceful Routing Engine switchover, nonstop active routing, and nonstop active routing trace options for VPLS.

```
[edit]
system commit {
   synchronize;
}
chassis {
   redundancy {
      graceful-switchover; # This enables graceful Routing Engine switchover on the
         # routing platform.
   }
}
routing-options {
   nonstop-routing; # This enables nonstop active routing on the routing platform.
   traceoptions {
      flag nsr-synchronization;
   }
}
```

For more information about the other-routing-engine statement, see the *JUNOS System Basics Configuration Guide*.

Synchronizing the Routing Engine Configuration

When you configure nonstop active routing, you must also include the commit synchronize statement at the [edit system] hierarchy level so that configuration changes are synchronized on both Routing Engines:

```
[edit system]
commit synchronize;
```

If you try to commit the nonstop active routing configuration without including the commit synchronize statement, the commit operation fails.

If you issue the commit synchronize command at the [edit] hierarchy level on the backup Routing Engine, the JUNOS system software displays a warning and commits the candidate configuration.

NOTE: A newly inserted backup Routing Engine automatically synchronizes its configuration with the master Routing Engine configuration.

When you configure nonstop active routing, you can bring the backup Routing Engine online after the master Routing Engine is already running. There is no requirement to start the two Routing Engines simultaneously.

Verifying VPLS Nonstop Active Routing Operation

To see whether or not nonstop active routing is enabled, issue the show task replication command.

NOTE: You must issue the show task replication command on the master Routing Engine. This command is not supported on the backup Routing Engine.

For more information on this command, see the *JUNOS System Basics and Services Command Reference*.

Tracing VPLS Nonstop Active Routing Synchronization Events

To trace the label and logical interface association that VPLS receives from the kernel replication state, include the nsr-synchronization statement at the [edit routing-options traceoptions flag] hierarchy level. This flag also traces the Layer 2 VPN signaling state replicated from routes advertised by BGP.

```
[edit routing-options]
traceoptions {
   flag nsr-synchronization;
}
```

Option: Configuring the Spanning Tree Protocol and VPLS on MX Series Routers

If multiple routers on a customer site are connected to the same PE, you should enable the Spanning Tree Protocol on that PE. To configure RSTP or MSTP and VPLS simultaneously, include the rstp or mstp statement at the [edit instance-type layer2-control] hierarchy level:

```
[edit]
instance-type layer2-control;
protocols {
   rstp {
      interface interface name;
      force-version stp; # To run STP instead of RSTP
   }
}
```

The Per-VLAN Spanning Tree (PVST) protocol maintains a separate spanning-tree instance for each VLAN. To enable PVST for a specific VLAN ID, there should be a VPLS instance with that VLAN ID and all of the logical interfaces assigned to that instance should have the same matching VLAN ID. To configure PVST with VPLS, include the vstp statement at the [edit instance-type layer2-control] hierarchy level:

```
[edit]
instance-type layer2-control;
protocols {
   vstp {
```

```
        interface interface name;
        vlan vlan-id;
    }
}
```

If you want only STP to run on a device, you can configure STP by including the force-version stp statement at the [edit protocols rstp] or [edit protocols vstp] hierarchy level:

```
[edit]
protocols {
    rstp {
        force-version stp;
    }
}
```

For more information about the Spanning Tree Protocol (VSTP, MSTP, RSTP, or STP), see the *MX Series Solutions Guide* and the *JUNOS Routing Protocols Configuration Guide*.

Filtering Layer 2 Packets in a VPLS Instance (MX Series Routers Only)

You can match the learn-vlan-id, user-vlan-id, and traffic-type terms for a VPLS instance on the MX Series platform. Packets entering or exiting the VPLS instance have a single VLAN tag. This VLAN tag is the same as what was received from the network. This VLAN tag corresponds to the one VLAN ID on a singly tagged logical interface or inner VLAN tag for the doubly tagged logical interface. The VLAN ID is used to qualify learned MAC addresses.

To configure a firewall filter for a VPLS instance, specify the conditions that the packet must match at the [edit firewall family vpls filter *filter-name* term *term-name* from] hierarchy level. To apply a firewall filter to a VPLS routing instance, include the input filter-name statement at [edit routing-instances *routing-instance-name* forwarding-options family vpls filter] hierarchy level. For more information, see the *JUNOS Policy Framework Configuration Guide*.

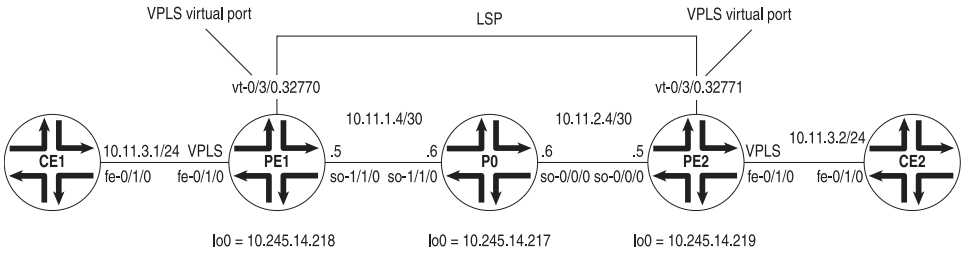
VPLS Configuration Examples

This section contains configuration examples and commands you can issue to verify your VPLS configuration:

- Example: VPLS Configuration (BGP Signaling) on page 538
- Example: VPLS Configuration (BGP and LDP Interworking) on page 549
- Example: Configuring VPLS Nonstop Active Routing on page 563

Example: VPLS Configuration (BGP Signaling)

Figure 48: VPLS Topology Diagram



VPLS table for PE1

Interface	MAC Address	Outgoing label	Received label
fe-0/1/0	aaaa.aaaa.aaaa	n/a	n/a
vt-0/3/0.32770	bbbb.bbbb.bbbb	800000	800002

g017142

In Figure 48 on page 538, a simple VPLS topology is enabled between routers PE1 and PE2. CE routers CE1 and CE2 use Ethernet-based interfaces to connect VLAN 600 to their local PE router. The PE routers PE1 and PE2 are connected to one another by LSPs enabled across a service provider backbone running MPLS, BGP, RSVP, and OSPF.

In a VPLS routing instance named green, PE1 has a local interface fe-0/1/0 and a virtual port of vt-0/3/0.32770 (the virtual port is created dynamically on the Tunnel Services PIC when VPLS is configured). PE2 has a local interface fe-0/1/0 and a virtual port of vt-0/3/0.32771 in the same green instance. As a result, routers CE1 and CE2 send Ethernet traffic to one another as if they were physically connected to each other on a LAN.

On Router CE1, the only item you need to configure is the Fast Ethernet interface that connects to PE1. Be sure to write down the VLAN identifier and IP address, so you can match them later on CE2.

Router CE1

```
[edit]
interfaces {
   fe-0/1/0 {
      vlan-tagging; # Configure VLAN tagging for VLAN VPLS or extended VLAN VPLS.
      unit 0 {
         vlan-id 600; # The Ethernet interface on CE2 must use the same VLAN ID.
         family inet {
            address 10.11.3.1/24; # The interface on CE2 must use the same prefix.
         }
      }
   }
}
```

On Router PE1, prepare the router for VPLS by configuring BGP, MPLS, OSPF, and RSVP. (These protocols are the basis for most Layer 2 VPN-related applications, including VPLS.) Include the signaling statement at the [edit protocols bgp group

group-name family l2vpn] hierarchy level, because VPLS uses the same infrastructure for internal BGP as Layer 2 VPNs.

NOTE: In JUNOS Release 7.3 and later, the signaling statement replaces the unicast statement at the [edit protocols bgp group *group-name* family l2vpn] hierarchy level. You must use the signaling statement if you wish to configure VPLS domains and Layer 2 VPNs simultaneously.

Next, configure VLAN tagging on the Fast Ethernet interface connected to Router CE1. Include VLAN VPLS encapsulation at both the physical and logical interface levels. Be sure to use the same VLAN ID for all Ethernet interfaces that are part of a single VPLS instance. Finally, add the Fast Ethernet interface into a VPLS routing instance and specify the site range, site ID number, and site name.

Router PE1

```
[edit]
interfaces {
   fe-0/1/0 {
      vlan-tagging;# Configure VLAN tagging for VLAN VPLS or extended VLAN VPLS.
      encapsulation vlan-vpls; # Configure VPLS encapsulation on both the
      unit 0 { # physical interface and the logical interface.
         encapsulation vlan-vpls;
         vlan-id 600;# The VLAN ID is the same one used by the CE routers.
      }
   }
   so-1/1/0 {
      unit 0 {
         family inet {
            address 10.11.1.5/30;
         }
         family mpls;
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.245.14.218/32;
         }
      }
   }
}
routing-options {
   autonomous-system 69;
   forwarding-table {
      export exp-to-fwd; # Apply a policy that selects an LSP for the VPLS instance.
   }
}
protocols {
   rsvp {
      interface all {
         aggregate;
      }
   }
   mpls {
      label-switched-path pe1-to-pe2 { # Configure an LSP to reach other VPLS PEs.
```

```
        to 10.245.14.219;
      }
      interface all;
    }
    bgp {
      group vpls-pe {
        type internal;
        local-address 10.245.14.218;
        family l2vpn { # VPLS uses the same infrastructure as Layer 2 VPNs
          signaling; # for internal BGP.
        }
        neighbor 10.245.14.217;
        neighbor 10.245.14.219;
      }
    }
    ospf {
      traffic-engineering;
      area 0.0.0.0 {
        interface so-1/1/0.0 {
          metric 11;
        }
        interface lo0.0 {
          passive;
        }
      }
    }
}
policy-options {
    policy-statement exp-to-fwd {
      term a {
        from community grn-com; # Matches the community in the VPLS instance.
        then {
          install-nexthop lsp pe1-to-pe2; # If there are multiple LSPs that exist
          accept; # between VPLS PE routers, this statement sends VPLS traffic
        } # over a specific LSP.
      }
    }
    community grn-com members target:11111:1; # Adds the instance to a BGP
} # community.
routing-instances {
    green {
      instance-type vpls; # Configure a VPLS routing instance.
      interface fe-0/1/0.0;
      route-distinguisher 10.245.14.218:1;
      vrf-target target:11111:1; # This value is important to the BGP community.
      protocols {
        vpls { # Configure a VPLS site range, site name, and site identifier.
          site-range 10;
          site greenPE1 {
            site-identifier 1;
          }
        }
      }
    }
}
```

On Router P0, configure BGP, MPLS, OSPF, and RSVP to interconnect PE1 and PE2.

Router P0

```
[edit]
interfaces {
   so-0/0/0 {
      unit 0 {
         family inet {
            address 10.11.2.6/30;
         }
         family mpls;
      }
   }
   so-1/1/0 {
      unit 0 {
         family inet {
            address 10.11.1.6/30;
         }
         family mpls;
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.245.14.217/32;
         }
      }
   }
}
protocols {
   rsvp {
      interface all {
         aggregate;
      }
   }
   mpls {
      interface all;
   }
   bgp {
      group vpls-pe {
         type internal;
         local-address 10.245.14.217;
         family l2vpn { # VPLS uses the same infrastructure as Layer 2 VPNs
            signaling; # for internal BGP.
         }
         neighbor 10.245.14.218;
         neighbor 10.245.14.219;
      }
   }
   ospf {
      traffic-engineering;
      area 0.0.0.0 {
         interface so-1/1/0.0 {
            metric 11;
         }
         interface so-0/0/0.0 {
            metric 15;
```

```
        }
        interface lo0.0 {
            passive;
        }
    }
  }
}
```

On Router PE2, configure BGP, MPLS, OSPF, and RSVP to complement the configuration on PE1. Next, configure VLAN tagging on the Fast Ethernet interface connected to Router CE2. Include VLAN VPLS encapsulation at both the physical and logical interface levels. Be sure to use the same VLAN ID for all Ethernet interfaces that are part of a single VPLS instance. Finally, add the Fast Ethernet interface into a VPLS routing instance and specify the site range, site ID number, and site name.

Router PE2

```
[edit]
interfaces {
  fe-0/1/0 {
    vlan-tagging; # Configure VLAN tagging for VLAN VPLS or extended VLAN VPLS.
    encapsulation vlan-vpls; # Configure VPLS encapsulation on both the
    unit 0 { # physical interface and logical interface.
      encapsulation vlan-vpls;
      vlan-id 600;# The VLAN ID is the same one used by the CE routers.
    }
  }
  so-0/0/0 {
    unit 0 {
      family inet {
        address 10.11.2.5/30;
      }
      family mpls;
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.245.14.219/32;
      }
    }
  }
}
routing-options {
  autonomous-system 69;
  forwarding-table {
    export exp-to-fwd; # Apply a policy that selects an LSP for the VPLS instance.
  }
}
protocols {
  rsvp {
    interface all {
      aggregate;
    }
  }
  mpls {
    label-switched-path pe2-to-pe1 { # Configure an LSP to other VPLS PE routers.
```

```
        to 10.245.14.218;
      }
      interface all;
    }
    bgp {
      group vpls-pe {
        type internal;
        local-address 10.245.14.219;
        family l2vpn { # VPLS uses the same infrastructure as Layer 2 VPNs
          signaling; # for internal BGP.
        }
        neighbor 10.245.14.217;
        neighbor 10.245.14.218;
      }
    }
    ospf {
      traffic-engineering;
      area 0.0.0.0 {
        interface so-0/0/0.0 {
          metric 15;
        }
        interface lo0.0 {
          passive;
        }
      }
    }
}
policy-options {
    policy-statement exp-to-fwd {
      term a {
        from community grn-com; # Matches the community with the VPLS instance.
        then {
          install-nexthop lsp pe2-to-pe1; # If there are multiple LSPs that exist
          accept; # between VPLS PE routers, this statement sends VPLS traffic
        } # over a specific LSP.
      }
    }
    community grn-com members target:11111:1; # This adds the instance into a BGP
      community.
}
routing-instances {
    green {
      instance-type vpls; # Configure a VPLS routing instance.
      interface fe-0/1/0.0;
      route-distinguisher 10.245.14.219:1;
      vrf-target target:11111:1; # This value is important for the BGP community.
      protocols {
        vpls { # Configure a VPLS site range, site name, and site identifier.
          site-range 10;
          site greenPE2 {
            site-identifier 2;
          }
        }
      }
    }
}
```

On Router CE2, complete your VPLS network by configuring the Fast Ethernet interface that connects to PE2. Use the same VLAN identifier and IP address prefix used on Router CE1.

Router CE2

```
[edit]
interfaces {
   fe-0/1/0 {
      vlan-tagging; # Configure VLAN tagging for VLAN VPLS or extended VLAN VPLS.
      unit 0 {
         vlan-id 600; # The Ethernet interface on CE1 must use the same VLAN ID.
         family inet {
            address 10.11.3.2/24; # The interface on CE1 must use the same prefix.
         }
      }
   }
}
```

Verifying Your Work

To verify proper operation of VPLS, use the following commands:

- clear vpls mac-address instance *instance-name*
- show interfaces terse
- show route forwarding-table family mpls
- show route forwarding-table family vpls (destination | extensive | matching | table)
- show route instance (detail)
- show system statistics vpls
- show vpls connections
- show vpls statistics

The following section shows the output of these commands on Router PE1 as a result of the configuration example:

```
user@PE1> show interfaces terse
Interface               Admin Link Proto Local                 Remote
so-1/1/0                up    up
so-1/1/0.0              up    up   inet  10.11.1.5/30
                                  mpls
so-1/1/1                up    up
so-1/1/2                up    up
so-1/1/3                up    up
fe-0/1/0                up    up
fe-0/1/0.0              up    up   vpls   # This is the local Fast Ethernet
# interface.
fe-0/1/1                up    up
fe-0/1/2                up    up
fe-0/1/3                up    up
gr-0/3/0                up    up
ip-0/3/0                up    up
mt-0/3/0                up    up
pd-0/3/0                up    up
```

```
pe-0/3/0                up    up
vt-0/3/0                up    up
vt-0/3/0.32770          up    up  # This is the dynamically generated virtual
port.
dsc                     up    up
fxp0                    up    up
fxp0.0                  up    up   inet  192.186.14.218/24
fxp1                    up    up
fxp1.0                  up    up   tnp   4
gre                     up    up
ipip                    up    up
lo0                     up    up
lo0.0                   up    up   inet  10.245.14.218       --> 0/0
                                          127.0.0.1           --> 0/0
                                     inet6 fe80::2a0:a5ff:fe28:13e0
                                          feee::10:245:14:218
lsi                     up    up
mtun                    up    up
pimd                    up    up
pime                    up    up
tap                     up    up

user@PE1> show system statistics vpls
vpls:
        0 total packets received
        0 with size smaller than minimum
        0 with incorrect version number
        0 packets for this host
        0 packets with no logical interface
        0 packets with no family
        0 packets with no route table
        0 packets with no auxiliary table
        0 packets with no corefacing entry
        0 packets with no CE-facing entry
         6 mac route learning requests # This indicates that VPLS is working.
        6 mac routes learnt
        0 mac routes aged
        0 mac routes moved
```

To display VPLS source and destination MAC address accounting information, use the destination, extensive, matching, or table option with the show route forwarding-table family vpls command. When you analyze the display output, keep in mind the following:

- VPLS MAC address accounting is handled on a per-MAC address basis for each VPLS instance. All information is retrieved from MAC address entries in the MAC address table. VPLS MAC address accounting is performed only on local CE routers.
- The VPLS counters for source and destination MAC addresses increment continuously until the oldest MAC address entries are removed from the memory buffer, either when the entries time out or if the VPLS instance is restarted.

```
user@PE1> show route forwarding-table family vpls extensive

Routing table: green.vpls [Index 2]
VPLS:

Destination:  default
```

```
  Route type: dynamic             Route reference: 0
  Flags: sent to PFE
  Next-hop type: flood                 Index: 353      Reference: 1

Destination:  default
  Route type: permanent           Route reference: 0
  Flags: none
  Next-hop type: discard               Index: 298      Reference: 1

Destination:  fe-0/1/0.0
  Route type: dynamic             Route reference: 0
  Flags: sent to PFE
  Next-hop type: flood                 Index: 355      Reference: 1

Destination:  bb:bb:bb:bb:bb:bb/48 # This MAC address belongs to remote
CE2.
  Route type: dynamic             Route reference: 0
  Flags: sent to PFE, prefix load balance
  Next-hop type: indirect              Index: 351      Reference: 4
  Next-hop type: Push 800000, Push 100002(top)
  Next-hop interface: so-1/1/0.0

Destination:  aa:aa:aa:aa:aa:aa/48 # This MAC address belongs to local
CE1.
  Route type: dynamic             Route reference: 0
  Flags: sent to PFE, prefix load balance
  Next-hop type: unicast               Index: 354      Reference: 2
  Next-hop interface: fe-0/1/0.0

user@PE1> show route forwarding-table family vpls

Routing table: green.vpls
VPLS:
Destination        Type RtRef Next hop          Type Index NhRef Netif
default            dynm     0                   flood  353     1
default            perm     0                   dscd   298     1
fe-0/1/0.0         dynm     0                   flood  355     1
bb:bb:bb:bb:bb:bb/48 # This MAC address belongs to remote CE2.
                   dynm     0                   indr   351     4
                                                Push 800000, Push
100002(top)
so-1/1/0.0
aa:aa:aa:aa:aa:aa/48 # This MAC address belongs to local CE1.
                  dynm     0                  ucst   354     2 fe-0/1/0.0


user@PE1> show route forwarding-table family mpls

Routing table: mpls
MPLS:
Destination        Type RtRef Next hop          Type Index NhRef Netif
default            perm     0                   dscd    19     1
0                  user     0                   recv    18     3
1                  user     0                   recv    18     3
2                  user     0                   recv    18     3
100000             user     0 10.11.1.6         swap  100001     so-1/1/0.0
800002             user     0                     Pop
vt-0/3/0.32770
vt-0/3/0.32770 (VPLS)
                   user     0                   indr   351     4
```

```
                                                    Push 800000, Push
100002(top) so-1/1/0.0


user@PE1> show route instance green detail

green:
  Router ID: 0.0.0.0
  Type: vpls              State: Active
  Interfaces:
    fe-0/1/0.0  # This is the local Fast Ethernet interface.
   vt-0/3/0.32770  # This is the dynamically generated VPLS virtual port.

  Route-distinguisher: 10.245.14.218:1
  Vrf-import: [ __vrf-import-green-internal__ ]
  Vrf-export: [ __vrf-export-green-internal__ ]
  Vrf-import-target: [ target:11111:1 ]
  Vrf-export-target: [ target:11111:1 ]
  Tables:
    green.l2vpn.0          : 2 routes (2 active, 0 holddown, 0 hidden)


user@PE1> show vpls connections

L2VPN Connections:
Legend for connection status (St)
OR -- out of range              WE -- intf encaps != instance encaps
EI -- encapsulation invalid     Dn -- down
EM -- encapsulation mismatch    VC-Dn -- Virtual circuit down
CM -- control-word mismatch     -> -- only outbound conn is up
CN -- circuit not present       <- -- only inbound  conn is up
OL -- no outgoing label         Up -- operational
NC -- intf encaps not CCC/TCC   XX -- unknown
NP -- interface not present
Legend for interface status
Up -- operational
Dn -- down
Instance: green
Local site: greenPE1 (1)
    connection-site           Type  St     Time last up          # Up
trans
    2                         rmt   Up     Jan 24 06:26:49 2003
  1
      Local interface: vt-0/3/0.32770, Status: Up, Encapsulation: VPLS
      Remote PE: 10.245.14.219, Negotiated control-word: No
      Incoming label: 800002, Outgoing label: 800000


user@PE1> show system statistics vpls

vpls:
        0 total packets received
        0 with size smaller than minimum
        0 with incorrect version number
        0 packets for this host
        0 packets with no logical interface
        0 packets with no family
        0 packets with no route table
        0 packets with no auxiliary table
        0 packets with no corefacing entry
```

```
        0 packets with no CE-facing entry
        7 mac route learning requests
        7 mac routes learnt
        0 mac routes aged
        0 mac routes moved


user@PE1> show route instance green detail

green:
  Router ID: 0.0.0.0
  Type: vpls              State: Active
   Interfaces:
    fe-0/1/0.0
    vt-0/3/0.32770
  Route-distinguisher: 10.245.14.218:1
  Vrf-import: [ __vrf-import-green-internal__ ]
  Vrf-export: [ __vrf-export-green-internal__ ]
  Vrf-import-target: [ target:11111:1 ]
  Vrf-export-target: [ target:11111:1 ]
  Tables:
    green.l2vpn.0          : 2 routes (2 active, 0 holddown, 0 hidden)

user@PE1> show vpls statistics

Layer-2 VPN Statistics:
Instance: green
   Local interface: fe-0/1/0.0, Index: 351
   Remote provider edge router: 10.245.14.219
     Multicast packets:                  363
     Multicast bytes  :                30956
     Flood packets    :                    0
     Flood bytes      :                    0
   Local interface: vt-0/3/0.32770, Index: 354
   Remote provider edge router: 10.245.14.219
     Multicast packets:                  135
     Multicast bytes  :                12014
     Flood packets    :                  135
     Flood bytes      :                12014
```

To clear all MAC address entries for a VPLS instance from the VPLS table, issue the clear vpls mac-address instance *instance-name* command. Add the logical-system *logical-system-name* option to clear entries in a VPLS instance within a logical system. Use the *mac-address* option to remove individual MAC addresses.

Example: VPLS Configuration (BGP and LDP Interworking)

Figure 49: Topology for VPLS Configuration Example

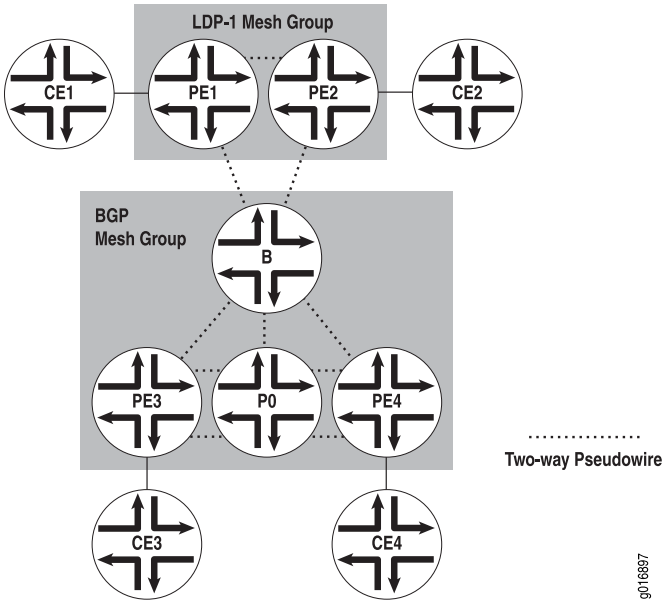


Figure 49 on page 549, shows two VPLS mesh groups: LDP-1 and the default BGP mesh group. The VPLS instance is named vi in the configuration. Table 7 on page 549 shows the addresses for the router interfaces in the example topology.

Table 7: Router Interface Addresses for VPLS Configuration Example

Router	Interface	Address
CE1	fe-0/0/3 (link to Router PE1)	10.12.31.1
	loopback	10.12.53.1
CE2	fe-0/0/1 (link to Router PE2)	10.12.31.2
	loopback	10.12.53.2
PE1	t1-1/1/1 (link to Router PE2)	10.12.100.17
	t1-0/1/0 (link to Router B)	10.12.100.2
	loopback	10.255.170.106
PE2	t1-0/1/1 (link to Router PE1)	10.12.100.18
	t1–0/1/3 (link to Router B)	10.12.100.6
	loopback	10.255.170.104

Table 7: Router Interface Addresses for VPLS Configuration Example *(continued)*

Router	Interface	Address
B	t1-0/1/2 (link to Router PE1)	10.12.100.1
	t1-0/1/3 (link to Router PE2)	10.12.100.5
	so-0/2/2 (link to Router PE3)	10.12.100.9
	fe-0/0/3 (link to Router PE4)	10.12.100.13
	loopback	10.255.170.98
PE3	s0-0/2/1 (link to Router B)	10.12.100.10
	so-0/2/2 (link to Router P0)	10.12.100.21
	loopback	10.255.170.96
P0	so-0/2/1 (link to Router PE3)	10.12.100.22
	t1-0/1/3 (link to Router PE4)	10.12.100.25
	loopback	10.255.170.100
PE4	fe-0/0/3 (link to Router B)	10.12.100.14
	t1-0/1/3 (link to Router P0)	10.12.100.26
	loopback	10.255.170.102
CE3	ge-1/2/1 (link to PE3)	10.12.31.3
	loopback	10.12.53.3
CE4	fe-0/0/2 (link to PE4)	10.12.31.4
	loopback	10.12.53.4

On Router CE3, the only item you need to configure is the Gigabit Ethernet interface that connects to PE3.

Router CE3

```
[edit]
interfaces {
    ge-1/2/1 {
        unit 0 {
            family inet {
                address 10.12.31.1/24;
            }
        }
    }
}
```

On Router PE3, prepare the router for VPLS by configuring BGP, MPLS, OSPF, and LDP. (These protocols are the basis for most Layer 2 VPN-related applications, including VPLS.) Include the signaling statement at the [edit protocols bgp group *group-name* family l2vpn] hierarchy level, because VPLS uses the same infrastructure for internal BGP as Layer 2 VPNs.

NOTE: In JUNOS Release 7.3 and later, the signaling statement replaces the unicast statement at the [edit protocols bgp group *group-name* family l2vpn] hierarchy level. You must use the signaling statement if you wish to configure VPLS domains and Layer 2 VPNs simultaneously.

Next, configure VLAN tagging on the Gigabit Ethernet interface connected to Router CE3. Finally, add the Gigabit Ethernet interface into a VPLS routing instance and specify the site range, site ID number, and site name.

Router PE3

```
[edit]
interfaces {
   so-0/2/1 {
      unit 0 {
         family inet {
            address 10.12.100.10/30;
         }
         family mpls;
      }
   so-0/2/2 {
      unit 0 {
         family inet {
            address 10.12.100.21/30;
         }
         family mpls;
      }
   }
      ge-1/3/1 {
         encapsulation ethernet-vpls;
         unit 0 {
            family vpls;
         }
      }
   }
}
protocols {
   mpls {
      interface all;
   }
   bgp {
      log-updown;
      group int {
         type internal;
         local-address 10.255.170.96;
         family l2vpn {
            signaling;
         }
         neighbor 10.255.170.98;
         neighbor 10.255.170.102;
```

```
        }
      }
      ospf {
        area 0.0.0.0 {
          interface so-0/2/1.0;
          interface so-0/2/2.0;
          interface lo0.0 {
            passive;
          }
        }
      }
      ldp {
        interface so-0/2/1.0;
        interface so-0/2/2.0;
      }
    }
    routing-instances {
      v1 {
        instance-type vpls;
        interface ge-1/3/1.0;
        route-distinguisher 10.255.170.96:1;
        vrf-target target:1:2;
        protocols {
          vpls {
            site-range 10;
            site 1 {
              site-identifier 3;
            }
          }
        }
      }
```

On Router P0, configure MPLS, OSPF, and LDP to interconnect PE3 and PE4.

Router P0

```
[edit]
interfaces {
  t1-0/1/3 {
    unit 0 {
      family inet {
        address 10.12.100.25/30;
      }
      family mpls;
    }
  so-0/2/1 {
    unit 0 {
      family inet {
        address 10.12.100.22/30;
      }
      family mpls;
    }
  }
}
protocols {
  mpls {
    interface all;
```

```
    }
    ospf {
      area 0.0.0.0 {
        interface so-0/2/1.0;
        interface t1-0/1/3.0;
        interface lo0.0 {
          passive;
        }
      }
    }
    ldp {
      interface t1-0/1/3.0;
      interface so-0/2/1.0;
    }
  }
```

On Router PE4, configure BGP, MPLS, OSPF, and LDP to complement the configuration on PE3. Next, configure VLAN tagging on the Fast Ethernet interface connected to Router CE4. Include VLAN VPLS encapsulation at both the physical and logical interface levels. Finally, add the Fast Ethernet interface into a VPLS routing instance and specify the site range, site ID number, and site name.

Router PE4

```
[edit]
interfaces {
  fe-0/0/2 {
  encapsulation ethernet-vpls;
    unit 0 {
      family vpls;
    }
  fe-0/0/3 {
    unit 0 {
      family inet {
        address 10.12.100.14/30;
      }
      family mpls;
    }
  }
  t1-0/1/3 {
    unit 0 {
      family inet {
        address 10.12.100.26/30;
      }
      family mpls;
    }
  }
}
protocols {
  mpls {
    interface all;
  }
  bgp {
    log-updown;
    group int {
      type internal;
      local-address 10.255.170.102;
```

```
            family l2vpn {
               signaling;
            }
            neighbor 10.255.170.96;
            neighbor 10.255.170.98;
         }
         }
      }
      ospf {
         area 0.0.0.0 {
            interface fe-0/0/3.0;
            interface t1-0/1/3.0;
            interface lo0.0 {
               passive;
            }
         }
      }
      ldp {
         interface fe-0/0/3.0;
         interface t1-0/1/3.0;
         interface lo0.0;
      }
   }
```

On Router CE4, configure the Fast Ethernet interface that connects to PE4.

Router CE4

```
[edit]
interfaces {
   fe-0/0/2 {
      unit 0 {
         family inet {
            address 10.12.31.4/24;
         }
      }
   }
}
```

On Router B, the area border router, configure the interfaces. Next, configure BGP, MPLS, OSPF, and LDP. Be sure to include the loopback interface in the LDP configuration by including the interface lo0.0 statement at the [edit protocols ldp] hierarchy level. For BGP, include the signaling statement at the [edit bgp group *group-name* family l2vpn] hierarchy level. Last, configure the VPLS instance with both BGP and LDP signaling. Configure the LDP-1 mesh group by including the mesh-group ldp1 statement at the [edit routing-instances v1 protocols vpls] hierarchy level.

Router B

```
[edit]
interfaces {
   fe-0/0/3 {
      unit 0 {
         family inet {
            address 10.12.100.13/30;
         }
         family mpls;
      }
   }
```

```
    t1-0/1/2 {
        unit 0 {
            family inet {
                address 10.12.100.1/30;
            }
            family mpls;
        }
    }
    t1-0/1/3 {
        unit 0 {
            family inet {
                address 10.12.100.5/30;
            }
            family mpls;
        }
    }
    so-0/2/2 {
        unit 0 {
            family inet {
                address 10.12.100.9/30;
            }
            family mpls;
        }
    }
}
protocols {
    mpls {
        interface all;
    }
    bgp {
        log-updown;
        group int {
            type internal;
            local-address 10.255.170.98;
            family l2vpn {
                signaling;
            }
            neighbor 10.255.170.96;
            neighbor 10.255.170.102;
        }
    }
    ospf {
        area 0.0.0.0 {
            interface t1-0/1/2.0;
            interface t1-0/1/3.0;
            interface so-0/2/2.0;
            interface fe-0/0/3.0;
            interface lo0.0 {
                passive;
            }
        }
    }
    ldp {
        interface fe-0/0/3.0;
        interface t1-0/1/2.0;
        interface t1-0/1/3.0;
```

```
        interface so-0/2/2.0;
        interface lo0.0;
    }
}
routing-instances {
    v1 {
        instance-type vpls;
        route-distinguisher 10.255.170.98:1;
        vrf-target target:1:2;
        protocols {
            vpls {
                site-range 10;
                site 1 {
                    site-identifier 1;
                }
                vpls-id 101;
                mesh-group ldp-1 {
                    neighbor 10.255.170.106;
                    neighbor 10.255.170.104;
                }
            }
        }
    }
}
```

Finally, configure the LDP PE routers. On Router PE1, prepare the router for VPLS by configuring LDP, MPLS, and OSPF. Next, configure VPLS encapsulation on the Fast Ethernet interface connected to CE1. Finally, add the Fast Ethernet interface to the routing instance, specifying the VPLS ID and the neighboring routers' loopback addresses.

Router PE1

```
[edit]
interfaces {
    fe-0/0/3 {
        encapsulation ethernet-vpls;
        unit 0 {
            family vpls;
        }
    }
    t1-0/1/0 {
        unit 0 {
            family inet {
                address 10.12.100.2/30;
            }
            family mpls;
        }
    }
    t1-1/1/1 {
        unit 0 {
            family inet {
                address 10.12.100.17/30;
            }
            family mpls;
        }
    }
```

```
}
protocols {
    mpls {
        interface all;
    }
    ospf {
        area 0.0.0.0 {
            interface t1-0/1/0.0;
            interface t1-1/1/1.0;
            interface lo0.0 {
                passive;
            }
        }
    }
    ldp {
        interface t1-0/1/0.0;
        interface t1-1/1/1.0;
        interface lo0.0;
    }
}
routing-instances {
    v1 {
        instance-type vpls;
        interface fe-0/0/3.0;
        protocols {
            vpls {
                vpls-id 101;
                neighbor 10.255.170.98;
                neighbor 10.255.170.104;
            }
        }
    }
}
```

Next, configure the Fast Ethernet interface on Router CE1 that connects to Router PE1.

Router CE1

```
[edit]
interfaces {
    fe-0/0/3 {
        unit 0 {
            family inet {
                address 10.12.31.1/24;
            }
        }
    }
}
```

On Router PE2, prepare the router for VPLS by configuring LDP, MPLS, and OSPF. Next, configure VPLS encapsulation on the Fast Ethernet interface connected to Router CE1. Finally, add the Fast Ethernet interface to the routing instance, specifying the VPLS ID and the neighboring routers' loopback addresses.

Router PE2

```
[edit]
interfaces {
    t1-0/1/1 {
```

```
        unit 0 {
            family inet {
                address 10.12.100.18/30;
            }
            family mpls;
        }
    t1-0/1/3 {
        unit 0 {
            family inet {
                address 10.12.100.6/30;
            }
            family mpls;
        }
    }
    fe-1/0/2 {
        encapsulation ethernet-vpls;
        unit 0 {
            family vpls;
        }
    }
}
protocols {
    mpls {
        interface all;
    }
    ospf {
        area 0.0.0.0 {
            interface t1-0/1/3.0;
            interface t1-0/1/1.0;
            interface lo0.0 {
                passive;
            }
        }
    }
    ldp {
        interface t1-0/1/1.0;
        interface t1-0/1/3.0;
        interface lo0.0;
    }
}
routing-instances {
    v1 {
        instance-type vpls;
        interface fe-1/0/2.0;
        protocols {
            vpls {
                vpls-id 101;
                neighbor 10.255.170.98;
                neighbor 10.255.170.106;
            }
        }
    }
}
```

Finally, on Router CE2 configure the Fast Ethernet interface connected to PE2:

Router CE2

```
[edit]
interfaces {
   fe-0/0/1 {
      unit 0 {
         family inet {
            address 10.12.31.2/24;
         }
      }
   }
}
```

Verifying Your Work

To verify proper operation of VPLS, use the following commands:

- show bgp summary
- show ldp neighbor
- show vpls connections
- show route forwarding-table family vpls (destination | extensive | matching | table)
- show interfaces vt* terse
- show vpls flood extensive
- show vpls statistics

The following section shows the output of some of these commands on Router B as a result of the configuration example.

Use the show bgp summary command to verify BGP signaling for VPLS is up.

```
user@PB> show bgp summary
Groups: 1 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l2vpn.0            2          2          0          0          0          0
Peer               AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Damped...
10.255.170.96   65000        124        125       0       0       54:26 Establ
  bgp.l2vpn.0: 1/1/0
  v1.l2vpn.0: 1/1/0
10.255.170.102  65000        122        124       0       0       54:18 Establ
  bgp.l2vpn.0: 1/1/0
  v1.l2vpn.0: 1/1/0
```

Use the show ldp neighbors command to verify that LDP signaling for VPLS is up.

```
user@B> show ldp neighbors
Address            Interface          Label space ID         Hold time
10.255.170.104     lo0.0              10.255.170.104:0         41
10.255.170.106     lo0.0              10.255.170.106:0         38
10.12.100.14       fe-0/0/3.0         10.255.170.102:0         12
10.12.100.10       so-0/2/2.0         10.255.170.96:0          14
10.12.100.2        t1-0/1/2.0         10.255.170.106:0         14
10.12.100.6        t1-0/1/3.0         10.255.170.104:0         13
```

To verify that the VPLS connections are up, use the show vpls connections command.

```
user@B>show vpls connections
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 availble

Legend for interface status
Up -- operational
Dn -- down

Instance: v1
BGP-VPLS State
  Local site: 1 (1)
    connection-site           Type  St     Time last up          # Up trans
    3                         rmt   Up     Jan 22 16:38:47 2008           1
      Local interface: vt-0/3/0.1048834, Status: Up, Encapsulation: VPLS
        Description: Intf - vpls v1 local site 1 remote site 3
      Remote PE: 10.255.170.96, Negotiated control-word: No
      Incoming label: 800258, Outgoing label: 800000
    4                         rmt   Up     Jan 22 16:38:54 2008           1
      Local interface: vt-0/3/0.1048835, Status: Up, Encapsulation: VPLS
        Description: Intf - vpls v1 local site 1 remote site 4
      Remote PE: 10.255.170.102, Negotiated control-word: No
      Incoming label: 800259, Outgoing label: 800000 LDP-VPLS State
VPLS-id: 101
  Mesh-group connections: m1
    Neighbor                  Type  St     Time last up          # Up trans
    10.255.170.104(vpls-id 101) rmt Up     Jan 22 16:38:40 2008           1
      Local interface: vt-0/3/0.1048833, Status: Up, Encapsulation: ETHERNET
        Description: Intf - vpls v1 neighbor 10.255.170.104 vpls-id 101
      Remote PE: 10.255.170.104, Negotiated control-word: No
      Incoming label: 800001, Outgoing label: 800000
    10.255.170.106(vpls-id 101) rmt Up     Jan 22 16:38:39 2008           1
      Local interface: vt-0/3/0.1048832, Status: Up, Encapsulation: ETHERNET
        Description: Intf - vpls v1 neighbor 10.255.170.106 vpls-id 101
      Remote PE: 10.255.170.106, Negotiated control-word: No
      Incoming label: 800000, Outgoing label: 800000
```

To display VPLS routes (MAC addresses) in the vpls forwarding table, use the show route forwarding-table family vpls command.

```
user@B> show route forwarding-table family vpls
Routing table: v1.vpls
VPLS:
Destination        Type RtRef Next hop           Type Index NhRef Netif
```

```
default             perm     0                    rjct   540     1
vt-0/3/0.1048832    user     0                    comp   587     3
vt-0/3/0.1048833    user     0                    comp   587     3
vt-0/3/0.1048834    user     0                    comp   589     3
vt-0/3/0.1048835    user     0                    comp   589     3
00:17:cb:c2:10:01/48
                    dynm     0                    indr 262143     4
                                                  Push 800000   580     2
t1-0/1/3.0
00:17:cb:c2:10:02/48
                    dynm     0                    indr 262145     4
                               10.12.100.14       Push 800000   594     2
fe-0/0/3.0
00:17:cb:c2:10:03/48
                    dynm     0                    indr 262142     4
                                                  Push 800000   576     2
t1-0/1/2.0
00:17:cb:c2:10:bd/48
                    dynm     0                    indr 262144     4
                                                  Push 800000   585     2
so-0/2/2.0
```

To display VPLS source and destination MAC address accounting information, use the destination, extensive, matching, or table option with the show route forwarding-table family vpls command. When you analyze the display output, keep in mind the following:

- VPLS MAC address accounting is handled on a per-MAC address basis for each VPLS instance. All information is retrieved from MAC address entries in the MAC address table. VPLS MAC address accounting is performed only on local CE routers.
- The VPLS counters for source and destination MAC addresses increment continuously until the oldest MAC address entries are removed from the memory buffer, either when the entries time out or if the VPLS instance is restarted.

To display status information about Virtual Loopback Tunnel interfaces in the VPLS instance, use the show interfaces vt* terse command.

```
user@B> show interfaces vt* terse
Interface               Admin Link Proto    Local                 Remote
vt-0/3/0                up    up
vt-0/3/0.1048832        up    up   vpls
vt-0/3/0.1048833        up    up   vpls
vt-0/3/0.1048834        up    up   vpls
vt-0/3/0.1048835        up    up   vpls
```

To display VPLS route information related to the flood process, use the show vpls flood extensive command.

```
user@B> show vpls flood extensive
Name: v1
CEs: 0
VEs: 4
  Flood route prefix: 0x4a/32
  Flood route type: IFF_FLOOD
  Flood route owner: vt-0/3/0.1048834
```

```
Flood group name: __ves__
Flood group index: 0
Nexthop type: comp
Nexthop index: 589
  Flooding to:
  Name             Type          NhType          Index
  m1               Group          comp            588
      Composition: flood-to-all
      Flooding to:
      Name             Type          NhType          Index
      vt-0/3/0.1048832 VE             indr           262142
      vt-0/3/0.1048833 VE             indr           262143

Flood route prefix: 0x4b/32
Flood route type: IFF_FLOOD
Flood route owner: vt-0/3/0.1048835
Flood group name: __ves__
Flood group index: 0
Nexthop type: comp
Nexthop index: 589
  Flooding to:
  Name             Type          NhType          Index
  m1               Group          comp            588
      Composition: flood-to-all
      Flooding to:
      Name             Type          NhType          Index
      vt-0/3/0.1048832 VE             indr           262142
      vt-0/3/0.1048833 VE             indr           262143

Flood route prefix: 0x48/32
Flood route type: IFF_FLOOD
Flood route owner: vt-0/3/0.1048832
Flood group name: m1
Flood group index: 2
Nexthop type: comp
Nexthop index: 587
  Flooding to:
  Name             Type          NhType          Index
  __ves__          Group          comp            586
      Composition: flood-to-all
      Flooding to:
      Name             Type          NhType          Index
      vt-0/3/0.1048834 VE             indr           262144
      vt-0/3/0.1048835 VE             indr           262145

Flood route prefix: 0x49/32
Flood route type: IFF_FLOOD
Flood route owner: vt-0/3/0.1048833
Flood group name: m1
Flood group index: 2
Nexthop type: comp
Nexthop index: 587
  Flooding to:
  Name             Type          NhType          Index
  __ves__          Group          comp            586
      Composition: flood-to-all
      Flooding to:
      Name             Type          NhType          Index
      vt-0/3/0.1048834 VE             indr           262144
      vt-0/3/0.1048835 VE             indr           262145
```

To view packet flow statistics for the VPLS instance, use the show vpls statistics command:

```
user@B> show vpls statistics
Instance: v1
   Local interface: vt-0/3/0.1048832, Index: 72
   Remote PE: 10.255.170.106
     Multicast packets:                     6
     Multicast bytes  :                   360
     Flooded packets  :                    16
     Flooded bytes    :                  1188
     Current MAC count:                     1
   Local interface: vt-0/3/0.1048833, Index: 73
   Remote PE: 10.255.170.104
     Multicast packets:                     4
     Multicast bytes  :                   240
     Flooded packets  :                     6
     Flooded bytes    :                   398
     Current MAC count:                     1
   Local interface: vt-0/3/0.1048834, Index: 74
   Remote PE: 10.255.170.96
     Multicast packets:                     2
     Multicast bytes  :                   120
     Flooded packets  :                     4
     Flooded bytes    :                   278
     Current MAC count:                     1
   Local interface: vt-0/3/0.1048835, Index: 75
   Remote PE: 10.255.170.102
     Multicast packets:                     1
     Multicast bytes  :                    60
     Flooded packets  :                     2
     Flooded bytes    :                   158
     Current MAC count:                     1
```

Example: Configuring VPLS Nonstop Active Routing

The following example enables graceful Routing Engine switchover, nonstop active routing, and nonstop active routing trace options for VPLS.

```
[edit]
system commit {
   synchronize;
}
chassis {
   redundancy {
      graceful-switchover; # This enables graceful Routing Engine switchover on the
         # routing platform.
   }
}
routing-options {
   nonstop-routing; # This enables nonstop active routing on the routing platform.
   traceoptions {
      flag nsr-synchronization;
   }
}
```

Example: Configuring Inter-AS VPLS with MAC Processing at the ASBR

This example describes how to configure inter-AS Virtual Private LAN Service (VPLS) with MAC processing between BGP-signaled VPLS and LDP-signaled VPLS. This feature is described in RFC 4761 as multi-AS VPLS option E or method E.

This example is organized in the following sections:

- Requirements on page 564
- Overview and Topology on page 564
- Configuration on page 565
- Verification on page 589

Requirements

To support inter-AS VPLS between BGP-signaled VPLS and LDP-signaled VPLS, your network must meet the following hardware and software requirements:

- MX Series or M320 routers for the ASBRs.
- JUNOS Software Release 9.3 or higher.
- Gigabit Ethernet or 10-Gigabit Ethernet interfaces.

Overview and Topology

VPLS is a key enabler for delivering multipoint Ethernet service. Major service providers have implemented IP and MPLS backbones and offer VPLS services to large enterprises. Growing demand requires the VPLS network to scale to support many VPLS customers with multiple sites spread across geographically dispersed regions. BGP-signaled VPLS signaling offers scaling advantages over LDP-signaled VPLS. In some environments there is a need for BGP-signaled VPLS to interoperate with existing LDP-signaled VPLS.

This example shows one way to configure BGP-signaled VPLS interworking with an existing LDP-signaled VPLS network.

The advantages of the configuration are:

- You can interconnect customer sites that are spread across different autonomous systems (ASs).
- LDP-signaled VPLS and BGP-signaled VPLS interworking is supported.
- Because the ASBR supports MAC operations, customer sites can be connected directly to the ASBR.
- The inter-AS link is not restricted to Ethernet interfaces.
- Additional configuration for multihoming is relatively straightforward.

Traffic from the interworking virtual private LAN services is switched at the ASBR. The ASBR does all the data plane operations: flooding, MAC learning, aging, and MAC

forwarding for each AS to switch traffic among any customer facing interfaces and between the fully meshed pseudowires in the AS. A single pseudowire is created between the ASBRs across the inter-AS link and the ASBRs forward traffic from the pseudowires in each AS to the peer ASBR.

Each ASBR performs VPLS operations within its own AS and performs VPLS operations with the ASBR in the other AS. The ASBR treats the other AS as a BGP-signaled VPLS site. To establish VPLS pseudowires, VPLS NLRI messages are exchanged across the EBGP sessions on the inter-AS links between the ASBRs.

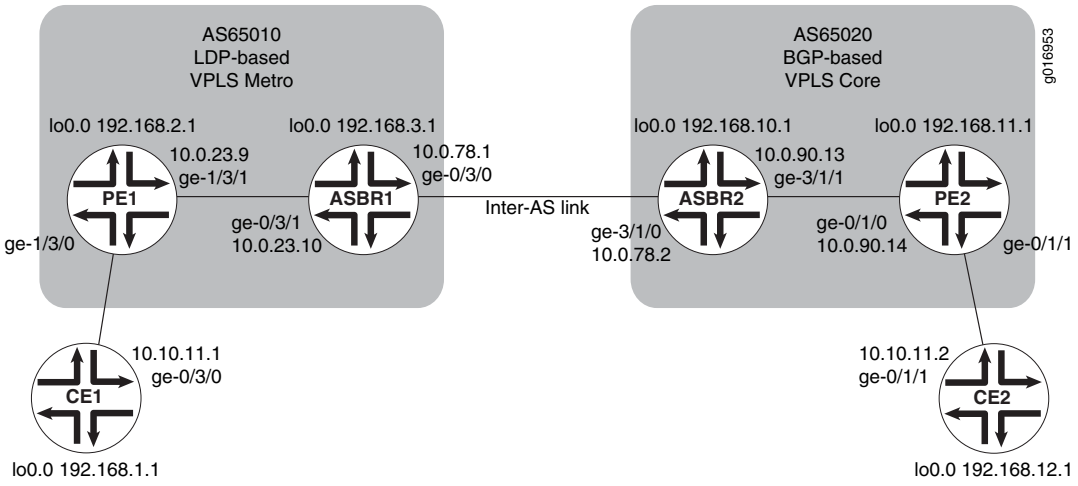
The sample metro network is configured for LDP-signaled VPLS. The core network is configured for BGP-signaled VPLS.

The first part of the example shows the basic configuration steps to configure the logical interfaces, OSPF, internal BGP, LDP, and MPLS. This part of the configuration is the same as other VPLS configurations for LDP-signaled VPLS and BGP-signaled VPLS.

The unique part of the example is configured in the VPLS routing instances, external BGP, and the policy that populates the BGP route table with routes learned from direct routes and OSPF routes. Additional details about the configuration statements are included in the step-by-step procedure.

Figure 50 on page 565 shows the topology used in this example.

Figure 50: Inter-AS VPLS with MAC Operations Example Topology



Configuration

To configure inter-AS VPLS between BGP-signaled VPLS and LDP-signaled VPLS, perform these tasks.

NOTE: In any configuration session it is a good practice to periodically use the commit check command to verify that the configuration can be committed.

- Configuring Interfaces on page 566
- Configuring OSPF on page 568
- Configuring the Internal BGP Peer Group on page 569
- Configuring LDP on page 571
- Configuring MPLS on page 572
- Configuring the External BGP Peer Group Between the Loopback Interfaces on page 573
- Configuring the External BGP Peer Group Between the Inter-AS Link Interfaces on page 574
- Configuring the VPLS Routing Instances on page 578

Configuring Interfaces

Step-by-Step Procedure To configure interfaces:

1. On each router, configure an IP address on the loopback logical interface 0 (lo0.0):

   user@CE1# **set interfaces lo0 unit 0 family inet address 192.168.1.1/32 primary**

   user@PE1# **set interfaces lo0 unit 0 family inet address 192.168.2.1/32 primary**

   user@ASBR1# **set interfaces lo0 unit 0 family inet address 192.168.3.1/32 primary**

   user@ASBR2# **set interfaces lo0 unit 0 family inet address 192.168.10.1/32 primary**

   user@PE2# **set interfaces lo0 unit 0 family inet address 192.168.11.1/32 primary**

   user@CE2# **set interfaces lo0 unit 0 family inet address 192.168.12.1/32 primary**

2. On each router, commit the configuration:

   ```
   user@host> commit check

   configuration check succeeds
   ```

```
user@host> commit


commit complete
```

3. On each router, display the interface information for lo0 and verify that the correct IP address is configured:

```
user@host> show interfaces lo0


Physical interface: lo0, Enabled, Physical link is Up
  Interface index: 6, SNMP ifIndex: 6
  Type: Loopback, MTU: Unlimited
  Device flags   : Present Running Loopback
  Interface flags: SNMP-Traps
  Link flags     : None
  Last flapped   : Never
    Input packets : 0
    Output packets: 0

  Logical interface lo0.0 (Index 75) (SNMP ifIndex 16)
    Flags: SNMP-Traps Encapsulation: Unspecified
    Input packets : 0
    Output packets: 0
    Protocol inet, MTU: Unlimited
      Flags: None
      Addresses
        Local: 127.0.0.1
      Addresses, Flags: Primary Is-Default Is-Primary
        Local: 192.168.3.1
Logical interface lo0.16384 (Index 64) (SNMP ifIndex 21)
    Flags: SNMP-Traps Encapsulation: Unspecified
    Input packets : 0
    Output packets: 0
    Protocol inet, MTU: Unlimited
      Flags: None
      Addresses
        Local: 127.0.0.1

  Logical interface lo0.16385 (Index 65) (SNMP ifIndex 22)
    Flags: SNMP-Traps Encapsulation: Unspecified
    Input packets : 0
    Output packets: 0
    Protocol inet, MTU: Unlimited
      Flags: None
```

In the example above notice that the primary lo0 local address for the inet protocol family on ASBR1 is 192:168:3:1.

4. On each router, configure an IP address and protocol family on the Gigabit Ethernet interfaces. Specify the inet protocol family.

```
user@CE1# set interfaces ge-0/3/0 unit 0 family inet address 10.10.11.1/24
```

```
user@PE1# set interfaces ge-1/3/1 unit 0 family inet address 10.0.23.9/30

user@ASBR1# set interfaces ge-0/3/1 unit 0 family inet address
  10.0.23.10/30
user@ASBR1# set interfaces ge-0/3/0 unit 0 family inet address 10.0.78.1/30

user@ASBR2# set interfaces ge-3/1/0 unit 0 family inet address 10.0.78.2/30
user@ASBR2# set interfaces ge-3/1/1 unit 0 family inet address
  10.0.90.13/30

user@PE2# set interfaces ge-0/1/0 unit 0 family inet address 10.0.90.14/30

user@CE2# set interfaces ge-0/1/1 unit 0 family inet address 10.10.11.2/24
```

5. On each router, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

6. Display information for Gigabit Ethernet interfaces and verify that the IP address and protocol family are configured correctly.

```
user@ASBR2> show interfaces ge-* terse

Interface               Admin Link Proto    Local             Remote
ge-3/1/0                up    up
ge-3/1/0.0              up    up   inet     10.0.78.2/30
                                   multiservice
ge-3/1/1                up    up
ge-3/1/1.0              up    up   inet     10.0.90.13/30
                                   multiservice
ge-3/1/2                up    down
ge-3/1/3                up    down
```

Configuring OSPF

Step-by-Step Procedure To configure OSPF:

1. On the PE and ASBR routers, configure the provider instance of OSPF. Configure OSPF traffic engineering support. Specify area 0.0.0.1 in the LDP-signaled VPLS network and area 0.0.0.0 in the BGP-signaled network. Specify the Gigabit Ethernet logical interfaces between the PE and ASBR routers. Specify lo0.0 as a passive interface.

```
user@PE1# set protocols ospf traffic-engineering
user@PE1# set protocols ospf area 0.0.0.1 interface ge-1/3/1.0
user@PE1# set protocols ospf area 0.0.0.1 interface lo0.0 passive

user@ASBR1# set protocols ospf traffic-engineering
user@ASBR1# set protocols ospf area 0.0.0.1 interface ge-0/3/1.0
user@ASBR1# set protocols ospf area 0.0.0.1 interface lo0.0 passive

user@ASBR2# set protocols ospf traffic-engineering
user@ASBR2# set protocols ospf area 0.0.0.0 interface ge-3/1/1.0
user@ASBR2# set protocols ospf area 0.0.0.0 interface lo0.0 passive

user@PE2# set protocols ospf traffic-engineering
user@PE2# set protocols ospf area 0.0.0.0 interface ge-0/1/0.0
user@PE2# set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

2. On each router, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

3. Display OSPF neighbor information and verify that the PE routers form adjacencies with the ASBR router in the same area. Verify that the neighbor state is Full.

```
user@host> show ospf neighbor

Address          Interface         State     ID            Pri  Dead
10.0.23.10       ge-1/3/1.0        Full      192.168.3.1   128  31
```

Configuring the Internal BGP Peer Group

Step-by-Step Procedure

The purpose of configuring an internal BGP peer group is to create a full mesh of BGP LSPs among the PE routers in the BGP-signaled AS, including the ASBRs. To configure the internal BGP peer group:

1. The purpose of this step is to create a full mesh of IBGP peers between the PE routers, including the ASBRs, within the BGP-signaled AS.

   On ASBR2, configure internal BGP. Specify the BGP type as internal. Specify the local address as the local lo0 IP address.

Specify the inet protocol family. Specify the labeled-unicast statement and the resolve-vpn option. The labeled-unicast statement causes the router to advertise labeled routes out of the IPv4 inet.0 route table and places labeled routes into the inet.0 route table. The resolve-vpn option puts labeled routes in the MPLS inet.3 route table The inet.3 route table is used to resolve routes for the PE router located in the other AS.

Specify the l2vpn family to indicate to the router that this is a VPLS. Specify the signaling option to configure BGP as the signaling protocol. This enables BGP to carry Layer 2 VPLS NLRI messages for this peer group.

Specify the lo0 interface IP address of the PE as the neighbor. Configure an autonomous system identifier.

```
user@ASBR2# set protocols bgp group core-ibgp type internal
user@ASBR2# set protocols bgp group core-ibgp local-address 192.168.10.1
user@ASBR2# set protocols bgp group core-ibgp family inet labeled-unicast
  resolve-vpn
user@ASBR2# set protocols bgp group core-ibgp family l2vpn signaling
user@ASBR2# set protocols bgp group core-ibgp neighbor 192.168.11.1
user@ASBR2# set routing-options autonomous-system 0.65020
```

2. On PE2, configure internal BGP. Specify the BGP type as internal. Specify the local address as the local lo0 IP address.

   Specify the l2vpn family to indicate this is a VPLS. Specify the signaling option to configure BGP as the signaling protocol. This enables BGP to carry Layer 2 VPLS NLRI messages.

   Specify the lo0 interface IP address of ASBR2 as the neighbor. Configure an autonomous system identifier.

   ```
   user@PE2# set protocols bgp group core-ibgp type internal
   user@PE2# set protocols bgp group core-ibgp local-address 192.168.11.1
   user@PE2# set protocols bgp group core-ibgp family l2vpn signaling
   user@PE2# set protocols bgp group core-ibgp neighbor 192.168.10.1
   user@PE2# set routing-options autonomous-system 0.65020
   ```

3. On each router, commit the configuration:

   ```
   user@host> commit check

   configuration check succeeds

   user@host> commit

   commit complete
   ```

4. On PE2 and ASBR2, display BGP neighbor information and verify that the peer connection state is Established.

```
user@ASBR2> show bgp neighbor

Peer: 192.168.11.1+49443 AS 65020 Local: 192.168.10.1+179 AS 65020
  Type: Internal    State: Established    Flags: ImportEval Sync
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Options: Preference LocalAddress AddressFamily Rib-group Refresh
  Address families configured: l2vpn-signaling inet-labeled-unicast
  Local Address: 192.168.10.1 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 192.168.11.1     Local ID: 192.168.10.1     Active Holdtime:
90
  Keepalive Interval: 30         Peer index: 0

...
```

Configuring LDP

Step-by-Step Procedure To configure LDP:

1. On the PE and ASBR routers, configure LDP with the Gigabit Ethernet interfaces between the PE and ASBR routers, and between the two ASBRs. To support LDP-signaled VPLS, additionally configure LDP with the lo0.0 interface on PE1 and ASBR1:

```
user@PE1# set protocols ldp interface ge-1/3/1.0
user@PE1# set protocols ldp interface lo0.0

user@ASBR1# set protocols ldp interface ge-0/3/1.0
user@ASBR1# set protocols ldp interface ge-0/3/0.0
user@ASBR1# set protocols ldp interface lo0.0

user@ASBR2# set protocols ldp interface ge-3/1/0.0
user@ASBR2# set protocols ldp interface ge-3/1/1.0

user@PE2# set protocols ldp interface ge-0/1/0.0
```

NOTE: The configuration of LDP signaling between the ASBRs is not required for Inter-AS VPLS. It is included here for reference only and might be used in LDP environments.

2. On each router, commit the configuration:

```
user@host> commit check

configuration check succeeds
```

```
user@host> commit

commit complete
```

3. Display LDP configuration information and verify that the correct interfaces are configured. LDP operation can be verified after MPLS is configured.

```
user@ASBR1> show configuration protocols ldp

interface ge-0/3/0.0;
interface ge-0/3/1.0;
interface lo0.0;
```

The preceding example is from ASBR1.

Configuring MPLS

Step-by-Step Procedure To configure MPLS:

1. On the PE and ASBR routers, configure MPLS. Enable MPLS on the logical interfaces. Add the Gigabit Ethernet interfaces to the MPLS protocol. This adds entries to the MPLS forwarding table.

```
user@pe1# set protocols mpls interface ge-1/3/1.0
user@pe1# set interfaces ge-1/3/1 unit 0 family mpls

user@ASBR1# set protocols mpls interface ge-0/3/1.0
user@ASBR1# set protocols mpls interface ge-0/3/0.0
user@ASBR1# set interfaces ge-0/3/1 unit 0 family mpls
user@ASBR1# set interfaces ge-0/3/0 unit 0 family mpls

user@ASBR2# set protocols mpls interface ge-3/1/0.0
user@ASBR2# set protocols mpls interface ge-3/1/1.0
user@ASBR2# set interfaces ge-3/1/0 unit 0 family mpls
user@ASBR2# set interfaces ge-3/1/1 unit 0 family mpls

user@pe2# set protocols mpls interface ge-0/1/0.0
user@pe2# set interfaces ge-0/1/0 unit 0 family mpls
```

2. On each router, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit
```

```
commit complete
```

3. On the PE and ASBR routers, display LDP neighbor information and verify that the directly connected LDP neighbors are listed:

```
user@ASBR1> show ldp neighbor

Address            Interface          Label space ID         Hold time
192.168.2.1        lo0.0              192.168.2.1:0            44
10.0.78.2          ge-0/3/0.0         192.168.10.1:0           13
10.0.23.9          ge-0/3/1.0         192.168.2.1:0            11
```

The preceding example is from ASBR1.

Configuring the External BGP Peer Group Between the Loopback Interfaces

Step-by-Step Procedure To configure the external BGP (EBGP) peer group between the loopback interfaces:

1. On ASBR1 and PE1, configure an autonomous system identifier:

```
user@PE1# set routing-options autonomous-system 0.65010

user@ASBR1# set routing-options autonomous-system 0.65010
```

2. On ASBR1, configure an external BGP peer group for the loopback interfaces. Specify the external BGP group type. Include the multihop statement. Specify the local address as the local lo0 IP address. Configure the l2vpn family for BGP signaling. Configure the peer AS as the core AS number. Specify the lo0 IP address of ASBR2 as the neighbor.

```
user@ASBR1# set protocols bgp group vpls-core type external
user@ASBR1# set protocols bgp group vpls-core multihop
user@ASBR1# set protocols bgp group vpls-core local-address 192.168.3.1
user@ASBR1# set protocols bgp group vpls-core family l2vpn signaling
user@ASBR1# set protocols bgp group vpls-core peer-as 65020
user@ASBR1# set protocols bgp group vpls-core neighbor 192.168.10.1
```

3. On ASBR2, configure an external BGP peer group for the loopback interfaces. Specify the external BGP group type. Include the multihop statement. The multihop statement is needed because the EBGP neighbors are in different ASs. Specify the local address as the local lo0 IP address. Configure the l2vpn family for BGP signaling. Configure the peer AS as the metro AS number. Specify the lo0 IP address of ASBR1 as the neighbor.

```
user@ASBR2# set protocols bgp group vpls-metro type external
user@ASBR2# set protocols bgp group vpls-metro multihop
user@ASBR2# set protocols bgp group vpls-metro local-address 192.168.10.1
user@ASBR2# set protocols bgp group vpls-metro family l2vpn signaling
user@ASBR2# set protocols bgp group vpls-metro peer-as 65010
```

```
user@ASBR2# set protocols bgp group vpls-metro neighbor 192.168.3.1
```

4. On each router, commit the configuration:

```
user@host> commit
```

Configuring the External BGP Peer Group Between the Inter-AS Link Interfaces

Step-by-Step Procedure

The purpose of configuring external BGP peer groups between the inter-AS link interfaces is to create a full mesh of BGP LSPs among the ASBRs. To configure the external BGP peer group between the inter-AS link interfaces:

1. On ASBR1, configure a policy to export OSPF and direct routes, including the lo0 address of the PE routers, into BGP for the establishment of label-switched paths (LSPs):

```
user@ASBR1# set policy-options policy-statement loopback term term1 from
  protocol ospf
user@ASBR1# set policy-options policy-statement loopback term term1 from
  protocol direct
user@ASBR1# set policy-options policy-statement loopback term term1 from
  route-filter 192.168.0.0/16 longer
user@ASBR1# set policy-options policy-statement loopback term term1 then
  accept
```

2. On ASBR1, configure an external BGP peer group for the inter-AS link. Specify the external BGP group type. Specify the local inter-AS link IP address as the local address. Configure the inet family and include the labeled-unicast and resolve-vpn statements. The labeled-unicast statement advertises labeled routes out of the IPv4 inet.0 route table and places labeled routes into the inet.0 route table. The resolve-vpn option stores labeled routes in the MPLS inet.3 route table.

   Include the export statement and specify the policy you created. Configure the peer AS as the core AS number. Specify the inter-AS link IP address of ASBR2 as the neighbor.

```
user@ASBR1# set protocols bgp group metro-core type external
user@ASBR1# set protocols bgp group metro-core local-address 10.0.78.1
user@ASBR1# set protocols bgp group metro-core family inet labeled-unicast
  resolve-vpn
user@ASBR1# set protocols bgp group metro-core export loopback
user@ASBR1# set protocols bgp group metro-core peer-as 65020
user@ASBR1# set protocols bgp group metro-core neighbor 10.0.78.2
```

3. On ASBR2, configure a policy to export OSPF and direct routes, including the lo0 address, into BGP for the establishment of LSPs:

```
user@ASBR2# set policy-options policy-statement loopback term term1 from
  protocol ospf
```

```
user@ASBR2# set policy-options policy-statement loopback term term1 from
  protocol direct
user@ASBR2# set policy-options policy-statement loopback term term1 from
  route-filter 192.168.0.0/16 longer
user@ASBR2# set policy-options policy-statement loopback term term1 then
  accept
```

4. On ASBR2, configure an external BGP peer group for the inter-AS link. Specify the external BGP group type. Specify the local inter-AS link IP address as the local address. Configure the inet family and include the labeled-unicast and resolve-vpn statements. Include the export statement and specify the policy you created. Configure the peer AS as the core AS number. Specify the inter-AS link IP address of ASBR1 as the neighbor.

```
user@ASBR2# set protocols bgp group core-metro type external
user@ASBR2# set protocols bgp group core-metro local-address 10.0.78.2
user@ASBR2# set protocols bgp group core-metro family inet labeled-unicast
  resolve-vpn
user@ASBR2# set protocols bgp group core-metro export loopback
user@ASBR2# set protocols bgp group core-metro peer-as 65010
user@ASBR2# set protocols bgp group core-metro neighbor 10.0.78.1
```

5. On each router, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

6. On ASBR1, display the BGP neighbors. Verify that the first peer is the IP address of the Gigabit Ethernet interface of ASBR2. Verify that the second peer is the IP address of the lo0 interface of ASBR2. Also verify that the state of each peer is Established. Notice that on ASBR1 the NLRI advertised by ASBR2 the inter-AS link peer is inet-labeled-unicast and the NLRI advertised by ASBR2 the loopback interface peer is l2vpn-signaling.

```
user@ASBR1> show bgp neighbor

Peer: 10.0.78.2+65473 AS 65020 Local: 10.0.78.1+179 AS 65010
  Type: External    State: Established    Flags: Sync
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: Cease
  Export: [ loopback ]
  Options: Preference LocalAddress AddressFamily PeerAS Rib-group Refresh

  Address families configured: inet-labeled-unicast
  Local Address: 10.0.78.1 Holdtime: 90 Preference: 170
  Number of flaps: 3
```

```
  Last flap event: Stop
  Error: 'Cease' Sent: 1 Recv: 2
  Peer ID: 192.168.10.1     Local ID: 192.168.3.1      Active Holdtime:
90
  Keepalive Interval: 30         Peer index: 0
  BFD: disabled, down
  Local Interface: ge-0/3/0.0
  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)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-labeled-unicast
  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 65020)
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              2
    Received prefixes:            3
    Accepted prefixes:            3
    Suppressed due to damping:    0
    Advertised prefixes:          3
  Last traffic (seconds): Received 8    Sent 3    Checked 60
  Input messages:  Total 8713   Updates 3       Refreshes 0     Octets
165688
  Output messages: Total 8745   Updates 2       Refreshes 0     Octets
166315
  Output Queue[0]: 0

Peer: 192.168.10.1+51234 AS 65020 Local: 192.168.3.1+179 AS 65010
  Type: External    State: Established    Flags: Sync
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: Cease
  Options: Multihop Preference LocalAddress AddressFamily PeerAS Rib-group
 Refresh
  Address families configured: l2vpn-signaling
  Local Address: 192.168.3.1 Holdtime: 90 Preference: 170
  Number of flaps: 3
  Last flap event: Stop
  Error: 'Cease' Sent: 1 Recv: 2
  Peer ID: 192.168.10.1     Local ID: 192.168.3.1      Active Holdtime:
90
  Keepalive Interval: 30         Peer index: 0
  BFD: disabled, down
  NLRI for restart configured on peer: l2vpn-signaling
  NLRI advertised by peer: l2vpn-signaling
  NLRI for this session: l2vpn-signaling
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: l2vpn-signaling
  NLRI that restart is negotiated for: l2vpn-signaling
  NLRI of received end-of-rib markers: l2vpn-signaling
```

```
  NLRI of all end-of-rib markers sent: l2vpn-signaling
  Peer supports 4 byte AS extension (peer-as 65020)
  Table bgp.l2vpn.0 Bit: 20000
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
    Send state: in sync
    Active prefixes:              1
    Received prefixes:            1
    Accepted prefixes:            1
    Suppressed due to damping:    0
    Advertised prefixes:          1
  Table inter-as.l2vpn.0
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
    Send state: not advertising
    Active prefixes:              1
    Received prefixes:            1
    Accepted prefixes:            1
    Suppressed due to damping:    0
  Last traffic (seconds): Received 19   Sent 18   Checked 42
  Input messages:  Total 8712   Updates 3       Refreshes 0     Octets
165715
  Output messages: Total 8744   Updates 2       Refreshes 0     Octets
166342
  Output Queue[1]: 0
  Output Queue[2]: 0
```

7. On ASBR2, display the BGP summary. Notice that the first peer is the IP address of the Gigabit Ethernet interface of ASBR1, the second peer is the IP address of the lo0 interface of ASBR1, and the third peer is the lo0 interface of PE2. Verify that the state of each peer is Established.

```
user@ASBR2> show bgp summary

Groups: 3 Peers: 3 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State
Pending
inet.0                 3          2          0          0          0
      0
bgp.l2vpn.0            2          2          0          0          0
      0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last
Up/Dwn State|#Active/Received/Accepted/Damped...
10.0.78.1             65010       8781       8748       0       2 2d
17:54:56 Establ
  inet.0: 2/3/3/0
192.168.3.1           65010       8780       8747       0       2 2d
17:54:54 Establ
  bgp.l2vpn.0: 1/1/1/0
  inter-as.l2vpn.0: 1/1/1/0
192.168.11.1          65020       8809       8763       0       1 2d
17:59:22 Establ
  bgp.l2vpn.0: 1/1/1/0
  inter-as.l2vpn.0: 1/1/1/0
```

8. On PE2, display the BGP group. Verify that the peer is the IP address of the lo0 interface of ASBR2. Verify that the number of established peer sessions is 1.

```
user@PE1> show bgp group

Group Type: Internal    AS: 65020                  Local AS: 65020
  Name: core-ibgp       Index: 1                   Flags: Export Eval
  Holdtime: 0
  Total peers: 1        Established: 1
  192.168.10.1+179
  bgp.l2vpn.0: 1/1/1/0
  inter-as.l2vpn.0: 1/1/1/0

Groups: 1  Peers: 1    External: 0    Internal: 1    Down peers: 0   Flaps:
 7
Table          Tot Paths  Act Paths Suppressed    History Damp State
Pending
bgp.l2vpn.0            1          1          0          0          0
       0
inte.l2vpn.0           1          1          0          0          0
       0
```

Configuring the VPLS Routing Instances

Step-by-Step Procedure To configure the VPLS routing instances:

1. On PE1, configure the VPLS routing instance. To enable a VPLS instance, specify the vpls instance type. Configure VPLS on the CE-facing Gigabit Ethernet interface. Configure the CE-facing interface to use ethernet-vpls encapsulation.

```
user@PE1# set routing-instances metro instance-type vpls
user@PE1# set routing-instances metro interface ge-1/3/0.0
```

2. On PE1, configure the VPLS protocol within the routing instance. To uniquely identify the virtual circuit, configure the VPLS identifier. The VPLS identifier uniquely identifies each VPLS in the router. Configure the same VPLS ID on all the routers for a given VPLS.

   Specify the IP address of the lo0 interface on ASBR2 as the neighbor.

   Configure the CE-facing interface to use ethernet-vpls encapsulation and the vpls protocol family.

```
user@PE1# set routing-instances metro protocols vpls vpls-id 101
user@PE1# set routing-instances metro protocols vpls neighbor 192.168.3.1
user@PE1# set interfaces ge-1/3/0 encapsulation ethernet-vpls
user@PE1# set interfaces ge-1/3/0 unit 0 family vpls
```

3. On ASBR1, configure the VPLS routing instance. To enable a VPLS instance, specify the vpls instance type. Configure a route distinguisher and a VRF target. The vrf-target statement causes default VRF import and export policies to be generated that accept and tag routes with the specified target community.

NOTE: A route distinguisher allows the router to distinguish between two identical IP prefixes used as VPN routes. Configure a different route distinguisher on each ASBR router.

NOTE: You must configure the same VRF target on both ASBR routers.

```
user@ASBR1# set routing-instances inter-as instance-type vpls
user@ASBR1# set routing-instances inter-as route-distinguisher 65010:1
user@ASBR1# set routing-instances inter-as vrf-target target:2:1
```

4. On ASBR1, configure the VPLS protocol within the routing instance.

   Configure the VPLS identifier. Specify the IP address of the lo0 interface on PE1 as the neighbor.

```
user@ASBR1# set routing-instances inter-as protocols vpls vpls-id 101
user@ASBR1# set routing-instances inter-as protocols vpls neighbor
  192.168.2.1
```

NOTE: The VPLS identifier uniquely identifies each LDP-signaled VPLS in the router. Configure the same VPLS ID on PE1 and ASBR1.

5. On ASBR1, configure the VPLS site within the routing instance. Configure the site identifier as required by the protocol to establish the EBGP pseudowire. As a best practice for more complex topologies involving multihoming, configure a site preference.

```
user@ASBR1# set routing-instances inter-as protocols vpls site ASBR-metro
  site-identifier 1
user@ASBR1# set routing-instances inter-as protocols vpls site ASBR-metro
  site-preference 10000
```

6. On ASBR1, configure the VPLS mesh group peer-as statement within the routing instance to specify which ASs belong to this AS mesh group. Configure the peer AS for the mesh group as all.

   This statement enables the router to establish a single pseudowire between the ASBRs. VPLS NLRI messages are exchanged across the EBGP sessions on the inter-AS links between the ASBRs. All autonomous systems are in one mesh group.

```
user@ASBR1# set routing-instances inter-as protocols vpls mesh-group metro
  peer-as all
```

7. On ASBR2, configure the VPLS routing instance. To enable a VPLS instance, specify the vpls instance type. Configure a route distinguisher and a VRF target.

The vrf-target statement causes default VRF import and export policies to be generated that accept and tag routes with the specified target community.

NOTE: A route distinguisher allows the router to distinguish between two identical IP prefixes used as VPN routes. Configure a different route distinguisher on each ASBR router.

NOTE: You must configure the same VRF target community on both ASBR routers.

```
user@ASBR2# set routing-instances inter-as instance-type vpls
user@ASBR2# set routing-instances inter-as route-distinguisher 65020:1
user@ASBR2# set routing-instances inter-as vrf-target target:2:1
```

8. On ASBR2, configure the VPLS site within the routing instance. Configure the site identifier as required by the protocol.

```
user@ASBR2# set routing-instances inter-as protocols vpls site ASBR-core
    site-identifier 2
```

9. On ASBR2, configure the VPLS mesh group within the routing instance to specify which VPLS PEs belong to this AS mesh group. Configure the peer AS for the mesh group as all.

   This statement enables the router to establish a single pseudowire between the ASBRs. VPLS NLRI messages are exchanged across the EBGP sessions on the inter-AS links between the ASBRs. All autonomous systems are in one mesh group.

```
user@ASBR1# set routing-instances inter-as protocols vpls mesh-group core
    peer-as all
```

10. On PE2, configure the VPLS routing instance. To enable a VPLS instance, specify the vpls instance type. Configure VPLS on the CE-facing Gigabit Ethernet interface. Configure a route distinguisher and a VRF target.

```
user@PE2# set routing-instances inter-as instance-type vpls
user@PE2# set routing-instances inter-as interface ge-0/1/1.0
user@PE2# set routing-instances inter-as route-distinguisher 65020:1
user@PE2# set routing-instances inter-as vrf-target target:2:1
```

11. On PE2, configure the VPLS site within the routing instance. Configure the site identifier as required by the protocol.

    Configure the CE-facing interface to use ethernet-vpls encapsulation and the vpls protocol family.

```
user@PE2# set routing-instances inter-as protocols vpls site PE2 site-identifier
    3
user@PE2# set interfaces ge-0/1/1 encapsulation ethernet-vpls
user@PE2# set interfaces ge-0/1/1 unit 0 family vpls
```

12. On each router, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

13. On the PE routers, display the CE-facing Gigabit Ethernet interface information and verify that the encapsulation is configured correctly:

```
user@host> show interfaces ge-1/3/0

Address            Interface          Label space ID         Hold time
10.0.23.10         ge-1/3/1.0         192.168.3.1:0            11

Physical interface: ge-1/3/0, Enabled, Physical link is Up
  Interface index: 147, SNMP ifIndex: 145
  Link-level type: Ethernet, MTU: 1514, Speed: 1000mbps, MAC-REWRITE Error:
 None,
  Loopback: Disabled, Source filtering: Disabled, Flow control: Enabled,
  Auto-negotiation: Enabled, Remote fault: Online
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 4 supported, 4 maximum usable queues
  Schedulers     : 256
  Current address: 00:12:1e:ee:34:db, Hardware address: 00:12:1e:ee:34:db

  Last flapped   : 2008-08-27 19:02:52 PDT (5d 22:32 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Ingress rate at Packet Forwarding Engine      : 0 bps (0 pps)
  Ingress drop rate at Packet Forwarding Engine : 0 bps (0 pps)
  Active alarms  : None
  Active defects : None

  Logical interface ge-1/3/0.0 (Index 84) (SNMP ifIndex 146)
    Flags: SNMP-Traps Encapsulation: ENET2
    Input packets : 0
    Output packets: 1
    Protocol inet, MTU: 1500
      Flags: None
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 10.10.11/24, Local: 10.10.11.11, Broadcast:
10.10.11.255
```

Results The relevant sample configuration for the CE1 router follows.

Router CE1

```
interfaces {
    lo0 {
```

```
        unit 0 {
          family inet {
            address 192.168.1.1/32 {
              primary;
            }
            address 127.0.0.1/32;
          }
        }
      }
      ge-0/3/0 {
        unit 0 {
          family inet {
            address 10.10.11.1/24;
          }
        }
      }
    }
```

The relevant sample configuration for the PE1 router follows.

Router PE1

```
interfaces {
  lo0 {
    unit 0 {
      family inet {
        address 192.168.2.1/32 {
          primary;
        }
        address 127.0.0.1/32;
      }
    }
  }
  ge-1/3/0 {
    encapsulation ethernet-vpls;
    unit 0 {
      family vpls;
    }
  }
  ge-1/3/1 {
    unit 0 {
      family inet {
        address 10.0.23.9/30;
      }
      family mpls;
    }
  }
}
routing-options {
  autonomous-system 0.65010;
}
protocols {
  mpls {
    interface ge-1/3/1.0;
  }
  ospf {
    traffic-engineering;
```

```
        area 0.0.0.1 {
            interface ge-1/3/1.0;
            interface lo0.0 {
                passive;
            }
        }
    }
    ldp {
        interface ge-1/3/1.0;
        interface lo0.0;
    }
}
routing-instances {
    metro {
        instance-type vpls;
        interface ge-1/3/0.0;
        protocols {
            vpls {
                vpls-id 101;
                neighbor 192.168.3.1;
            }
        }
    }
}
```

The relevant sample configuration for the ASBR1 router follows.

Router ASBR1

```
interfaces {
    lo0 {
    unit 0 {
        family inet {
            address 192.168.3.1/32 {
                primary;
            }
            address 127.0.0.1/32;
        }
        }
    }
    }
    ge-0/3/0 {
        unit 0 {
            family inet {
                address 10.0.78.1/30;
            }
            family mpls;
        }
    }
    ge-0/3/1 {
        unit 0 {
            family inet {
                address 10.0.23.10/30;
            }
            family mpls;
        }
    }
```

```
}
routing-options {
   autonomous-system 0.65010;
}
protocols {
   mpls {
      interface ge-0/3/1.0;
      interface ge-0/3/0.0;
   }
   bgp {
      group vpls-core {
         type external;
         multihop;
         local-address 192.168.3.1;
         family l2vpn {
            signaling;
         }
         peer-as 65020;
         neighbor 192.168.10.1;
      }
      group metro-core {
         type external;
         local-address 10.0.78.1;
         family inet {
            labeled-unicast {
               resolve-vpn;
            }
         }
         export loopback;
         peer-as 65020;
         neighbor 10.0.78.2;
      }
   }
   ospf {
      traffic-engineering;
      area 0.0.0.1 {
         interface ge-0/3/1.0;
         interface lo0.0 {
            passive;
         }
      }
   }
   ldp {
      interface ge-0/3/0.0;
      interface ge-0/3/1.0;
      interface lo0.0;
   }
}
policy-options {
   policy-statement loopback {
      term term1 {
         from {
            protocol [ ospf direct ];
            inactive: route-filter 10.0.0.0/8 longer;
            route-filter 192.168.0.0/16 longer;
         }
```

```
            then accept;
        }
    }
}
routing-instances {
    inter-as {
        instance-type vpls;
        route-distinguisher 65010:1;
        vrf-target target:2:1;
        protocols {
            vpls {
                site ASBR-metro {
                    site-identifier 1;
                    site-preference 10000;
                }
                vpls-id 101;
                neighbor 192.168.2.1;
                mesh-group metro {
                    peer-as {
                        all;
                    }
                }
            }
        }
    }
}
```

The relevant sample configuration for the ASBR2 router follows.

Router ASBR2

```
interfaces {
    lo0 {
        unit 0 {
            family inet {
                address 192.168.10.1/32 {
                    primary;
                }
                address 127.0.0.1/32;
            }
        }
    }
    ge-3/1/0 {
        unit 0 {
            family inet {
                address 10.0.78.2/30;
            }
            family mpls;
        }
    }
    ge-3/1/1 {
        unit 0 {
            family inet {
                address 10.0.90.13/30;
            }
            family mpls;
        }
```

```
    }
}
routing-options {
    autonomous-system 0.65020;
}
protocols {
    mpls {
        interface ge-3/1/0.0;
        interface ge-3/1/1.0;
    }
    bgp {
        group core-ibgp {
            type internal;
            local-address 192.168.10.1;
            family inet {
                labeled-unicast {
                    resolve-vpn;
                }
            }
            family l2vpn {
                signaling;
            }
            neighbor 192.168.11.1;
        }
        group vpls-metro {
            type external;
            multihop;
            local-address 192.168.10.1;
            family l2vpn {
                signaling;
            }
            peer-as 65010;
            neighbor 192.168.3.1;
        }
        group core-metro {
            type external;
            local-address 10.0.78.2;
            family inet {
                labeled-unicast {
                    resolve-vpn;
                }
            }
            export loopback;
            peer-as 65010;
            neighbor 10.0.78.1;
        }
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface ge-3/1/1.0;
            interface lo0.0 {
                passive;
            }
        }
    }
```

```
    ldp {
        interface ge-3/1/0.0;
        interface ge-3/1/1.0;
    }
}
policy-options {
    policy-statement loopback {
        term term1 {
            from {
                protocol [ ospf direct ];
                route-filter 192.168.0.0/16 longer;
            }
            then accept;
        }
    }
}
routing-instances {
    inter-as {
        instance-type vpls;
        route-distinguisher 65020:1;
        vrf-target target:2:1;
        protocols {
            vpls {
                site ASBR-core {
                    site-identifier 2;
                }
                mesh-group core {
                    peer-as {
                        all;
                    }
                }
            }
        }
    }
}
```

The relevant sample configuration for the PE2 router follows.

Router PE2

```
interfaces {
    lo0 {
        unit 0 {
            family inet {
                address 192.168.11.1/32 {
                    primary;
                }
                address 127.0.0.1/32;
            }
        }
    }
    ge-0/1/0 {
        unit 0 {
            family inet {
                address 10.0.90.14/30;
            }
            family mpls;
```

```
        }
    }
    ge-0/1/1 {
        encapsulation ethernet-vpls;
        unit 0 {
            family vpls;
        }
    }
}
routing-options {
    autonomous-system 0.65020;
}
protocols {
    mpls {
        interface ge-0/1/0.0;
    }
    bgp {
        group core-ibgp {
            type internal;
            local-address 192.168.11.1;
            family l2vpn {
                signaling;
            }
            neighbor 192.168.10.1;
        }
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface ge-0/1/0.0;
            interface lo0.0 {
                passive;
            }
        }
    }
    ldp {
        interface ge-0/1/0.0;
    }
}
routing-instances {
    inter-as {
        instance-type vpls;
        interface ge-0/1/1.0;
        route-distinguisher 65020:1;
        vrf-target target:2:1;
        protocols {
            vpls {
                site PE2 {
                    site-identifier 3;
                }
            }
        }
    }
}
```

The relevant sample configuration for the CE2 router follows.

Router CE2

```
interfaces {
    lo0 {
        unit 0 {
            family inet {
                address 192.168.12.1/32 {
                    primary;
                }
                address 127.0.0.1/32;
            }
        }
    }
    ge-0/1/1 {
        unit 0 {
            family inet {
                address 10.10.11.2/24;
            }
        }
    }
}
```

Verification

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

- Verifying VPLS Connections on page 589
- Verifying End-to-End Traffic Flow on page 591

Verifying VPLS Connections

Purpose To verify the VPLS connections have been established, enter the following command on the ASBR and PE routers.

Action

```
user@PE1> show vpls connections
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 availble
BK -- Backup connection          ST -- Standby connection

Legend for interface status
Up -- operational
Dn -- down
```

```
Instance: metro
  VPLS-id: 101
    Neighbor                Type  St     Time last up          # Up trans
    192.168.3.1(vpls-id 101)  rmt   Up     Sep  9 14:05:18 2008           1
      Remote PE: 192.168.3.1, Negotiated control-word: No
      Incoming label: 800001, Outgoing label: 800000
      Local interface: vt-1/2/0.1048576, Status: Up, Encapsulation: ETHERNET
        Description: Intf - vpls metro neighbor 192.168.3.1 vpls-id 101

user@ASBR1> show vpls connections
...
Instance: inter-as
  BGP-VPLS State
  Mesh-group connections: metro
    Neighbor         Local-site   Remote-site   St      Time last up
    192.168.10.1     1            2             Up      Sep  8 20:16:28 2008
      Incoming label: 800257, Outgoing label: 800000
      Local interface: vt-1/2/0.1049088, Status: Up, Encapsulation: VPLS
  LDP-VPLS State
  VPLS-id: 101
  Mesh-group connections: __ves__
    Neighbor                Type  St     Time last up          # Up trans
    192.168.2.1(vpls-id 101)  rmt   Up     Sep  9 14:05:22 2008           1
      Remote PE: 192.168.2.1, Negotiated control-word: No
      Incoming label: 800000, Outgoing label: 800001
      Local interface: vt-0/1/0.1049089, Status: Up, Encapsulation: ETHERNET
        Description: Intf - vpls inter-as neighbor 192.168.2.1 vpls-id 101

user@ASBR2> show vpls connections
...
Instance: inter-as
  BGP-VPLS State
  Mesh-group connections: __ves__
    Neighbor         Local-site   Remote-site   St      Time last up
    192.168.11.1     2            3             Up      Sep 11 15:18:23 2008
      Incoming label: 800002, Outgoing label: 800001
      Local interface: vt-4/0/0.1048839, Status: Up, Encapsulation: VPLS
  Mesh-group connections: core
    Neighbor         Local-site   Remote-site   St      Time last up
    192.168.3.1      2            1             Up      Sep  8 20:16:28 2008
      Incoming label: 800000, Outgoing label: 800257
      Local interface: vt-4/0/0.1048834, Status: Up, Encapsulation: VPLS

user@PE2> show vpls connections
...
Instance: inter-as
  Local site: PE2 (3)
    connection-site           Type  St     Time last up          # Up trans
    2                         rmt   Up     Sep  8 20:16:28 2008           1
      Remote PE: 192.168.10.1, Negotiated control-word: No
      Incoming label: 800001, Outgoing label: 800002
      Local interface: vt-0/3/0.1048832, Status: Up, Encapsulation: VPLS
        Description: Intf - vpls inter-as local site 3 remote site 2
```

Meaning In the display from PE1, notice that the neighbor is the lo0 address of ASBR1 and that the status is up.

In the display from ASBR1, notice that the neighbor is the lo0 address of PE1 and that the status is up.

In the display from ASBR2, notice that the neighbor is the lo0 address of PE2 and that the status is up.

In the display from PE2, notice that the neighbor is the lo0 address of ASBR2 and that the status is up.

Verifying End-to-End Traffic Flow

Purpose To verify that the CEs can send and receive traffic across the VPLS, use the ping command.

Action

```
user@CE1> ping 10.10.11.2
PING 10.10.11.2 (10.10.11.2): 56 data bytes
64 bytes from 10.10.11.2: icmp_seq=0 ttl=64 time=1.369 ms
64 bytes from 10.10.11.2: icmp_seq=1 ttl=64 time=1.360 ms
64 bytes from 10.10.11.2: icmp_seq=2 ttl=64 time=1.333 ms
^C

user@CE2> ping 10.10.11.1
PING 10.10.11.1 (10.10.11.1): 56 data bytes
64 bytes from 10.10.11.1: icmp_seq=0 ttl=64 time=6.209 ms
64 bytes from 10.10.11.1: icmp_seq=1 ttl=64 time=1.347 ms
64 bytes from 10.10.11.1: icmp_seq=2 ttl=64 time=1.324 ms
^C
```

Meaning If CE1 can send and receive traffic from CE2 and CE2 can send and receive traffic from CE1, the VPLS is performing correctly.

For More Information

For additional information about VPLS, see the following:

- *JUNOS VPNs Configuration Guide*
- *JUNOS Network Interfaces Configuration Guide*
- *JUNOS Class of Service Configuration Guide*
- *JUNOS Routing Protocols Configuration Guide*
- *JUNOS Routing Protocols and Policies Command Reference*
- *JUNOS Interfaces Command Reference*
- RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*
- RFC 4762, *Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling*
- RFC 4761, *Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling*

Revision History

5 October 2009—10.0R1 Release. Roy Spencer.

13 June 2009—9.6R1 Release. Roy Spencer.

13 April 2009—9.5R1 Release. Roy Spencer.

15 January 2009—9.4R1 Release. Roy Spencer.

10 October 2008—9.3R1 Release. Roy Spencer.

10 July 2008—9.2R1 Release. Roy Spencer.

10 April 2008—9.1R1 Release. Added support for VPLS nonstop active routing. Roy Spencer.

1 February 2008—Added support for VPLS interworking between LDP and BGP signaling. Added support for 802.1p classification with Ethernet VPLS over ATM LLC interface encapsulation. Added support for VSTP. Added support for Layer 2 VPLS filters for MX960 router. 9.0R1 Release. Fawn Damitio.

29 June 2007—Added support for LDP signaling and MX Series routers. 8.4R1 Release. Fawn Damitio.

12 January 2007—Added configuration information for aggregated Ethernet and multihoming feature. Added support for MX960 Ethernet Services Router, 8.2R1 Release. Fawn Damitio.

15 September 2006—8.1R1 Release. Richard Hendricks.

29 June 2006—Added limitations for MSTP and logical routers, 8.0R1 Release. Richard Hendricks.

27 March 2006—Added support for VPLS on LSI logical interfaces, 7.6R1 Release. Richard Hendricks.

9 January 2006—Added support for multihoming a CE router to multiple PE routers, 7.5R1 Release. Richard Hendricks.

14 September 2005—7.4R1 Release. Richard Hendricks.

13 June 2005—Added VPLS per-packet load balancing, support for limiting MAC address learning per interface in a VPLS domain, and migration to the new VPLS and Layer 2 VPN signaling statement at the [edit protocols bgp groups *group-name* family l2vpn] hierarchy level, 7.3R1 Release. Richard Hendricks.

5 April 2005—7.2R1 Release. Richard Hendricks.

2 February 2005—7.1R1 Release. Richard Hendricks.

6 October 2004—7.0R1 Release. Richard Hendricks.

6 July 2004—Added support for Ethernet VPLS over ATM LLC interface encapsulation on T Series and M320 routers, the show vpls statistics command, and manual selection of tunnel-enabled PICs used to provide virtual ports for VPLS operation, 6.4R1 Release. Richard Hendricks.

5 April 2004—Updated lt interface families and encapsulation types and added new commands to clear MAC addresses from the VPLS table and modify the VPLS table timeout intervals, 6.3R1 Release. Richard Hendricks.

21 January 2004—Added new PIC support for VPLS. Richard Hendricks.

22 December 2003—Added VPLS CoS, VPLS graceful restart, VPLS interinstance bridging and routing, VPLS support on the T Series routers, and operational mode commands for VPLS source and destination MAC accounting, 6.2R1 Release. Richard Hendricks.

22 September 2003—Added VPLS policers and filters, 6.1R1 Release. Richard Hendricks.

30 June 2003—Added the ether-vpls-over-atm-llc interface encapsulation type and LSP selection for VPLS instances, 6.0R1 Release. Elizabeth Lichtenberg and Richard Hendricks.

2 April 2003—Initial document written, 5.7R1. Richard Hendricks.

Part 4
Index

■ Index on page 597

Index

Symbols

#, comments in configuration statements.....xxviii
( ), in syntax descriptions.....xxviii
< >, in syntax descriptions.....xxviii
[ ], in configuration statements.....xxviii
{ }, in configuration statements.....xxviii
| (pipe), in syntax descriptions.....xxviii

A

APS
    Layer 2 circuits.....297
ATM encapsulation
    Layer 2 switching cross-connects.....477

B

bootstrap IPv4 messages.....355
braces, in configuration statements.....xxviii
brackets
    angle, in syntax descriptions.....xxviii
    square, in configuration statements.....xxviii

C

Cisco HDLC encapsulation
    Layer 2 switching cross-connect.....477
comments, in configuration statements.....xxviii
connections statement
    usage guidelines.....480
conventions
    text and syntax.....xxvii
curly braces, in configuration statements.....xxviii
customer support.....xxix
    contacting JTAC.....xxix

D

documentation
    comments on.....xxix

E

encapsulation
    TCC.....477
encapsulation statement
    Layer 2 switching cross-connect.....477

F

font conventions.....xxvii
Frame Relay encapsulation
    Layer 2 switching cross-connect.....477

G

GMPLS
    configuration procedure.....9
    example configuration
        static peers.....19
    operational mode commands.....23
    options
        administratively down nonpacket LSPs.....14
        graceful restart.....15
        graceful teardown.....14
        LMP control channel.....16
        overlay model.....15
        peer model.....15
    overview.....4
    sample configuration
        LMP control channel.....31
    system requirements.....6
graceful restart
    options
        PIM sparse mode in a Layer 3 VPN.....445

I

icons defined, notice.....xxvii
instances
    route sharing
        configuration procedure.....169
        example configuration.....175, 185
        operational mode commands.....181, 188
        overview.....167

interinstance route sharing
configuration procedure....169
example configuration....175, 185
operational mode commands....181, 188
overview....167
IPv6
tunneling over MPLS
configuration procedure....46
example configuration....48
operational mode commands....54
overview....43
system requirements....45

L

Layer 2 circuits
configuration procedure....290
example configuration
APS....334
ATM2 IQ....315
SONET/SDH....310
traffic engineering....324
operational mode commands
ATM2 IQ....321
Ethernet....307
SONET/SDH....314
traffic engineering....333
options
APS....297
control word....297
local interface switching....302
MTU....301
reserving LSP bandwidth....300
simultaneous RSVP and LDP LSPs....302
traffic engineering....296
trunk mode for ATM2 IQ interfaces....298
overview....286
system requirements....289
Layer 2 switching
TCC
configuration procedure....476
example configuration....481, 483
overview....473
system requirements....474
Layer 2 switching cross-connect
MPLS....480
TCC connections....480
TCC encapsulation....477
Layer 2.5 VPNs
TCC
configuration procedure....485
example configuration....488
operational mode commands....493
Layer 3 VPNs
multicast
configuration procedure....442
example configuration....448, 461
MDT....445
MSDP....446
operational mode commands....360, 451, 465
overview....340
PIM sparse mode graceful restart....445
point-to-multipoint LSPs....350
system requirements....342
traffic engineering....350
Layer 2 circuits
example configuration
Ethernet....303
operational mode commands
APS....335
LDP
multiple instances
example configuration....63
operational mode commands....82
overview....61
system requirements....62
Link Management Protocol *See* LMP
link protection
configuration procedure....106
example configuration....110
operational mode commands....115
options
class of service....108
enhanced operational mode commands....109
manual bypass LSPs....107
multiple bypass LSPs....107
priority....107
system log messages....109
overview....101
system requirements....105
LMP....16
control channel....16
See also GMPLS
logical routers *See* logical systems
logical systems....195
configuration procedure....200
operational mode commands....224
options
DHCP....203
filter-based forwarding....203
logical tunnel (lt) interface....202
selective view for operational mode....203
overview....195
sample configuration....205
system requirements....199

M

manuals
  comments on....xxix
MBGP
  for multicast VPNs....385
MBGP MVPN
  IPv6 transport
    configuring....355
  pim-asm provider tunnels
    configuring....389
  pim-sm provider tunnels
    configuring....358
  rsvp-te provider tunnels
    configuring....358
  sender and receiver sites
    configuring....358, 389
MDT
  Layer 3 VPNs....445
MPLS
  Layer 2 switching cross-connect....480
  link protection
    configuration procedure....106
    example configuration....110
    operational mode commands....115
    overview....101
    system requirements....105
  node-link protection
    configuration procedure....106
    example configuration....129
    operational mode commands....135
    overview....101
    system requirements....105
MSDP
  Layer 3 VPNs....446
multicast
  Layer 3 VPNs
    configuration procedure....442
    example configuration....448, 461
    MDT....445
    MSDP....446
    operational mode
      commands....360, 451, 465
    overview....340
    PIM sparse mode graceful restart....445
    point-to-multipoint LSPs....350
    system requirements....342
multicast distribution trees *See* MDT
multicast VPN extranets
  applications....397
  configuration guidelines....398
  configuring....398
  overview....397
multicast VPNs
  using MBGP....385
multihoming
  VPLS
    configuration....514, 516
multihoming, VPLS
  configuration....514, 516
Multiprotocol BGP-Based Multicast VPNs
  configuration procedure....344
MVPNs *See* multicast VPNs

N

node-link protection
  configuration procedure....106
  example configuration....129
  operational mode commands....135
  overview....101
  system requirements....105
nonstop active routing
  enabling....534
  example configuration....563
  trace options....536
  verifying status of 536
notice icons defined....xxvii

O

OSPFv3
  configuration procedure....241
  example configuration....243
  operational mode commands....249
  overview....239
  system requirements....241

P

parentheses, in syntax descriptions....xxviii
PIM
  bootstrap router....355

R

RSVP LSP tunnels
  configuration procedure....143
  example configuration....147
  operational mode commands....159
  options
    graceful teardown....146
  overview....141
  system requirements....142

S

simplified interinstance route sharing
  configuration procedure....169
  example configuration....175, 185
  operational mode commands....181, 188

overview....167
system requirements....168
support, technical *See* technical support
synchronizing Routing Engines
nonstop active routing....535
syntax conventions....xxvii
system requirements
GMPLS....6
IPv6 tunneling over IPv4 MPLS....45
Layer 2 circuits....289
link protection....105
logical systems....199
multicast Layer 3 VPNs point-to-multipoint LSPs....350
multicast over Layer 3 VPNs....342
multiple instance LDP....62
OSPFv3....241
RSVP LSP tunnels....142
simplified interinstance route sharing....168
TCC....474
VPLS....505

T

TCC
configuration procedure....476
encapsulation....486
example configuration
Frame Relay to Fast Ethernet....483
PPP to ATM....481
Layer 2.5 VPNs
configuration procedure....485
example configuration....488
operational mode commands....493
operational mode commands
Frame Relay to Fast Ethernet....485
PPP to ATM....482
options
static ARP....479
overview....473
system requirements....474
technical support
contacting JTAC....xxix
traffic engineering
Layer 2 circuits....296

V

VPLS
configuration procedure....507
multihoming, configuration....516
operational mode commands....544, 559
options
class of service....528
clearing MAC addresses....529
configuring VPLS without a tunnel PIC....522, 523
graceful restart....528
interface MAC address limits....532
interinstance bridging and routing....530
manually selecting virtual port PICs....531
multihoming....516
multihoming for the area border router....514
MX Series tunnel-services....523
per-packet load balancing....533
policers and filters....525
selecting an LSP for the VPLS instance....515
table timeout interval....529
overview....502
requirements
interface encapsulation....508
interworking between signaling protocols....512
routing protocols....508
sample configuration....538, 549
system requirements....505