JUNIPER NETWORKS | Engineering Simplicity

Junos® OS

Logical Systems User Guide for Routers and Switches

Published
2022-06-13

JUNOS

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

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

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

Junos® OS Logical Systems User Guide for Routers and Switches
Copyright © 2022 Juniper Networks, Inc. All rights reserved.

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

Table of Contents

About This Guide | ix

1 Overview

Understanding Logical Systems for Routers and Switches | 2

- Comparing Junos OS Device Virtualization Technologies | 2
- Introduction to Logical Systems | 4
- Logical Systems Applications | 5
- Junos OS Features That Are Supported on Logical Systems | 6
- Logical Systems System Requirements | 8
- Logical Systems Operations and Restrictions | 9

2 Configuring Logical Systems

User Access on Logical Systems | 14

- Example: Configuring Logical System Administrators | 14
  - Requirements | 14
  - Overview | 14
  - Configuration | 15
  - Verification | 17

Setting Up Logical Systems | 17

- Using Junos OS to Configure Logical System Administrators | 18
- Example: Creating an Interface on a Logical System | 19
  - Requirements | 19
  - Overview | 20
  - Configuration | 21
  - Verification | 22
- Configuring Logical System Interface Properties | 22
- Example: Connecting a Logical System to a Physical Router | 23
  - Requirements | 24

Overview | 24

Configuration | 25

Verification | 26

Example: Connecting Logical Systems Within the Same Device Using Logical Tunnel Interfaces on MX Series Routers and EX Series Switches | 27

Requirements | 28

Overview | 28

Configuration | 29

Verification | 31

Example: Connecting Logical Systems Within the Same Router Using Logical Tunnel Interfaces | 33

Requirements | 33

Overview | 34

Configuration | 35

Verification | 38

Static Routing on Logical Systems | 40

Example: Configuring Static Routes Between Logical Systems Within the Same Router | 40

Requirements | 40

Overview | 40

Configuration | 42

Verification | 46

Firewall Filters on Logical Systems | 49

Understanding How to Use Standard Firewall Filters | 49

Example: Configuring a Stateless Firewall Filter to Protect a Logical System Against ICMP Floods | 51

Requirements | 51

Overview | 51

Configuration | 52

Verification | 55

Example: Configuring Filter-Based Forwarding on Logical Systems | 57

Requirements | 57

Overview | 57

Configuration | 60

Verification | 68

Layer 2 Learning and Forwarding on Logical Systems | 71

Layer 2 Learning and Forwarding in a Logical System Overview | **71**

Enabling Layer 2 Learning and Forwarding in a Logical System | **72**

Example: Configuring Layer 2 Learning and Forwarding and RSTP in a Logical System | **73**

Multichassis Link Aggregation on Logical Systems Overview | 75

VPNs and VPLS | 80

Introduction to VPLS | **81**

Example: Using Logical Systems to Configure Provider Edge and Provider Routers in a Layer 3 VPN and VPLS Scenario | **82**

Requirements | **82**

Overview | **82**

Configuration | **84**

Verification | **120**

Virtualized Data Center Using Logical Systems | 139

Two-Tiered Virtualized Data Center Solution for Large Enterprise Networks | **139**

Requirements of a Two-Tiered Virtualized Data Center for Large Enterprise Networks | **142**

Example: Configuring a Two-Tiered Virtualized Data Center for Large Enterprise Networks | **145**

Requirements | **145**

Configuring a Two-Tiered Virtualized Data Center Overview | **145**

Configuring the Access Layer | **148**

Configuring the Aggregation Layer in the Trusted Logical Systems | **154**

Configuring the Core Layer in the Untrusted Logical Systems | **164**

Configuring the Security Device | **171**

3 **Routing Protocols and Routing Policies on Logical Systems**

BGP on Logical Systems | 200

Example: Configuring Internal BGP Peering Sessions on Logical Systems | **200**

Requirements | **200**

Overview | **200**

Configuration | **201**

Verification | **210**

Example: Configuring External BGP on Logical Systems with IPv6 Interfaces | **214**

Requirements | **215**

Overview | **215**

Configuration | **218**

Verification | **230**

Example: Configuring BFD on Internal BGP Peer Sessions | **237**

Requirements | **237**

Overview | **237**

Configuration | **239**

Verification | **245**

Example: Configuring EBGP Multihop Sessions on Logical Systems | **250**

Requirements | **250**

Overview | **250**

Configuration | **251**

Verification | **262**

IS-IS on Logical Systems | 265

Example: Configuring IS-IS on Logical Systems Within the Same Router | **265**

Requirements | **266**

Overview | **266**

Configuration | **267**

Verification | **275**

Example: Configuring an IS-IS Default Route Policy on Logical Systems | **280**

Requirements | **281**

Overview | **281**

Configuration | **282**

Verification | **286**

OSPF on Logical Systems | 289

Example: Configuring OSPF on Logical Systems Within the Same Router | **290**

Requirements | **290**

Overview | **290**

Configuration | **291**

Verification | **297**

Understanding OSPF Routing Policy | **300**

Example: Configuring an OSPF Default Route Policy on Logical Systems | **303**

Requirements | **303**

Overview | **303**

Configuration | **304**

Verification | **307**

Example: Configuring a Conditional OSPF Default Route Policy on Logical Systems | **310**

Requirements | **310**

Overview | **310**

Configuration | **311**

Verification | **317**

Example: Configuring an OSPF Import Policy on Logical Systems | **321**

Requirements | **321**

Overview | **321**

Configuration | **323**

Verification | **329**

RSVP-Signaled Point-to-Multipoint LSPs on Logical Systems | 333

Point-to-Multipoint LSPs Overview | **333**

Example: Configuring an RSVP-Signaled Point-to-Multipoint LSP on Logical Systems | **335**

Requirements | **335**

Overview | **336**

Configuration | **337**

Verification | **363**

4 **Monitoring Logical Systems**

Monitoring Logical Systems for Routers and Switches | 368

Example: Running Operational Mode Commands on Logical Systems | **368**

Requirements | **368**

Overview | **368**

Configuration | **369**

Logical System Logging Overview | **371**

Example: Viewing BGP Trace Files on Logical Systems | **371**

Requirements | **372**

Overview | **372**

Configuration | 373

Verification | 378

5 Configuration Statements

logical-system-mux | 381

6 Operational Commands

Operational-Mode Commands | 385

CLI Operational Mode Command Overview | 385

clear cli logical-system | 388

set cli logical-system | 390

show cli authorization | 391

show interfaces detail | 397

About This Guide

Use this guide to configure logical systems on devices running Junos OS.

With Junos OS, you can partition a single router into multiple logical devices that perform independent routing tasks, known as logical systems. Logical systems are defined largely by the resources allocated to them, features enabled for the logical context, their routing configurations, and their logical interface assignments. 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.

CHAPTER 1

Overview

Understanding Logical Systems for Routers and Switches | 2

Understanding Logical Systems for Routers and Switches

IN THIS SECTION

- Comparing Junos OS Device Virtualization Technologies | 2
- Introduction to Logical Systems | 4
- Logical Systems Applications | 5
- Junos OS Features That Are Supported on Logical Systems | 6
- Logical Systems System Requirements | 8
- Logical Systems Operations and Restrictions | 9

Logical Systems enable you to partition a single router into multiple logical devices that perform independent routing tasks. It offers an effective way to maximize the use of a single routing or switching platform. For more information, see the following topics:

Comparing Junos OS Device Virtualization Technologies

The Junos OS supports multiple device virtualization technologies. The technologies have similar names, which can lead to confusion.

The Junos OS device virtualization technologies are:

- Logical systems—Offer routing and management separation. Management separation means multiple user access. Each logical system has its own routing tables.

  Logical routers is the old name for logical systems. Beginning with Junos OS Release 9.3, the logical router feature has been renamed logical system. All configuration statements, operational commands, `show` command output, error messages, log messages, and SNMP MIB objects that contain the string logical-router have been changed to logical-system.

- Virtual routers—Offer separate routing tables and scalable routing separation. Virtual routers are similar to VPN routing and forwarding instance types except that they are used for non-VPN-related applications. Virtual routers typically consist of the routing tables, the interfaces assigned to the routing tables, routing protocol configurations, and routing option configurations. There are no

virtual routing and forwarding (VRF) import, VRF export, VRF target, or route distinguisher requirements for the virtual router instance-type.

You can use virtual router routing instance types on a single device to segment your network, for example, as opposed to configuring multiple devices to achieve the same result.. Virtual router instances can isolate traffic by separating the device into multiple, independent virtual routers, each with its own routing table.

- VRF-Lite—Offers routing separation. The functionality of VRF-Lite is similar to virtual routers, but VRF-Lite is for smaller environments.
- Virtual switches—Offer scalable switching separation.
- Junos node slicing—Junos node slicing allows a single MX Series router to be partitioned to appear as multiple, independent routers. Each partition has its own Junos OS control plane, which runs as a virtual machine (VM), and a dedicated set of line cards. Each partition is called a guest network function (GNF). You can configure logical systems inside a GNF. For more information on Junos node slicing, see *Understanding Junos Node Slicing*.

Table 1 on page 3 summarizes the benefits of virtual routers, VRF-Lite, and logical systems.

Table 1: Benefits of Virtual Routers, VRF-Lite, and Logical Systems

Benefits	Virtual Router	VRF-Lite	Logical Systems
Logical platform partitioning	Yes	Yes	Yes
Fault isolation on the routing plane	No	No	Yes
Multiple user access (management separation)	No	No	Yes
Scalable routing separation	Yes	No	Yes

Introduction to Logical Systems

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 the Junos® operating system (Junos OS), 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.

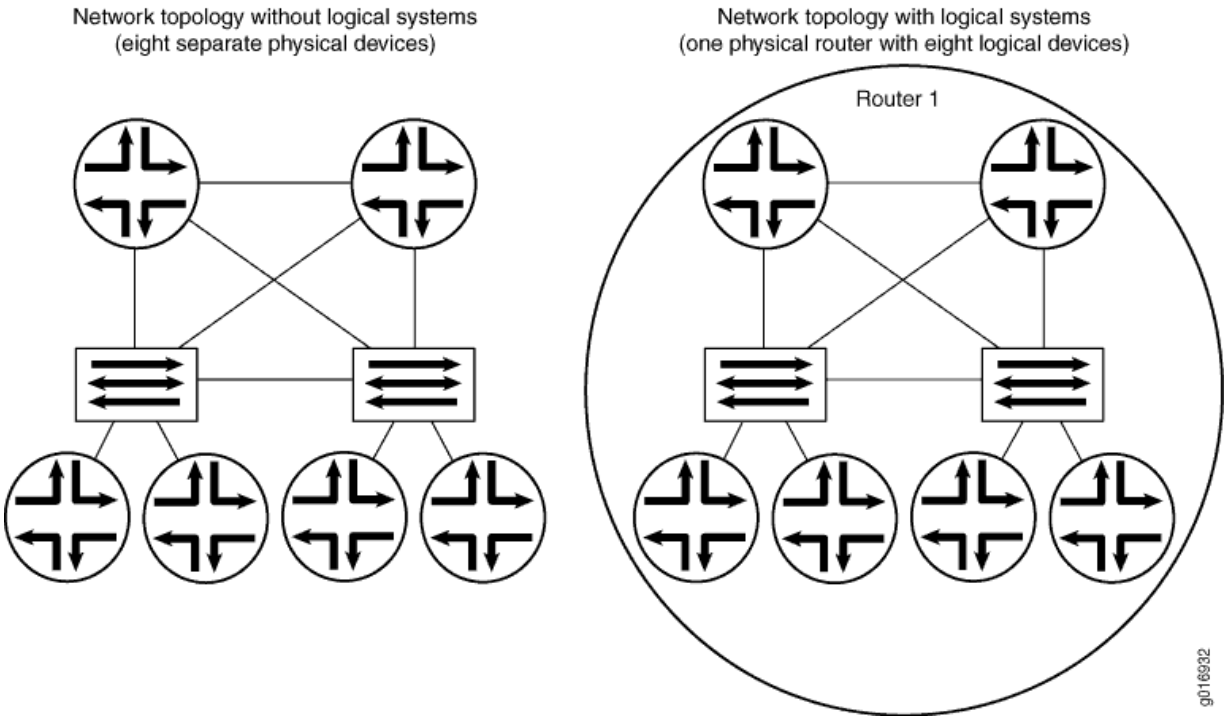
> **NOTE**: Beginning with Junos OS Release 9.3, the logical router feature has been renamed logical system.
>
> All configuration statements, operational commands, `show` command output, error messages, log messages, and SNMP MIB objects that contain the string logical-router have been changed to logical-system.

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 1 on page 5, 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 supports 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 1 on page 5, a set of logical systems within a single router can handle the functions previously performed by several small routers.

Figure 1: Logical Systems Concepts



Logical Systems Applications

Logical systems are discrete contexts that virtually divide a supported device into multiple devices, isolating one from another and protecting them from faulty conditions outside their own contexts.

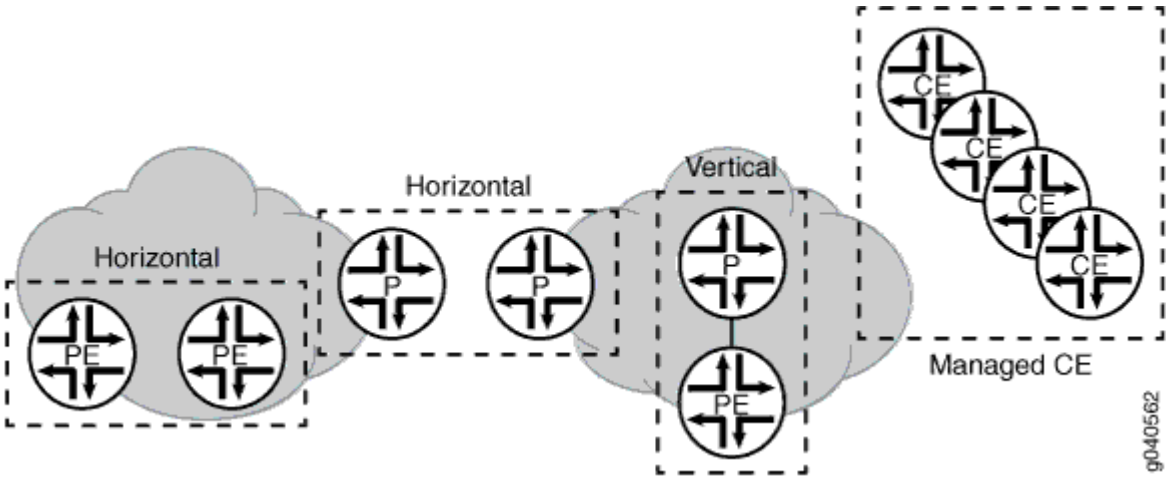
The logical systems functionality enables you to partition the device and assign private logical systems to groups or organizations. Logical systems are defined largely by the resources allocated to them, features enabled for the logical context, their routing configurations, and their *logical interface* assignments. Logical systems segment a physical routing device to be configured and operated as multiple independent routers within a platform. This isolates routing protocols and interfaces among up to 16 logical systems (including the primary logical system). User permissions and access are defined separately for each logical system, enabling different groups to manage the same physical device. Logical systems enable the use of large routing devices in small routing device roles and provide flexible segmentation of routing by service type. Multiple service capabilities bring improved asset optimization by consolidating services into one device.

For example, logical systems enable the following services on a single routing device platform:

- Internet BGP peering
- Core transit
- Edge aggregation and dedicated access
- MPLS provider edge (PE) and provider (P) VPN label-switched routing routers (LSRs)

Figure 2 on page 6 shows how logical systems can be used for horizontal consolidation, vertical consolidation, and managed services. Horizontal consolidation occurs when you combine routing device functions of the same layer into a single routing device. Vertical consolidation occurs when you collapse routing device functions of different layers into a single routing device. With managed services, each logical system is a customer routing device.

Figure 2: Applications of Logical Systems



Junos OS Features That Are Supported on Logical Systems

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).

- 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).
- Ethernet Virtual Private Network (EVPN) has been added for logical systems running on MX devices. Running EVPN in a logical system provides the same options and performance as running EVPN on a physical system, which adheres to the standards described in RFC 7432.
- Resource Reservation Protocol (RSVP) point-to-multipoint label-switched paths (LSPs).
- Multicast protocols, such as Protocol Independent Multicast (PIM), Distance Vector Multicast Routing Protocol (DVMRP), rendezvous point (RP), and source designated router (DR).
- All policy-related statements available at the `[edit policy-options]` hierarchy level.
- Most routing options statements available at the `[edit routing-options]` hierarchy level.
- *Graceful Routing Engine switchover* (GRES). Configure graceful Routing Engine switchover on the main router with the `graceful-switchover` statement at the `[edit chassis redundancy]` hierarchy level.
- Graceful restart. Include the `graceful-restart` statement at the `[edit logical-systems logical-system-name routing-options]` hierarchy level.
- You can assign most interface types to a logical system. For a list of unsupported PICs, see "Logical Systems Operations and Restrictions" on page 9.
- Starting in Junos OS Release 11.4, flow aggregation in logical systems is supported. In the logical system, sampling based on the Routing Engine is not supported. Only PIC-based sampling is supported. Logical systems support only cflowd version 9. Currently, cflowd version 5 and cflowd version 8 are not supported in logical systems. Flow aggregation in logical systems is slightly different from flow aggregation on the main router in that when you configure flow aggregation in logical systems, the `route-record` statement is not required.
- Flow aggregation is supported by the Multiservices DPC (MS-DPC). Jflow is not supported on logical systems for by MS-MPC and MS-MICs.
- *Port mirroring*, 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.
- The 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. The SNMP manager for a routing instance can request and manage SNMP data only for that routing instance and other routing instances in the same logical system. 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.

- Starting in Junos OS Release 11.4, support for system logging at the [`edit logical-system` *`logical-system-name`* `system syslog`] hierarchy level is introduced.
- Starting in Junos OS Release 13.3R1, the Non Stop Active Routing (NSR) feature is supported on logical systems to preserve interface and kernel information. The `nonstop-routing` option is introduced at the [`edit logical-systems logical-system-name routing-options`] hierarchy to enable nonstop active routing for logical systems.
- Starting in Junos OS Release 14.1, you can configure multichassis link aggregation (MC-LAG) interfaces on logical systems within a router. On MX Series routers, MC-LAG enables a device to form a logical LAG interface with two or more other devices. MC-LAG provides additional benefits over traditional LAG in terms of node level redundancy, multi- homing support, and loop-free Layer 2 network without running Spanning Tree Protocol (STP). The MC-LAG devices use Inter-Chassis Communication Protocol (ICCP) to exchange the control information between two MC-LAG network devices.
- Starting in Junos OS Release 14.2, an MX Series Virtual Chassis configuration supports the use of logical systems on MX Series routers with Modular Port Concentrators (MPCs). A *Virtual Chassis* enables a collection of member routers to function as a single virtual router, and extends the features available on a single router to the member routers in the Virtual Chassis.

Logical Systems System Requirements

To implement logical systems, your system must meet the minimum requirements listed here.

Software Requirements

- Junos OS Release 12.1x48 or later for support on PTX1000, PTX3000, PTX5000, PTX10008, and PTX10016 routers
- Junos OS Release 8.5 or later for logical system administrator support
- Junos OS Release 8.4 or later for SNMP enhancements and limits
- Junos OS Release 8.3 or later for Bidirectional Forwarding Detection (BFD) on logical systems
- Junos OS Release 8.2 or later for support on MX Series routers

- Junos OS Release 7.5 or later for SNMP support within a logical system
- Junos OS Release 7.4 or later for multicast protocol RP and source designated router functionality within a logical system
- Junos OS Release 7.0 or later to implement a logical tunnel (**lt**) interface on an integrated Adaptive Services Module in an M7i router
- Junos OS 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 OS Release 6.0 or later for basic logical system functionality

Hardware Requirements

- One or more M Series, MX Series, PTX Series, or T Series routers
- On M Series and T Series routers, a variety of PICs to assign interfaces to each logical system
- One or more EX9200 switches

Logical Systems Operations and Restrictions

Logical systems have the following operations and restrictions:

- You can configure a maximum of 15 logical systems plus the primary logical system on a routing device. When a configuration session is in use, users who are tied to the same logical system cannot commit configuration changes.
- The routing device has only one running configuration database, which contains configuration information for the main routing device and all associated logical systems. When configuring a logical system, a user has his own candidate configuration database, which does not become part of the running configuration database until the user issues the `commit` command.

  NOTE: The flow routes are not supported in a non-default logical systems.

- Configuring the out-of-band management interface, such as **em0** or **fxp0**, in a logical system is not supported.
- The following guidelines describe how firewall filters affect the main routing device, logical systems, and virtual routers. The "default loopback interface" refers to **lo0.0** (associated with the default routing table), the “loopback interface in a logical system” refers to **lo0.*n*** configured in the logical

system, and the “loopback interface in the virtual router” refers to **lo0.*n*** configured in the virtual router.

If you configure Filter A on the default loopback interface in the main routing device but do not configure a filter on the loopback interface in a logical system, the logical system does not use a filter.

If you configure Filter A on the default loopback interface in the main routing device but do not configure a loopback interface in a logical system, the logical system uses Filter A.

If you configure Filter A on the default loopback interface on the main routing device and Filter B on the loopback interface in a logical system, the logical system uses Filter B. In a special case of this rule, when you also configure a routing instance of type **virtual-router** on the logical system, the following rules apply:

- If you configure Filter C on the loopback interface in the virtual router, traffic belonging to the virtual router uses Filter C.
- If you do not configure a filter on the loopback interface in the virtual router, traffic belonging to the virtual router does not use a filter.
- If you do not configure a loopback interface in the virtual router, traffic belonging to the virtual router uses Filter A.

- If a logical system experiences an interruption of its routing protocol process (**rpd**), the core dump output is placed in **/var/tmp/** in a file called **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 PICs are not supported with logical systems: Adaptive Services, Multiservices, ES, Monitoring Services, and Monitoring Services II.
- Generalized MPLS (GMPLS), IP Security (IPsec), and sampling are not supported.
- Ethernet VPN (EVPN) support, including EVPN-MPLS, EVPN + VXLAN, and PBB EVPN, has been extended to logical systems running on MX devices. The same EVPN options and performance that are available in the default EVPN instance are available in a logical system. Note that Graceful Restart, Graceful Routing Engine switchover (GRES), and nonstop active routing (NSR) are not supported. Configure EVPN on a logical system under the `[edit logical-systems` *`logical-system-name`* `routing-instances` *`routing-instance-name`* `protocols evpn]` hierarchy.
- *Class of service* (CoS) on a logical tunnel (**lt**) or virtual loopback tunnel (**vt**) interface 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). However, you can use the `vrf-table-label` statement on multiple logical systems if the core-facing interface is located on MX Series routers with MPCs.
- The *primary 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.
- You can configure only Frame Relay interface encapsulation on a logical tunnel interface (**lt-**) when it is configured with an IPv6 address.
- IPv6 tunneling is not supported with point-to-multipoint label-switched paths (LSPs) configured on logical systems.
- IGMP snooping is not supported.
- BGP MVPNs and NG MVPNs are supported in logical systems. Draft-rosen multicast VPNs are not supported in a logical system environment even though the configuration statements can be configured under the logical-systems hierarchy.
- Inline services are not supported in logical systems.
- Carrier support Carrier(CsC) are not supported in logical systems.
- If you configure virtual private LAN service (VPLS) for a logical system, the `no-tunnel-services` statement is visible but not supported on DPC cards.
- In a VPLS multihoming scenario in which a logical tunnel interface (lt-) is used for connecting the dual-home VPLS, Junos OS creates a unique static MAC address for every logical tunnel interface configured. This MAC address is not flushed when a CCC down event occurs on the link and when traffic is switched from the primary link to the backup link (or the reverse). As a result, any traffic that is destined for hosts behind the logical tunnel MAC address does not take the new path.

Release History Table

Release	Description
17.4	Ethernet Virtual Private Network (EVPN) has been added for logical systems running on MX devices. Running EVPN in a logical system provides the same options and performance as running EVPN on a physical system, which adheres to the standards described in RFC 7432.

17.4	Ethernet VPN (EVPN) support, including EVPN-MPLS, EVPN + VXLAN, and PBB EVPN, has been extended to logical systems running on MX devices. The same EVPN options and performance that are available in the default EVPN instance are available in a logical system. Note that Graceful Restart, Graceful Routing Engine switchover (GRES), and nonstop active routing (NSR) are not supported. Configure EVPN on a logical system under the `[edit logical-systems` *`logical-system-name`* `routing-instances` *`routing-instance-name`* `protocols evpn]` hierarchy.
14.2	Starting in Junos OS Release 14.2, an MX Series Virtual Chassis configuration supports the use of logical systems on MX Series routers with Modular Port Concentrators (MPCs).
14.1	Starting in Junos OS Release 14.1, you can configure multichassis link aggregation (MC-LAG) interfaces on logical systems within a router.
13.3R1	Starting in Junos OS Release 13.3R1, the Non Stop Active Routing (NSR) feature is supported on logical systems to preserve interface and kernel information. The `nonstop-routing` option is introduced at the `[edit logical-systems logical-system-name routing-options]` hierarchy to enable nonstop active routing for logical systems.
11.4	Starting in Junos OS Release 11.4, flow aggregation in logical systems is supported.
11.4	Starting in Junos OS Release 11.4, support for system logging at the `[edit logical-system` *`logical-system-name`* `system syslog]` hierarchy level is introduced.
9.3	Beginning with Junos OS Release 9.3, the logical router feature has been renamed logical system.

RELATED DOCUMENTATION

Setting Up Logical Systems | 17

Monitoring Logical Systems for Routers and Switches | 368

2

CHAPTER

Configuring Logical Systems

User Access on Logical Systems | 14

Setting Up Logical Systems | 17

Static Routing on Logical Systems | 40

Firewall Filters on Logical Systems | 49

Layer 2 Learning and Forwarding on Logical Systems | 71

Multichassis Link Aggregation on Logical Systems Overview | 75

VPNs and VPLS | 80

Virtualized Data Center Using Logical Systems | 139

User Access on Logical Systems

IN THIS SECTION

- Example: Configuring Logical System Administrators | 14

Example: Configuring Logical System Administrators

IN THIS SECTION

- Requirements | 14
- Overview | 14
- Configuration | 15
- Verification | 17

This example shows how to configure logical system administrators.

Requirements

You must be the primary administrator to assign system administrators to logical systems.

Overview

IN THIS SECTION

- Diagram | 15

The primary administrator can assign one or more 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 logical system administrators cannot access any global configuration statements. This also

means that command output is restricted to the context to which the logical system administrators are assigned.

Configuring a user account for each logical system helps in navigating the CLI. This enables you to log in to each logical system and be positioned within the root of that logical system as if you were in the root of a physical router.

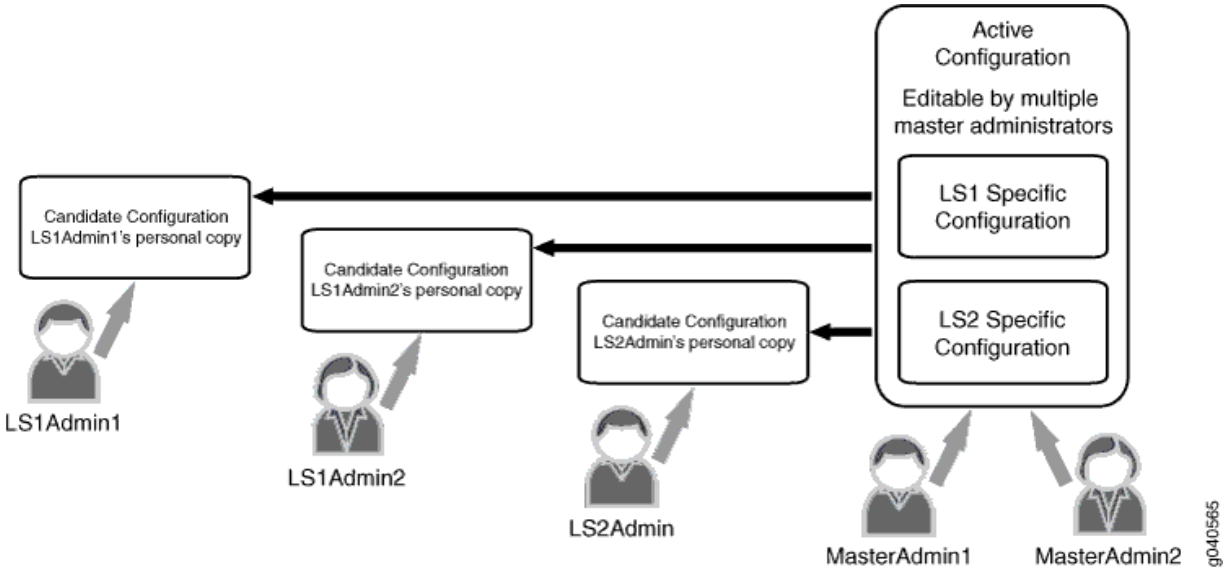
In this example, **LS1Admin** has full permissions on Logical System LS1.

In this example, **LS2Admin** has the ability to view Logical System LS2 but not to change the configuration.

Diagram

Figure 3 on page 15 shows how logical system administration works.

Figure 3: Logical System Administrators



Configuration

IN THIS SECTION

- CLI Quick Configuration | 16
- Procedure | 16

CLI Quick Configuration

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

```
set logical-systems LS1
set logical-systems LS2
set system login class admin1 logical-system LS1
set system login class admin2 logical-system LS2
set system login class admin1 permissions all
set system login class admin2 permissions view
set system login user LS1Admin class admin1
set system login user LS2Admin class admin2
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To assign logical system administrators to a logical systems:

1. Configure the logical systems.

```
[edit]
user@host# set logical-systems LS1
user@host# set logical-systems LS2
```

2. Create the login classes and assign logical systems to the classes.

```
[edit]
user@host# set system login class admin1 logical-system LS1
user@host# set system login class admin2 logical-system LS2
```

3. Assign permissions to the login classes.

```
[edit]
user@host# set system login class admin1 permissions all
user@host# set system login class admin2 permissions view
```

4. Assign users to the login classes.

```
[edit]
user@host# set system login user LS1Admin class admin1
user@host# set system login user LS2Admin class admin2
```

5. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

To verify that the configuration is working properly, issue the `show cli authorization` command to view permissions for the current user.

SEE ALSO

Understanding Junos OS Access Privilege Levels

Setting Up Logical Systems

IN THIS SECTION

- Using Junos OS to Configure Logical System Administrators | 18
- Example: Creating an Interface on a Logical System | 19
- Configuring Logical System Interface Properties | 22

- Example: Connecting a Logical System to a Physical Router | 23
- Example: Connecting Logical Systems Within the Same Device Using Logical Tunnel Interfaces on MX Series Routers and EX Series Switches | 27
- Example: Connecting Logical Systems Within the Same Router Using Logical Tunnel Interfaces | 33

Logical Systems enable you to create an interface and configure IP addresses. It shows how to add users to a user logical system. For more information, see the following topics:

Using Junos OS to Configure Logical System Administrators

Using Junos OS, you can partition a single router or switch into multiple logical devices that perform independent routing or switching tasks. When creating logical systems, you must configure logical system administrators and interfaces, assign logical interfaces to logical systems, and configure various other logical system statements.

The primary administrator can assign one or more logical system administrators to each logical system. Once assigned to a logical system, administrators are restricted to viewing only configurations of the logical system to which they are assigned and accessing only the operational commands that apply to that particular logical system. This restriction means that these administrators cannot access global configuration statements, and all command output is restricted to the logical system to which the 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 and apply the class to the user. For example:

```
[edit]
system {
    login {
        class admin1 {
            permissions all;
            logical-system logical-system-LS1;
        }
        class admin2 {
            permissions view; # Gives users assigned to class admin2 the ability to view
# but not to change the configuration.
            logical-system logical-system-LS2;
        }
```

```
        user user1 {
            class admin1;
        }
        user user2 {
            class admin2;
        }
    }
}
```

Fully implementing logical systems requires that you also configure any protocols, routing statements, switching statements, and policy statements for the logical system.

SEE ALSO

Defining Junos OS Login Classes

Example: Creating an Interface on a Logical System

IN THIS SECTION

- Requirements | 19
- Overview | 20
- Configuration | 21
- Verification | 22

This example shows how to create an interface on a logical system.

Requirements

For the interface on the logical system to have connectivity, the corresponding physical interface must be administratively up, and the physical link must be up. You can verify the status of the physical interface by running the `show interfaces terse` command.

Overview

In logical systems, 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 (primary 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 | 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);
            }
        }
    }
}
```

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 primary 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).

In this example, you create the **fe-1/1/3** physical interface on the main router. You can also add values for properties that you need to configure on the physical interface, such as physical encapsulation, VLAN tagging (enabling), and link speed.

The example then shows how to assign logical interfaces to a logical system. Once you do this, the logical interfaces are considered part of the logical system.

Any logical interface unit can only be assigned to one system, including the main router. For example, if you configure logical unit 3 in the main router, you cannot configure logical unit 3 in a logical system.

In this example, you create logical unit 0 on Logical System LS1. You can also add values for properties that you need to configure on the logical interface, such as logical interface encapsulation, VLAN ID number, and protocol family.

Configuration

IN THIS SECTION

- CLI Quick Configuration | 21
- Procedure | 21

CLI Quick Configuration

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

```
set interfaces fe-1/1/3 description "main router interface"
set logical-systems LS1 interfaces fe-1/1/3 unit 0 description "LS1 interface"
set logical-systems LS1 interfaces fe-1/1/3 unit 0 family inet address 10.11.2.2/24
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure an interface on a logical system:

1. As the primary administrator, configure the physical interface on the main router.

```
[edit]
user@host# set interfaces fe-1/1/3 description "main router interface"
```

2. Create the logical system interface on the logical unit.

```
[edit]
user@host# set logical-systems LS1 interfaces fe-1/1/3 unit 0 description "LS1 interface"
user@host# set logical-systems LS1 interfaces fe-1/1/3 unit 0 family inet address 10.11.2.2/24
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

To verify that the configuration is working properly, issue the `show interfaces` command.

SEE ALSO

- ping
- CLI Explorer
- show interfaces detail | **397**
- CLI Explorer

Configuring Logical System Interface Properties

With Junos OS, you can partition a single physical router into multiple logical devices that perform independent routing tasks. Because logical systems perform a subset of the tasks once handled by the physical router, logical systems offer an effective way to maximize the use of a single router.

1. Configure the physical interface that needs to be partitioned into multiple logical systems.

```
[edit]
user@host# set interfaces interface-name  description description
```

2. Create the logical system interface on the logical unit.

```
[edit]
user@host# set logical-systems name interfaces interface-name  unit logical-unit-number
description description
```

3. Configure the required properties for the logical system.

```
[edit logical-systems name]
user@host# set interfaces interface-name  unit logical-unit-number family name address address
```

SEE ALSO

Setting Up Logical Systems | 17

Example: Connecting a Logical System to a Physical Router

IN THIS SECTION

- Requirements | 24
- Overview | 24
- Configuration | 25
- Verification | 26

This example shows how to configure an interface on a logical system to connect to a separate router. The separate router can be a physical router or a logical system on a physical router.

Requirements

PICs must be installed on the two routers.

Overview

IN THIS SECTION

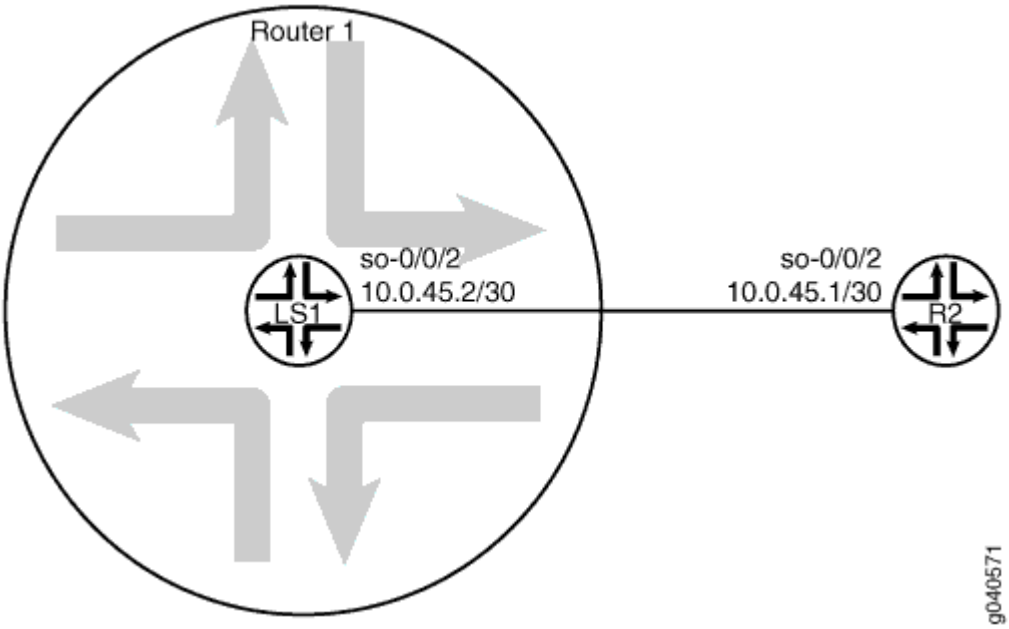
- Topology | 24

In this example, Logical System LS1 is configured on Router R1. The Logical System LS1 has a direct connection to Router R2.

Topology

Figure 4 on page 24 shows the topology used in this example.

Figure 4: Logical System Connected to a Physical Router



Configuration

IN THIS SECTION

- CLI Quick Configuration | 25
- Procedure | 25

CLI Quick Configuration

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

Router R1

```
set interfaces so-0/0/2 description "main router interface to R2"
set logical-systems LS1 interfaces so-0/0/2 unit 0 description LS1->R2
set logical-systems LS1 interfaces so-0/0/2 unit 0 family inet address 10.0.45.2/30
```

Device R2

```
set interfaces so-0/0/2 description R2->LS1
set interfaces so-0/0/2 unit 0 family inet address 10.0.45.1/30
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To connect a logical system to a physical router:

1. On Router R1, configure the interface.

```
[edit]
user@R1# set interfaces so-0/0/2 description "main router interface to R2"
```

2. On Router R1, configure the Logical System LS1 interface.

```
[edit]
user@R1# set logical-systems LS1 interfaces so-0/0/2 unit 0 description LS1->R2
user@R1# set logical-systems LS1 interfaces so-0/0/2 unit 0 family inet address 10.0.45.2/30
```

3. On Device R2, configure the interface to Logical System LS1.

```
[edit]
user@R2# set interfaces so-0/0/2 description R2->LS1
user@R2# set interfaces so-0/0/2 unit 0 family inet address 10.0.45.1/30
```

4. If you are done configuring the devices, commit the configurations.

```
[edit]
user@host# commit
```

Verification

IN THIS SECTION

- Verifying Connectivity | 26

Confirm that the configuration is working properly.

Verifying Connectivity

Purpose

Make sure that the devices can ping each other.

Action

```
user@R2> ping 10.0.45.2
PING 10.0.45.2 (10.0.45.2): 56 data bytes
64 bytes from 10.0.45.2: icmp_seq=0 ttl=64 time=3.910 ms
64 bytes from 10.0.45.2: icmp_seq=1 ttl=64 time=3.559 ms
64 bytes from 10.0.45.2: icmp_seq=2 ttl=64 time=3.503 ms
```

```
user@R1> set cli logical-system LS1
Logical system: LS1
```

```
user@R1:LS1> ping 10.0.45.1
PING 10.0.45.1 (10.0.45.1): 56 data bytes
64 bytes from 10.0.45.1: icmp_seq=0 ttl=64 time=1.217 ms
64 bytes from 10.0.45.1: icmp_seq=1 ttl=64 time=1.183 ms
64 bytes from 10.0.45.1: icmp_seq=2 ttl=64 time=1.121 ms
```

Example: Connecting Logical Systems Within the Same Device Using Logical Tunnel Interfaces on MX Series Routers and EX Series Switches

IN THIS SECTION

- Requirements | 28
- Overview | 28
- Configuration | 29
- Verification | 31

This example shows how to configure logical tunnel interfaces to connect two logical systems that are configured in a single router.

Requirements

On M Series and T Series routers, you can create a logical tunnel interface if you have a Tunnel Services PIC installed on an Enhanced FPC in your routing platform.

On M40e routers, you can create a logical tunnel interface if you have a Tunnel Services PIC. (An Enhanced FPC is not required.)

On an M7i router, logical tunnel interfaces can be created by using the integrated Adaptive Services Module.

On an MX Series router, the primary administrator can configure logical tunnel interfaces by including the `tunnel-services` statement at the `[edit chassis fpc` *`slot-number`* `pic` *`number`*`]` hierarchy level.

Overview

IN THIS SECTION

- Topology | **29**

To connect two logical systems, you configure a logical tunnel interface on both logical systems. Then you configure a peer relationship between the logical tunnel interfaces, thus creating a point-to-point connection. Logical tunnel interfaces behave like regular interfaces. You can configure them with Ethernet, Frame Relay, or another encapsulation type. You can also configure routing protocols across them. In effect, the logical tunnel (**lt**) interfaces connect two logical systems within the same router. The two logical systems do not share routing tables. This means that you can run dynamic routing protocols between different logical systems within the same router.

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, configure a corresponding protocol family, and set the logical interface unit number of the peering **lt** interface.

In this example, the logical tunnel interfaces are configured to behave as Ethernet interfaces with the `encapsulation ethernet` statement. The IS-IS Protocol is enabled on the logical tunnel interfaces with the `family iso` statement.

When configuring logical tunnel interfaces, note the following:

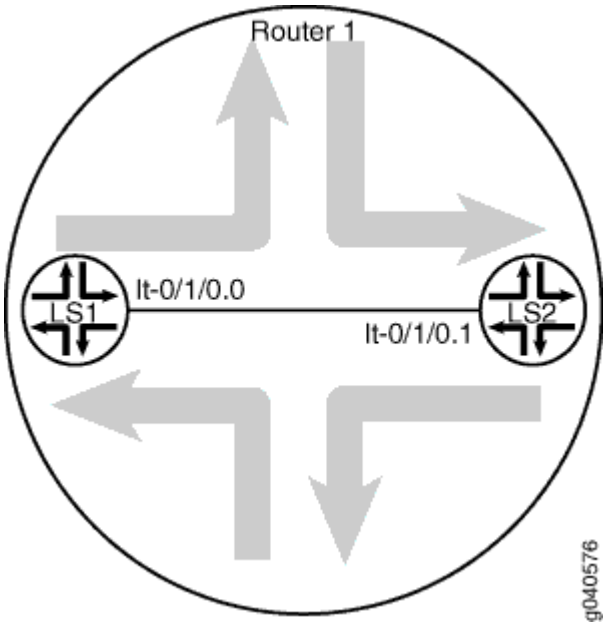
- The peering logical interfaces must have the same physical **lt** interface name. For example, a logical unit on **lt-0/1/0** cannot peer with a logical unit on **lt-0/0/10**. The FPC, PIC, and port numbers must match.

- The peering logical interfaces must be derived from the same PIC or module.
- You can configure only one peer unit for each logical interface. For example, unit 0 cannot peer with both unit 1 and unit 2.
- Logical tunnels are not supported with Adaptive Services, MultiServices, or Link Services PICs, but they are supported on the Adaptive Services Module on M7i routers.

Topology

Figure 5 on page 29 shows the topology used in this example.

Figure 5: Connecting Two Logical Systems



Configuration

IN THIS SECTION

- CLI Quick Configuration | 30
- Procedure | 30

CLI Quick Configuration

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

```
set logical-systems LS1 interfaces lt-0/1/0 unit 0 description LS1->LS2
set logical-systems LS1 interfaces lt-0/1/0 unit 0 encapsulation ethernet
set logical-systems LS1 interfaces lt-0/1/0 unit 0 peer-unit 1
set logical-systems LS1 interfaces lt-0/1/0 unit 0 family inet address 10.0.8.13/30
set logical-systems LS1 interfaces lt-0/1/0 unit 0 family iso
set logical-systems LS2 interfaces lt-0/1/0 unit 1 description LS2->LS1
set logical-systems LS2 interfaces lt-0/1/0 unit 1 encapsulation ethernet
set logical-systems LS2 interfaces lt-0/1/0 unit 1 peer-unit 0
set logical-systems LS2 interfaces lt-0/1/0 unit 1 family inet address 10.0.8.14/30
set logical-systems LS2 interfaces lt-0/1/0 unit 1 family iso
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To connect logical system interfaces:

1. Run the `show interfaces terse` command to verify that the router has a logical tunnel (**lt**) interface.

```
user@host> show interfaces terse
Interface               Admin Link Proto    Local                 Remote
so-0/0/0                up    down
so-0/0/1                up    down
so-0/0/2                up    down
so-0/0/3                up    down
gr-0/1/0                up    up
ip-0/1/0                up    up
lt-0/1/0                up    up
...
```

2. Configure the logical tunnel interface on Logical System LS1.

```
[edit]
user@host# set logical-systems LS1 interfaces lt-0/1/0 unit 0 description LS1->LS2
user@host# set logical-systems LS1 interfaces lt-0/1/0 unit 0 encapsulation ethernet
user@host# set logical-systems LS1 interfaces lt-0/1/0 unit 0 peer-unit 1
user@host# set logical-systems LS1 interfaces lt-0/1/0 unit 0 family inet address
10.0.8.13/30
user@host# set logical-systems LS1 interfaces lt-0/1/0 unit 0 family iso
```

3. Configure the logical tunnel interface on Logical System LS2.

```
[edit]
user@host# set logical-systems LS2 interfaces lt-0/1/0 unit 1 description LS2->LS1
user@host# set logical-systems LS2 interfaces lt-0/1/0 unit 1 encapsulation ethernet
user@host# set logical-systems LS2 interfaces lt-0/1/0 unit 1 peer-unit 0
user@host# set logical-systems LS2 interfaces lt-0/1/0 unit 1 family inet address
10.0.8.14/30
user@host# set logical-systems LS2 interfaces lt-0/1/0 unit 1 family iso
```

4. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

IN THIS SECTION

- Verifying That the Logical Systems Are Up | 32
- Verifying Connectivity Between the Logical Systems | 32

Confirm that the configuration is working properly.

Verifying That the Logical Systems Are Up

Purpose

Make sure that the interfaces are properly configured.

Action

```
user@host> show interfaces terse
Interface               Admin Link Proto    Local                 Remote
so-0/0/0                up    down
so-0/0/1                up    down
so-0/0/2                up    down
so-0/0/3                up    down
gr-0/1/0                up    up
ip-0/1/0                up    up
lt-0/1/0                up    up
lt-0/1/0.0              up    up   inet     10.0.8.13/30
                                   iso
lt-0/1/0.1              up    up   inet     10.0.8.14/30
                                   iso
...
```

Verifying Connectivity Between the Logical Systems

Purpose

Make sure that the network address appears as directly connected.

Action

```
user@host> show route logical-system all
logical-system: LS1

inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.8.12/30       *[Direct/0] 00:00:34
                    > via lt-0/1/0.0
10.0.8.13/32       *[Local/0] 00:00:34
```

```
                      Local via lt-0/1/0.0
-----

logical-system: LS2

inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.8.12/30       *[Direct/0] 00:00:34
                    > via lt-0/1/0.1
10.0.8.14/32       *[Local/0] 00:00:34
                      Local via lt-0/1/0.1
...
```

Example: Connecting Logical Systems Within the Same Router Using Logical Tunnel Interfaces

IN THIS SECTION

- Requirements | 33
- Overview | 34
- Configuration | 35
- Verification | 38

This example shows how to configure logical tunnel interfaces to connect two logical systems that are configured in a single MX Series 5G Universal Routing Platform.

Requirements

The MX Series router chassis must have a DPC, MPC, or MIC installed and in the online state.

Overview

IN THIS SECTION

- Topology | 35

To connect two logical systems, you configure a logical tunnel interface on both logical systems. Then you configure a peer relationship between the logical tunnel interfaces, thus creating a point-to-point connection. Logical tunnel interfaces behave like regular interfaces. You can configure them with Ethernet, Frame Relay, or another encapsulation type. You can also configure routing protocols across them. In effect, the logical tunnel (**lt**) interfaces connect two logical systems within the same router. The two logical systems do not share routing tables. This means that you can run dynamic routing protocols between different logical systems within the same router.

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, configure a corresponding protocol family, and set the logical interface unit number of the peering **lt** interface.

In this example, the logical tunnel interfaces are configured to behave as Ethernet interfaces with the `encapsulation ethernet` statement. The IS-IS Protocol is enabled on the logical tunnel interfaces with the `family iso` statement.

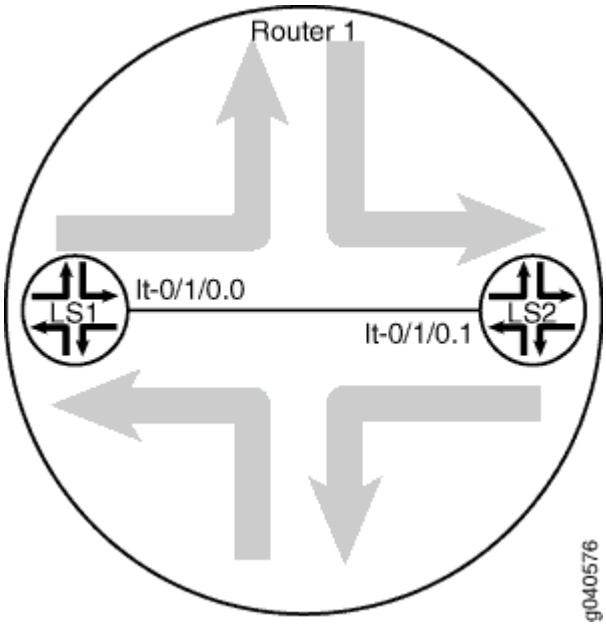
When configuring logical tunnel interfaces, note the following:

- The peering logical interfaces must have the same **lt** interface name. For example, a logical unit on **lt-0/1/0** cannot peer with a logical unit on **lt-0/0/10**. The FPC (DPC, MPC, or MIC), PIC, and port numbers must match.
- The peering logical interfaces must be derived from the same module.
- You can configure only one peer unit for each logical interface. For example, unit 0 cannot peer with both unit 1 and unit 2.

Topology

Figure 6 on page 35 shows the topology used in this example.

Figure 6: Connecting Two Logical Systems



Configuration

IN THIS SECTION

- CLI Quick Configuration | 35
- Procedure | 36

CLI Quick Configuration

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

```
set chassis fpc 0 pic 1 tunnel-services bandwidth 1g
set logical-systems LS1 interfaces lt-0/1/0 unit 0 description LS1->LS2
```

```
set logical-systems LS1 interfaces lt-0/1/0 unit 0 encapsulation ethernet
set logical-systems LS1 interfaces lt-0/1/0 unit 0 peer-unit 1
set logical-systems LS1 interfaces lt-0/1/0 unit 0 family inet address 10.0.8.13/30
set logical-systems LS1 interfaces lt-0/1/0 unit 0 family iso
set logical-systems LS2 interfaces lt-0/1/0 unit 1 description LS2->LS1
set logical-systems LS2 interfaces lt-0/1/0 unit 1 encapsulation ethernet
set logical-systems LS2 interfaces lt-0/1/0 unit 1 peer-unit 0
set logical-systems LS2 interfaces lt-0/1/0 unit 1 family inet address 10.0.8.14/30
set logical-systems LS2 interfaces lt-0/1/0 unit 1 family iso
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To connect logical system interfaces:

1. Run the `show chassis fpc` command to verify that the router has a DPC, MPC, or MIC installed and is in the online state.

```
user@host> show chassis fpc
                    Temp  CPU Utilization (%)   Memory    Utilization (%)
Slot State           (C)  Total  Interrupt      DRAM (MB) Heap     Buffer
  0  Online           31      4          0       1024       14         21
  1  Empty
  2  Empty
```

   This output shows that slot 1 and slot 2 are empty. Slot 0 is online.

2. Configure FPC slot 0 to support logical tunnel (**lt**) interfaces.

```
[edit]
user@host# set chassis fpc 0 pic 1 tunnel-services bandwidth 1g
```

   This command creates several tunnel interface types, including **gr**, **ip**, and **lt**. For this example, the important one is the logical tunnel (**lt**) interface.

3. Commit the configuration.

```
[edit]
user@host# commit
```

4. Run the `show interfaces terse` command to verify that the router has a logical tunnel (**lt**) interface.

```
user@host> show interfaces terse
Interface               Admin Link Proto    Local                 Remote
...
gr-0/1/0                up    up
ip-0/1/0                up    up
lt-0/1/0                up    up
...
```

5. Configure the logical tunnel interface on Logical System LS1.

```
[edit]
user@host# set logical-systems LS1 interfaces lt-0/1/0 unit 0 description LS1->LS2
user@host# set logical-systems LS1 interfaces lt-0/1/0 unit 0 encapsulation ethernet
user@host# set logical-systems LS1 interfaces lt-0/1/0 unit 0 peer-unit 1
user@host# set logical-systems LS1 interfaces lt-0/1/0 unit 0 family inet address
10.0.8.13/30
user@host# set logical-systems LS1 interfaces lt-0/1/0 unit 0 family iso
```

6. Configure the logical tunnel interface on Logical System LS2.

```
[edit]
user@host# set logical-systems LS2 interfaces lt-0/1/0 unit 1 description LS2->LS1
user@host# set logical-systems LS2 interfaces lt-0/1/0 unit 1 encapsulation ethernet
user@host# set logical-systems LS2 interfaces lt-0/1/0 unit 1 peer-unit 0
user@host# set logical-systems LS2 interfaces lt-0/1/0 unit 1 family inet address
10.0.8.14/30
user@host# set logical-systems LS2 interfaces lt-0/1/0 unit 1 family iso
```

7. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

IN THIS SECTION

- Verifying That the Logical Systems Are Up | 38
- Verifying Connectivity Between the Logical Systems | 39

Confirm that the configuration is working properly.

Verifying That the Logical Systems Are Up

Purpose

Make sure that the interfaces are properly configured.

Action

```
user@host> show interfaces terse
Interface               Admin Link Proto    Local                 Remote
so-0/0/0                up    down
so-0/0/1                up    down
so-0/0/2                up    down
so-0/0/3                up    down
gr-0/1/0                up    up
ip-0/1/0                up    up
lt-0/1/0                up    up
lt-0/1/0.0              up    up   inet     10.0.8.13/30
                                   iso
lt-0/1/0.1              up    up   inet     10.0.8.14/30
                                   iso
...
```

Verifying Connectivity Between the Logical Systems

Purpose

Make sure that the network address appears as directly connected by running the command below. You can also ping the IP addresses to confirm the connectivity.

Action

```
user@host> show route logical-system all
logical-system: LS1

inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.8.12/30       *[Direct/0] 00:00:34
                    > via lt-0/1/0.0
10.0.8.13/32       *[Local/0] 00:00:34
                      Local via lt-0/1/0.0
-----

logical-system: LS2

inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.8.12/30       *[Direct/0] 00:00:34
                    > via lt-0/1/0.1
10.0.8.14/32       *[Local/0] 00:00:34
                      Local via lt-0/1/0.1
...
```

SEE ALSO

Tunnel Interface Configuration on MX Series Routers Overview

RELATED DOCUMENTATION

Monitoring Logical Systems for Routers and Switches | 368

Static Routing on Logical Systems

IN THIS SECTION

- Example: Configuring Static Routes Between Logical Systems Within the Same Router | 40

Example: Configuring Static Routes Between Logical Systems Within the Same Router

IN THIS SECTION

- Requirements | 40
- Overview | 40
- Configuration | 42
- Verification | 46

This example shows how to configure static routes between logical systems. The logical systems are configured in a single physical router and are connected by logical tunnel interfaces.

Requirements

You must connect the logical systems by using logical tunnel (lt) interfaces. See "Example: Connecting Logical Systems Within the Same Device Using Logical Tunnel Interfaces on MX Series Routers and EX Series Switches" on page 27.

Overview

IN THIS SECTION

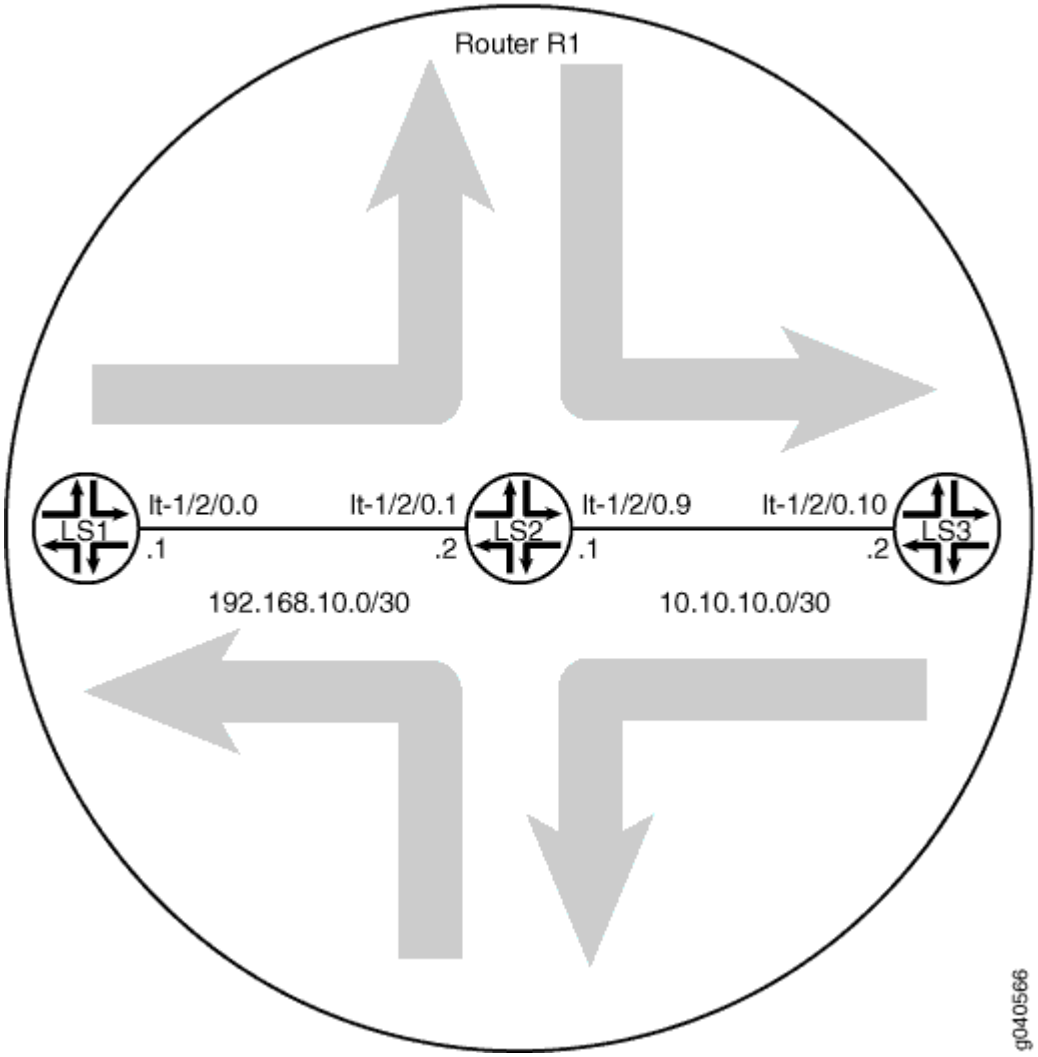
- Topology | 41

A static route is a hard-coded path in the device that specifies how the route gets to a certain subnet by using a certain path. Routers that are connected to stub networks are often configured to use static routes. A *stub network* is a network with no knowledge of other networks. Stub networks send non-local traffic by way of a single path, with the network aware only of a default route to non-local destinations. In this example, you configure Logical System LS1 with a static route to the 10.10.10.0/30 network and define the next-hop address as 192.168.10.2. You also configure Logical System LS1 with a static route to the 192.168.10.0/30 network and define a next-hop address of 10.10.10.1.

Topology

Figure 7 on page 41 shows the sample network.

Figure 7: Static Routes Between Logical Systems



Configuration

IN THIS SECTION

- CLI Quick Configuration | 42
- Procedure | 43
- Results | 45

CLI Quick Configuration

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

```
set logical-systems LS1 interfaces lt-1/2/0 unit 0 description LS1->LS2
set logical-systems LS1 interfaces lt-1/2/0 unit 0 encapsulation ethernet
set logical-systems LS1 interfaces lt-1/2/0 unit 0 peer-unit 1
set logical-systems LS1 interfaces lt-1/2/0 unit 0 family inet address 192.168.10.1/30
set logical-systems LS2 interfaces lt-1/2/0 unit 1 description LS2->LS1
set logical-systems LS2 interfaces lt-1/2/0 unit 1 encapsulation ethernet
set logical-systems LS2 interfaces lt-1/2/0 unit 1 peer-unit 0
set logical-systems LS2 interfaces lt-1/2/0 unit 1 family inet address 192.168.10.2/30
set logical-systems LS2 interfaces lt-1/2/0 unit 9 description LS2->LS3
set logical-systems LS2 interfaces lt-1/2/0 unit 9 encapsulation ethernet
set logical-systems LS2 interfaces lt-1/2/0 unit 9 peer-unit 10
set logical-systems LS2 interfaces lt-1/2/0 unit 9 family inet address 10.10.10.1/30
set logical-systems LS3 interfaces lt-1/2/0 unit 10 description LS3->LS2
set logical-systems LS3 interfaces lt-1/2/0 unit 10 encapsulation ethernet
set logical-systems LS3 interfaces lt-1/2/0 unit 10 peer-unit 9
set logical-systems LS3 interfaces lt-1/2/0 unit 10 family inet address 10.10.10.2/30
set logical-systems LS1 routing-options static route 10.10.10.0/30 next-hop 192.168.10.2
set logical-systems LS3 routing-options static route 192.168.10.0/30 next-hop 10.10.10.1
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure static routes between logical systems:

1. Run the `show interfaces terse` command to verify that the router has a logical tunnel (**lt**) interface.

```
user@host> show interfaces terse
Interface               Admin Link Proto    Local                 Remote
so-0/0/0                up    down
so-0/0/1                up    down
so-0/0/2                up    down
so-0/0/3                up    down
gr-1/2/0                up    up
ip-1/2/0                up    up
lt-1/2/0                up    up
...
```

2. Configure the logical tunnel interface on Logical System LS1 connecting to Logical System LS2.

```
[edit]
user@host# set logical-systems LS1 interfaces lt-1/2/0 unit 0 description LS1->LS2
user@host# set logical-systems LS1 interfaces lt-1/2/0 unit 0 encapsulation ethernet
user@host# set logical-systems LS1 interfaces lt-1/2/0 unit 0 peer-unit 1
user@host# set logical-systems LS1 interfaces lt-1/2/0 unit 0 family inet address
192.168.10.1/30
```

3. Configure the logical tunnel interface on Logical System LS2 connecting to Logical System LS1.

```
[edit]
user@host# set logical-systems LS2 interfaces lt-1/2/0 unit 1 description LS2->LS1
user@host# set logical-systems LS2 interfaces lt-1/2/0 unit 1 encapsulation ethernet
user@host# set logical-systems LS2 interfaces lt-1/2/0 unit 1 peer-unit 0
user@host# set logical-systems LS2 interfaces lt-1/2/0 unit 1 family inet address
192.168.10.2/30
```

4. Configure the logical tunnel interface on Logical System LS2 connecting to Logical System LS3.

```
[edit]
user@host# set logical-systems LS2 interfaces lt-1/2/0 unit 9 description LS2->LS3
user@host# set logical-systems LS2 interfaces lt-1/2/0 unit 9 encapsulation ethernet
user@host# set logical-systems LS2 interfaces lt-1/2/0 unit 9 peer-unit 10
user@host# set logical-systems LS2 interfaces lt-1/2/0 unit 9 family inet address
10.10.10.1/30
```

5. Configure the logical tunnel interface on Logical System LS3 connecting to Logical System LS2.

```
[edit]
user@host# set logical-systems LS3 interfaces lt-1/2/0 unit 10 description LS3->LS2
user@host# set logical-systems LS3 interfaces lt-1/2/0 unit 10 encapsulation ethernet
user@host# set logical-systems LS3 interfaces lt-1/2/0 unit 10 peer-unit 9
user@host# set logical-systems LS3 interfaces lt-1/2/0 unit 10 family inet address
10.10.10.2/30
```

6. Configure the static route on Logical System LS1 connecting to the 10.10.10.0/30 network.

```
[edit]
user@host# set logical-systems LS1 routing-options static route 10.10.10.0/30 next-hop
192.168.10.2
```

7. Configure the static route on Logical System LS3 connecting to the 192.168.10.0/30 network.

```
[edit]
user@host# set logical-systems LS3 routing-options static route 192.168.10.0/30 next-hop
10.10.10.1
```

8. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Results

Confirm your configuration by issuing the `show logical-systems` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show logical-systems
LS1 {
    interfaces {
        lt-1/2/0 {
            unit 0 {
                description LS1->LS2;
                encapsulation ethernet;
                peer-unit 1;
                family inet {
                    address 192.168.10.1/30;
                }
            }
        }
    }
    routing-options {
        static {
            route 10.10.10.0/30 next-hop 192.168.10.2;
        }
    }
}
LS2 {
    interfaces {
        lt-1/2/0 {
            unit 1 {
                description LS2->LS1;
                encapsulation ethernet;
                peer-unit 0;
                family inet {
                    address 192.168.10.2/30;
                }
            }
            unit 9 {
                description LS2->LS3;
                encapsulation ethernet;
                peer-unit 10;
                family inet {
                    address 10.10.10.1/30;
```

```
                }
            }
        }
    }
}
LS3 {
    interfaces {
        lt-1/2/0 {
            unit 10 {
                description LS3->LS2;
                encapsulation ethernet;
                peer-unit 9;
                family inet {
                    address 10.10.10.2/30;
                }
            }
        }
    }
    routing-options {
        static {
            route 192.168.10.0/30 next-hop 10.10.10.1;
        }
    }
}
```

Verification

IN THIS SECTION

- Verifying That the Logical Systems Are Up | 46
- Verifying Connectivity Between the Logical Systems | 47

Confirm that the configuration is working properly.

Verifying That the Logical Systems Are Up

Purpose

Make sure that the interfaces are properly configured.

Action

```
user@host> show interfaces terse
Interface               Admin Link Proto    Local                 Remote
...
lt-1/2/0                up    up
lt-1/2/0.0              up    up   inet     192.168.10.1/30
lt-1/2/0.1              up    up   inet     192.168.10.2/30
lt-1/2/0.9              up    up   inet     10.10.10.1/30
lt-1/2/0.10             up    up   inet     10.10.10.2/30
...
```

Verifying Connectivity Between the Logical Systems

Purpose

Make sure that the static routes appear in the routing tables of Logical Systems LS1 and LS3. Also, make sure that the logical systems can ping each other.

Action

```
user@host> show route logical-system LS1
inet.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.10.0/30      *[Static/5] 18:43:25
                    > to 192.168.10.2 via lt-1/2/0.0
192.168.10.0/30    *[Direct/0] 18:43:25
                    > via lt-1/2/0.0
192.168.10.1/32    *[Local/0] 18:43:25
                      Local via lt-1/2/0.0
```

```
user@host> show route logical-system LS3
inet.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.10.0/30      *[Direct/0] 23:11:21
                    > via lt-1/2/0.10
10.10.10.2/32      *[Local/0] 23:11:21
```

```
                      Local via lt-1/2/0.10
192.168.10.0/30    *[Static/5] 00:23:31
                    > to 10.10.10.1 via lt-1/2/0.10
```

From LS1, Ping LS3

```
user@host> set cli logical-system LS1
```

```
user@host:LS1> ping 10.10.10.2
PING 10.10.10.2 (10.10.10.2): 56 data bytes
64 bytes from 10.10.10.2: icmp_seq=0 ttl=63 time=1.263 ms
64 bytes from 10.10.10.2: icmp_seq=1 ttl=63 time=1.086 ms
64 bytes from 10.10.10.2: icmp_seq=2 ttl=63 time=1.077 ms
```

From LS3, Ping LS1

```
user@host> set cli logical-system LS3
```

```
user@host:LS3> ping 192.168.10.1
PING 192.168.10.1 (192.168.10.1): 56 data bytes
64 bytes from 192.168.10.1: icmp_seq=0 ttl=63 time=10.781 ms
64 bytes from 192.168.10.1: icmp_seq=1 ttl=63 time=1.167 ms
64 bytes from 192.168.10.1: icmp_seq=2 ttl=63 time=1.152 ms
```

SEE ALSO

Example: Creating an Interface on a Logical System | 19

Example: Connecting Logical Systems Within the Same Device Using Logical Tunnel Interfaces on MX Series Routers and EX Series Switches | 27

Firewall Filters on Logical Systems

IN THIS SECTION

- Understanding How to Use Standard Firewall Filters | 49
- Example: Configuring a Stateless Firewall Filter to Protect a Logical System Against ICMP Floods | 51
- Example: Configuring Filter-Based Forwarding on Logical Systems | 57

Firewall filters provide rules that define to accept or discard packets that are transiting an interface. For more information, see the following topics:

Understanding How to Use Standard Firewall Filters

IN THIS SECTION

- Using Standard Firewall Filters to Affect Local Packets | 49
- Using Standard Firewall Filters to Affect Data Packets | 50

Using Standard Firewall Filters to Affect Local Packets

On a router, you can configure one physical loopback interface, **lo0**, and one or more addresses on the interface. The loopback interface is the interface to the Routing Engine, which runs and monitors all the control protocols. The loopback interface carries local packets only. Standard firewall filters applied to the loopback interface affect the local packets destined for or transmitted from the Routing Engine.

NOTE: When you create an additional loopback interface, it is important to apply a filter to it so the Routing Engine is protected. We recommend that when you apply a filter to the loopback interface, you include the **apply-groups** statement. Doing so ensures that the filter is automatically inherited on every loopback interface, including **lo0** and other loopback interfaces.

Trusted Sources

The typical use of a standard stateless *firewall filter* is to protect the Routing Engine processes and resources from malicious or untrusted packets. To protect the processes and resources owned by the Routing Engine, you can use a standard stateless firewall filter that specifies which protocols and services, or applications, are allowed to reach the Routing Engine. Applying this type of filter to the loopback interface ensures that the local packets are from a trusted source and protects the processes running on the Routing Engine from an external attack.

Flood Prevention

You can create standard stateless firewall filters that limit certain TCP and ICMP traffic destined for the Routing Engine. A router without this kind of protection is vulnerable to TCP and ICMP flood attacks, which are also called denial-of-service (DoS) attacks. For example:

- A TCP flood attack of SYN packets initiating connection requests can overwhelm the device until it can no longer process legitimate connection requests, resulting in denial of service.
- An ICMP flood can overload the device with so many echo requests (ping requests) that it expends all its resources responding and can no longer process valid network traffic, also resulting in denial of service.

Applying the appropriate firewall filters to the Routing Engine protects against these types of attacks.

Using Standard Firewall Filters to Affect Data Packets

Standard firewall filters that you apply to your router's transit interfaces evaluate only the user data packets that transit the router from one interface directly to another as they are being forwarded from a source to a destination. To protect the network as a whole from unauthorized access and other threats at specific interfaces, you can apply firewall filters router transit interfaces .

SEE ALSO

- How Standard Firewall Filters Evaluate Packets
- Guidelines for Configuring Firewall Filters
- Guidelines for Applying Standard Firewall Filters

Example: Configuring a Stateless Firewall Filter to Protect a Logical System Against ICMP Floods

IN THIS SECTION

- Requirements | 51
- Overview | 51
- Configuration | 52
- Verification | 55

This example shows how to configure a stateless firewall filter that protects against ICMP denial-of-service attacks on a logical system.

Requirements

In this example, no special configuration beyond device initialization is required.

Overview

IN THIS SECTION

- Topology | 52

This example shows a stateless firewall filter called protect-RE that polices ICMP packets. The `icmp-policer` limits the traffic rate of the ICMP packets to 1,000,000 bps and the burst size to 15,000 bytes. Packets that exceed the traffic rate are discarded.

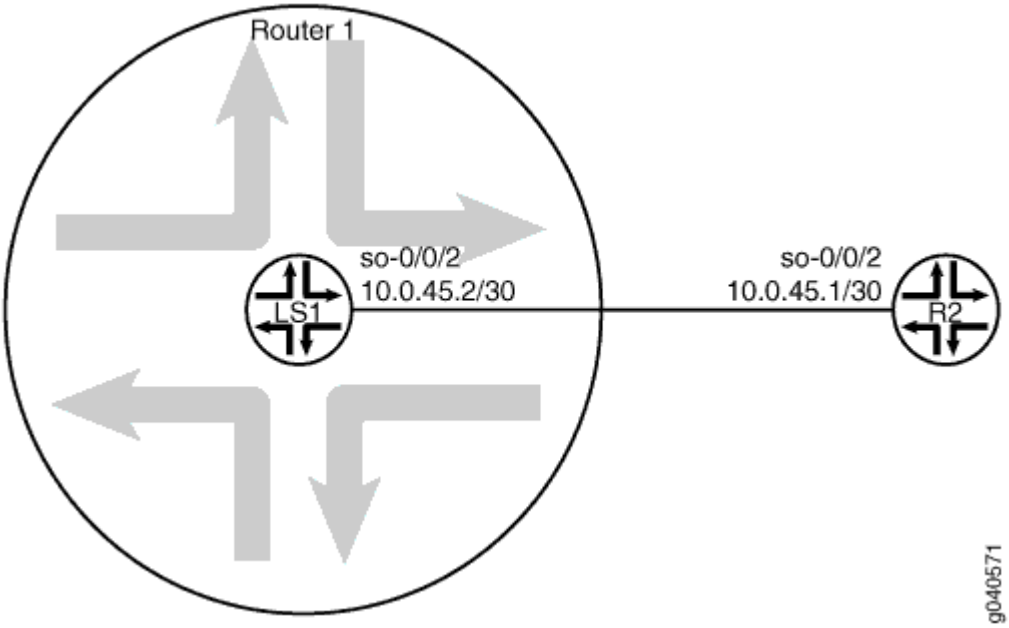
The policer is incorporated into the action of a filter term called `icmp-term`.

In this example, a ping is sent from a directly connected physical router to the interface configured on the logical system. The logical system accepts the ICMP packets if they are received at a rate of up to 1 Mbps (bandwidth-limit). The logical system drops all ICMP packets when this rate is exceeded. The `burst-size-limit` statement accepts traffic bursts up to 15 Kbps. If bursts exceed this limit, all packets are dropped. When the flow rate subsides, ICMP packets are again accepted.

Topology

Figure 8 on page 52 shows the topology used in this example.

Figure 8: Logical System with a Stateless Firewall



Configuration

IN THIS SECTION

- CLI Quick Configuration | 53
- Procedure | 53
- Results | 54

CLI Quick Configuration

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

```
set logical-systems LS1 interfaces so-0/0/2 unit 0 family inet policer input icmp-policer
set logical-systems LS1 interfaces so-0/0/2 unit 0 family inet address 10.0.45.2/30
set logical-systems LS1 firewall family inet filter protect-RE term icmp-term from protocol icmp
set logical-systems LS1 firewall family inet filter protect-RE term icmp-term then policer icmp-
policer
set logical-systems LS1 firewall family inet filter protect-RE term icmp-term then accept
set logical-systems LS1 firewall policer icmp-policer if-exceeding bandwidth-limit 1m
set logical-systems LS1 firewall policer icmp-policer if-exceeding burst-size-limit 15k
set logical-systems LS1 firewall policer icmp-policer then discard
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see No Link Title in the CLI User Guide.

To configure an ICMP firewall filter on a logical system:

1. Configure the interface on the logical system.

```
[edit]
user@host# set logical-systems LS1 interfaces so-0/0/2 unit 0 family inet address
10.0.45.2/30
```

2. Explicitly enable ICMP packets to be received on the interface.

```
[edit]
user@host# set logical-systems LS1 firewall family inet filter protect-RE term icmp-term from
protocol icmp
user@host# set logical-systems LS1 firewall family inet filter protect-RE term icmp-term then
accept
```

3. Create the policer.

```
[edit]
user@host# set logical-systems LS1 firewall policer icmp-policer if-exceeding bandwidth-limit
1m
user@host# set logical-systems LS1 firewall policer icmp-policer if-exceeding burst-size-
limit 15k
user@host# set logical-systems LS1 firewall policer icmp-policer then discard
```

4. Apply the policer to a filter term.

```
[edit]
user@host# set logical-systems LS1 firewall family inet filter protect-RE term icmp-term then
policer icmp-policer
```

5. Apply the policer to the logical system interface.

```
[edit]
user@host# set logical-systems LS1 interfaces so-0/0/2 unit 0 family inet policer input icmp-
policer
```

6. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Results

Confirm your configuration by issuing the `show logical-systems LS1` command.

```
user@host# show logical-systems LS1
interfaces {
    so-0/0/2 {
        unit 0 {
            family inet {
                policer {
                    input icmp-policer;
                }
```

```
                address 10.0.45.2/30;
            }
        }
    }
}
firewall {
    family inet {
        filter protect-RE {
            term icmp-term {
                from {
                    protocol icmp;
                }
                then {
                    policer icmp-policer;
                    accept;
                }
            }
        }
    }
    policer icmp-policer {
        if-exceeding {
            bandwidth-limit 1m;
            burst-size-limit 15k;
        }
        then discard;
    }
}
```

Verification

IN THIS SECTION

- Verifying That Ping Works Unless the Limits Are Exceeded | 56

Confirm that the configuration is working properly.

Verifying That Ping Works Unless the Limits Are Exceeded

Purpose

Make sure that the logical system interface is protected against ICMP-based DoS attacks.

Action

Log in to a system that has connectivity to the logical system and run the `ping` command.

```
user@R2> ping 10.0.45.2
PING 10.0.45.2 (10.0.45.2): 56 data bytes
64 bytes from 10.0.45.2: icmp_seq=0 ttl=64 time=1.316 ms
64 bytes from 10.0.45.2: icmp_seq=1 ttl=64 time=1.277 ms
64 bytes from 10.0.45.2: icmp_seq=2 ttl=64 time=1.269 ms
```

```
user@R2> ping 10.0.45.2 size 20000
PING 10.0.45.2 (10.0.45.2): 20000 data bytes
^C
--- 10.0.45.2 ping statistics ---
4 packets transmitted, 0 packets received, 100% packet loss
```

Meaning

When you send a normal ping, the packet is accepted. When you send a ping packet that exceeds the filter limit, the packet is discarded.

SEE ALSO

Example: Creating an Interface on a Logical System

Example: Configuring Filter-Based Forwarding on Logical Systems

IN THIS SECTION

- Requirements | 57
- Overview | 57
- Configuration | 60
- Verification | 68

This example shows how to configure filter-based forwarding within a logical system. The filter classifies packets to determine their forwarding path within the ingress routing device.

Requirements

In this example, no special configuration beyond device initialization is required.

Overview

IN THIS SECTION

- Topology | 59

Filter-based forwarding is supported for IP version 4 (IPv4) and IP version 6 (IPv6).

Use filter-based forwarding for service provider selection when customers have Internet connectivity provided by different ISPs yet share a common access layer. When a shared media (such as a cable modem) is used, a mechanism on the common access layer looks at Layer 2 or Layer 3 addresses and distinguishes between customers. You can use filter-based forwarding when the common access layer is implemented using a combination of Layer 2 switches and a single router.

With filter-based forwarding, all packets received on an interface are considered. Each packet passes through a filter that has match conditions. If the match conditions are met for a filter and you have created a routing instance, filter-based forwarding is applied to a packet. The packet is forwarded based on the next hop specified in the routing instance. For static routes, the next hop can be a specific LSP.

NOTE: Source-class usage filter matching and unicast reverse-path forwarding checks are not supported on an interface configured with filter-based forwarding (FBF).

To configure filter-based forwarding, perform the following tasks:

- Create a match filter on an ingress router or switch. To specify a match filter, include the `filter` *`filter-name`* statement at the `[edit firewall]` hierarchy level. A packet that passes through the filter is compared against a set of rules to classify it and to determine its membership in a set. Once classified, the packet is forwarded to a routing table specified in the accept action in the filter description language. The routing table then forwards the packet to the next hop that corresponds to the destination address entry in the table.

- Create routing instances that specify the routing table(s) to which a packet is forwarded, and the destination to which the packet is forwarded at the `[edit routing-instances]` or `[edit logical-systems` *`logical-system-name`* `routing-instances]` hierarchy level. For example:

```
[edit]
routing-instances {
    routing-table-name1 {
        instance-type forwarding;
        routing-options {
            static {
                route 0.0.0.0/0 nexthop 10.0.0.1;
            }
        }
    }
    routing-table-name2 {
        instance-type forwarding;
        routing-options {
            static {
                route 0.0.0.0/0 nexthop 10.0.0.2;
            }
        }
    }
}
```

- Create a routing table group that adds interface routes to the forwarding routing instances used in filter-based forwarding (FBF), as well as to the default routing instance `inet.0`. This part of the configuration resolves the routes installed in the routing instances to directly connected next hops

on that interface. Create the routing table group at the [edit routing-options] or [edit logical-systems *logical-system-name* routing-options] hierarchy level.

NOTE: Specify inet.0 as one of the routing instances that the interface routes are imported into. If the default instance inet.0 is not specified, interface routes are not imported into the default routing instance.

This example shows a packet filter that directs customer traffic to a next-hop router in the domains, SP 1 or SP 2, based on the packet's source address.

If the packet has a source address assigned to an SP 1 customer, destination-based forwarding occurs using the sp1-route-table.inet.0 routing table. If the packet has a source address assigned to an SP 2 customer, destination-based forwarding occurs using the sp2-route-table.inet.0 routing table. If a packet does not match either of these conditions, the filter accepts the packet, and destination-based forwarding occurs using the standard inet.0 routing table.

One way to make filter-based forwarding work within a logical system is to configure the firewall filter on the logical system that receives the packets. Another way is to configure the firewall filter on the main router and then reference the logical system in the firewall filter. This example uses the second approach. The specific routing instances are configured within the logical system. Because each routing instance has its own routing table, you have to reference the routing instances in the firewall filter, as well. The syntax looks as follows:

```
[edit firewall filter filter-name term term-name]
user@host# set then logical-system logical-system-name routing-instance routing-instance-name
```

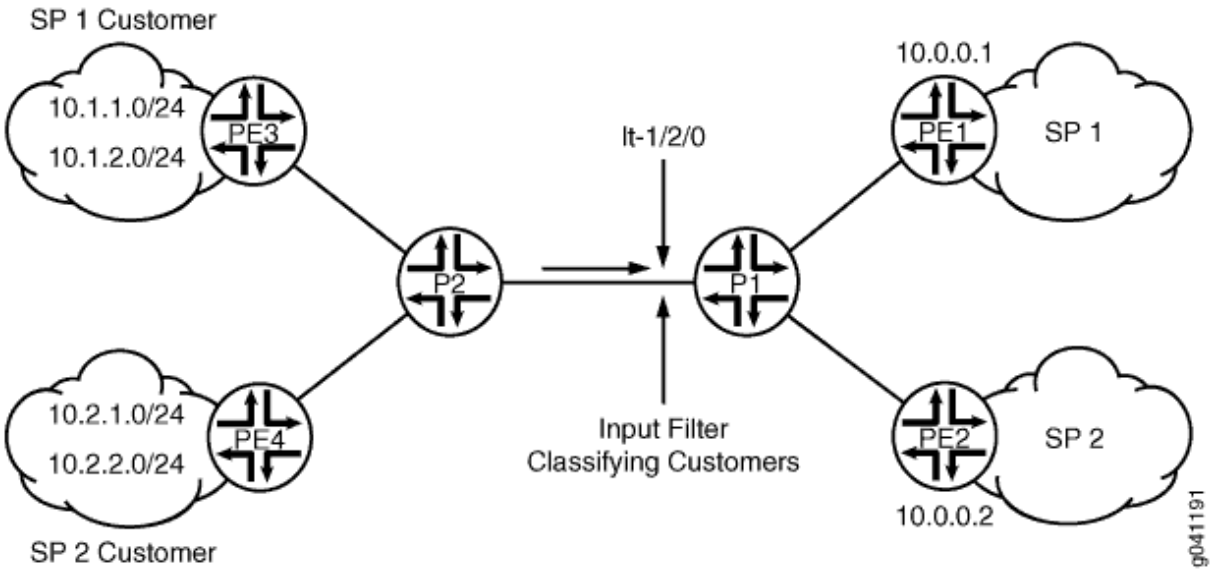
Topology

Figure 9 on page 60 shows the topology used in this example.

On Logical System P1, an input filter classifies packets received from Logical System PE3 and Logical System PE4. The packets are routed based on the source addresses. Packets with source addresses in

the 10.1.1.0/24 and 10.1.2.0/24 networks are routed to Logical System PE1. Packets with source addresses in the 10.2.1.0/24 and 10.2.2.0/24 networks are routed to Logical System PE2.

Figure 9: Logical Systems with Filter-Based Forwarding



To establish connectivity, OSPF is configured on all of the interfaces. For demonstration purposes, loopback interface addresses are configured on the routing devices to represent networks in the clouds.

The "CLI Quick Configuration" on page 61 section shows the entire configuration for all of the devices in the topology. The "Configuring the Routing Instances on the Logical System P1" on page 64 and "Configuring the Firewall Filter on the Main Router" on page 62 sections shows the step-by-step configuration of the ingress routing device, Logical System P1.

Configuration

IN THIS SECTION

- CLI Quick Configuration | 61
- Configuring the Firewall Filter on the Main Router | 62
- Configuring the Routing Instances on the Logical System P1 | 64
- Results | 66

CLI Quick Configuration

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

```
set firewall filter classify-customers term sp1-customers from source-address 10.1.1.0/24
set firewall filter classify-customers term sp1-customers from source-address 10.1.2.0/24
set firewall filter classify-customers term sp1-customers then log
set firewall filter classify-customers term sp1-customers then logical-system P1 routing-
instance sp1-route-table
set firewall filter classify-customers term sp2-customers from source-address 10.2.1.0/24
set firewall filter classify-customers term sp2-customers from source-address 10.2.2.0/24
set firewall filter classify-customers term sp2-customers then log
set firewall filter classify-customers term sp2-customers then logical-system P1 routing-
instance sp2-route-table
set firewall filter classify-customers term default then accept
set logical-systems P1 interfaces lt-1/2/0 unit 10 encapsulation ethernet
set logical-systems P1 interfaces lt-1/2/0 unit 10 peer-unit 9
set logical-systems P1 interfaces lt-1/2/0 unit 10 family inet filter input classify-customers
set logical-systems P1 interfaces lt-1/2/0 unit 10 family inet address 172.16.0.10/30
set logical-systems P1 interfaces lt-1/2/0 unit 13 encapsulation ethernet
set logical-systems P1 interfaces lt-1/2/0 unit 13 peer-unit 14
set logical-systems P1 interfaces lt-1/2/0 unit 13 family inet address 172.16.0.13/30
set logical-systems P1 interfaces lt-1/2/0 unit 17 encapsulation ethernet
set logical-systems P1 interfaces lt-1/2/0 unit 17 peer-unit 18
set logical-systems P1 interfaces lt-1/2/0 unit 17 family inet address 172.16.0.17/30
set logical-systems P1 protocols ospf rib-group fbf-group
set logical-systems P1 protocols ospf area 0.0.0.0 interface all
set logical-systems P1 protocols ospf area 0.0.0.0 interface fxp0.0 disable
set logical-systems P1 routing-instances sp1-route-table instance-type forwarding
set logical-systems P1 routing-instances sp1-route-table routing-options static route 0.0.0.0/0
next-hop 172.16.0.13
set logical-systems P1 routing-instances sp2-route-table instance-type forwarding
set logical-systems P1 routing-instances sp2-route-table routing-options static route 0.0.0.0/0
next-hop 172.16.0.17
set logical-systems P1 routing-options rib-groups fbf-group import-rib inet.0
set logical-systems P1 routing-options rib-groups fbf-group import-rib sp1-route-table.inet.0
set logical-systems P1 routing-options rib-groups fbf-group import-rib sp2-route-table.inet.0
set logical-systems P2 interfaces lt-1/2/0 unit 2 encapsulation ethernet
set logical-systems P2 interfaces lt-1/2/0 unit 2 peer-unit 1
set logical-systems P2 interfaces lt-1/2/0 unit 2 family inet address 172.16.0.2/30
```

```
set logical-systems P2 interfaces lt-1/2/0 unit 6 encapsulation ethernet
set logical-systems P2 interfaces lt-1/2/0 unit 6 peer-unit 5
set logical-systems P2 interfaces lt-1/2/0 unit 6 family inet address 172.16.0.6/30
set logical-systems P2 interfaces lt-1/2/0 unit 9 encapsulation ethernet
set logical-systems P2 interfaces lt-1/2/0 unit 9 peer-unit 10
set logical-systems P2 interfaces lt-1/2/0 unit 9 family inet address 172.16.0.9/30
set logical-systems P2 protocols ospf area 0.0.0.0 interface all
set logical-systems P2 protocols ospf area 0.0.0.0 interface fxp0.0 disable
set logical-systems PE1 interfaces lt-1/2/0 unit 14 encapsulation ethernet
set logical-systems PE1 interfaces lt-1/2/0 unit 14 peer-unit 13
set logical-systems PE1 interfaces lt-1/2/0 unit 14 family inet address 172.16.0.14/30
set logical-systems PE1 interfaces lo0 unit 3 family inet address 172.16.1.1/32
set logical-systems PE1 protocols ospf area 0.0.0.0 interface all
set logical-systems PE1 protocols ospf area 0.0.0.0 interface fxp0.0 disable
set logical-systems PE2 interfaces lt-1/2/0 unit 18 encapsulation ethernet
set logical-systems PE2 interfaces lt-1/2/0 unit 18 peer-unit 17
set logical-systems PE2 interfaces lt-1/2/0 unit 18 family inet address 172.16.0.18/30
set logical-systems PE2 interfaces lo0 unit 4 family inet address 172.16.2.2/32
set logical-systems PE2 protocols ospf area 0.0.0.0 interface all
set logical-systems PE2 protocols ospf area 0.0.0.0 interface fxp0.0 disable
set logical-systems PE3 interfaces lt-1/2/0 unit 1 encapsulation ethernet
set logical-systems PE3 interfaces lt-1/2/0 unit 1 peer-unit 2
set logical-systems PE3 interfaces lt-1/2/0 unit 1 family inet address 172.16.0.1/30
set logical-systems PE3 interfaces lo0 unit 1 family inet address 10.1.1.1/32
set logical-systems PE3 interfaces lo0 unit 1 family inet address 10.1.2.1/32
set logical-systems PE3 protocols ospf area 0.0.0.0 interface all
set logical-systems PE3 protocols ospf area 0.0.0.0 interface fxp0.0 disable
set logical-systems PE4 interfaces lt-1/2/0 unit 5 encapsulation ethernet
set logical-systems PE4 interfaces lt-1/2/0 unit 5 peer-unit 6
set logical-systems PE4 interfaces lt-1/2/0 unit 5 family inet address 172.16.0.5/30
set logical-systems PE4 interfaces lo0 unit 2 family inet address 10.2.1.1/32
set logical-systems PE4 interfaces lo0 unit 2 family inet address 10.2.2.1/32
set logical-systems PE4 protocols ospf area 0.0.0.0 interface all
set logical-systems PE4 protocols ospf area 0.0.0.0 interface fxp0.0 disable
```

Configuring the Firewall Filter on the Main Router

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the firewall filter on the main router:

1. Configure the source addresses for SP1 customers.

```
[edit firewall filter classify-customers term sp1-customers]
user@host# set from source-address 10.1.1.0/24
user@host# set from source-address 10.1.2.0/24
```

2. Configure the actions that are taken when packets are received with the specified source addresses.

   To track the action of the firewall filter, a log action is configured. The sp1-route-table.inet.0 routing table on Logical System P1 routes the packets.

```
[edit firewall filter classify-customers term sp1-customers]
user@host# set then log
user@host# set then logical-system P1 routing-instance sp1-route-table
```

3. Configure the source addresses for SP2 customers.

```
[edit firewall filter classify-customers term sp2-customers]
user@host# set from source-address 10.2.1.0/24
user@host# set from source-address 10.2.2.0/24
```

4. Configure the actions that are taken when packets are received with the specified source addresses.

   To track the action of the firewall filter, a log action is configured. The sp2-route-table.inet.0 routing table on Logical System P1 routes the packet.

```
[edit firewall filter classify-customers term sp2-customers]
user@host# set then log
user@host# set then logical-system P1 routing-instance sp2-route-table
```

5. Configure the action to take when packets are received from any other source address.

   All of these packets are simply accepted and routed using the default IPv4 unicast routing table, inet.0.

```
[edit firewall filter classify-customers term default]
user@host# set then accept
```

Configuring the Routing Instances on the Logical System P1

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the routing instances on a logical system:

1. Configure the interfaces on the logical system.

```
[edit logical-systems P1 interfaces lt-1/2/0]
user@host# set unit 10 encapsulation ethernet
user@host# set unit 10 peer-unit 9
user@host# set unit 10 family inet address 172.16.0.10/30
user@host# set unit 13 encapsulation ethernet
user@host# set unit 13 peer-unit 14
user@host# set unit 13 family inet address 172.16.0.13/30
user@host# set unit 17 encapsulation ethernet
user@host# set unit 17 peer-unit 18
user@host# set unit 17 family inet address 172.16.0.17/30
```

2. Assign the `classify-customers` firewall filter to router interface lt-1/2/0.10 as an input packet filter.

```
[edit logical-systems P1 interfaces lt-1/2/0]
user@host# set unit 10 family inet filter input classify-customers
```

3. Configure connectivity, using either a routing protocol or static routing.

   As a best practice, disable routing on the management interface.

```
[edit  logical-systems P1 protocols ospf area 0.0.0.0]
user@host# set interface all
user@host# set interface fxp0.0 disable
```

4. Create the routing instances.

   These routing instances are referenced in the `classify-customers` firewall filter.

The forwarding instance type provides support for filter-based forwarding, where interfaces are not associated with instances. All interfaces belong to the default instance, in this case Logical System P1.

```
[edit logical-systems P1 routing-instances]
user@host# set sp1-route-table instance-type forwarding
user@host# set sp2-route-table instance-type forwarding
```

5. Resolve the routes installed in the routing instances to directly connected next hops.

```
[edit logical-systems P1 routing-instances]
user@host# set sp1-route-table routing-options static route 0.0.0.0/0 next-hop 172.16.0.13
user@host# set sp2-route-table routing-options static route 0.0.0.0/0 next-hop 172.16.0.17
```

6. Group together the routing tables to form a routing table group.

   The first routing table, inet.0, is the primary routing table, and the additional routing tables are the secondary routing tables.

   The primary routing table determines the address family of the routing table group, in this case IPv4.

```
[edit logical-systems P1 routing-options]
user@host# set rib-groups fbf-group import-rib inet.0
user@host# set rib-groups fbf-group import-rib sp1-route-table.inet.0
user@host# set rib-groups fbf-group import-rib sp2-route-table.inet.0
```

7. Apply the routing table group to OSPF.

   This causes the OSPF routes to be installed into all the routing tables in the group.

```
[edit logical-systems P1 protocols ospf]
user@host# set rib-group fbf-group
```

8. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Results

Confirm your configuration by issuing the `show firewall` and `show logical-systems P1` commands.

```
user@host# show firewall
filter classify-customers {
    term sp1-customers {
        from {
            source-address {
                10.1.1.0/24;
                10.1.2.0/24;
            }
        }
        then {
            log;
            logical-system P1 routing-instance sp1-route-table;
        }
    }
    term sp2-customers {
        from {
            source-address {
                10.2.1.0/24;
                10.2.2.0/24;
            }
        }
        then {
            log;
            logical-system P1 routing-instance sp2-route-table;
        }
    }
    term default {
        then accept;
    }
}
```

```
user@host# show logical-systems P1
interfaces {
    lt-1/2/0 {
        unit 10 {
            encapsulation ethernet;
            peer-unit 9;
```

```
            family inet {
                filter {
                    input classify-customers;
                }
                address 172.16.0.10/30;
            }
        }
        unit 13 {
            encapsulation ethernet;
            peer-unit 14;
            family inet {
                address 172.16.0.13/30;
            }
        }
        unit 17 {
            encapsulation ethernet;
            peer-unit 18;
            family inet {
                address 172.16.0.17/30;
            }
        }
    }
}
protocols {
    ospf {
        rib-group fbf-group;
        area 0.0.0.0 {
            interface all;
            interface fxp0.0 {
                disable;
            }
        }
    }
}
routing-instances {
    sp1-route-table {
        instance-type forwarding;
        routing-options {
            static {
                route 0.0.0.0/0 next-hop 172.16.0.13;
            }
        }
    }
```

```
    sp2-route-table {
        instance-type forwarding;
        routing-options {
            static {
                route 0.0.0.0/0 next-hop 172.16.0.17;
            }
        }
    }
}
routing-options {
    rib-groups {
        fbf-group {
            import-rib [ inet.0 sp1-route-table.inet.0 sp2-route-table.inet.0 ];
        }
    }
}
```

Verification

IN THIS SECTION

- Pinging with Specified Source Addresses | 68
- Verifying the Firewall Filter | 70

Confirm that the configuration is working properly.

Pinging with Specified Source Addresses

Purpose

Send some ICMP packets across the network to test the firewall filter.

Action

1. Log in to Logical System PE3.

```
user@host> set cli logical-system PE3
Logical system: PE3
```

2. Run the `ping` command, pinging the lo0.3 interface on Logical System PE1.

   The address configured on this interface is 172.16.1.1.

   Specify the source address 10.1.2.1, which is the address configured on the lo0.1 interface on Logical System PE3.

```
user@host:PE3> ping 172.16.1.1 source 10.1.2.1
PING 172.16.1.1 (172.16.1.1): 56 data bytes
64 bytes from 172.16.1.1: icmp_seq=0 ttl=62 time=1.444 ms
64 bytes from 172.16.1.1: icmp_seq=1 ttl=62 time=2.094 ms
^C
--- 172.16.1.1 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.444/1.769/2.094/0.325 ms
```

3. Log in to Logical System PE4.

```
user@host:PE3> set cli logical-system PE4
Logical system: PE4
```

4. Run the `ping` command, pinging the lo0.4 interface on Logical System PE2.

   The address configured on this interface is 172.16.2.2.

   Specify the source address 10.2.1.1, which is the address configured on the lo0.2 interface on Logical System PE4.

```
user@host:PE4> ping 172.16.2.2 source 10.2.1.1
PING 172.16.2.2 (172.16.2.2): 56 data bytes
64 bytes from 172.16.2.2: icmp_seq=0 ttl=62 time=1.473 ms
64 bytes from 172.16.2.2: icmp_seq=1 ttl=62 time=1.407 ms
^C
--- 172.16.2.2 ping statistics ---
```

```
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.407/1.440/1.473/0.033 ms
```

Meaning

Sending these pings activates the firewall filter actions.

Verifying the Firewall Filter

Purpose

Make sure the firewall filter actions take effect.

Action

1. Log in to Logical System P1.

```
user@host> set cli logical-system P1
Logical system: P1
```

2. Run the `show firewall log` command on Logical System P1.

```
user@host:P1> show firewall log
Log :
Time      Filter    Action Interface     Protocol        Src Addr           Dest Addr
13:52:20  pfe       A      lt-1/2/0.10   ICMP            10.2.1.1           172.16.2.2
13:52:19  pfe       A      lt-1/2/0.10   ICMP            10.2.1.1           172.16.2.2
13:51:53  pfe       A      lt-1/2/0.10   ICMP            10.1.2.1           172.16.1.1
13:51:52  pfe       A      lt-1/2/0.10   ICMP            10.1.2.1           172.16.1.1
```

SEE ALSO

- Configuring Filter-Based Forwarding
- Copying and Redirecting Traffic with Port Mirroring and Filter-Based Forwarding
- Example: Configuring Filter-Based Forwarding on the Source Address
- Using Filter-Based Forwarding to Export Monitored Traffic to Multiple Destinations
- Filter-Based Forwarding Overview

Layer 2 Learning and Forwarding on Logical Systems

IN THIS SECTION

- Layer 2 Learning and Forwarding in a Logical System Overview | 71
- Enabling Layer 2 Learning and Forwarding in a Logical System | 72
- Example: Configuring Layer 2 Learning and Forwarding and RSTP in a Logical System | 73

Logical Systems enables you to configure Layer 2 MAC address and VLAN learning and forwarding properties in support of Layer 2 bridging. For more information, see the following topics:

Layer 2 Learning and Forwarding in a Logical System Overview

You can partition a single physical router into multiple logical devices called *logical systems* that perform independent routing tasks. Logical systems perform a subset of the actions of a physical router and have their own unique routing tables, interfaces, policies, and routing instances. On MX Series routers only, you can enable Layer 2 learning and forwarding in a logical system for bridge domains or other virtual-switch routing instances.

When enabling Layer 2 learning and forwarding in a logical system for bridge domains or other virtual-switch routing instances, the following guidelines apply:

- You can only configure 16 logical systems.
- Logging is performed for the entire device and not per logical system.
- You cannot restart Layer 2 learning for an individual logical system.

SEE ALSO

Enabling Layer 2 Learning and Forwarding in a Logical System | 72

Example: Configuring Layer 2 Learning and Forwarding and RSTP in a Logical System | 73

Enabling Layer 2 Learning and Forwarding in a Logical System

On MX Series router, you can enable Layer 2 learning and forwarding in a logical system for bridge domains or other virtual-switch routing instances.

Before you begin, configure the interfaces for the logical system.

To configure Layer 2 learning and forwarding in a logical system for bridge domains or other virtual-switch routing instances:

1. Enable configuration of a logical system:

```
[edit]
user@host# edit logical-systems logical-system-name
```

   For detailed information about logical systems, see the Logical Systems User Guide for Routers and Switches.

2. Enable configuration of a virtual-switch routing instance:

```
[edit logical-systems logical-system-name]
user@host# edit routing-instances routing-instance-name
user@host# set instance-type virtual-switch
```

3. Configure the set of bridge-domains or other virtual-switch routing instances.
4. Configure Layer 2 learning and forwarding properties for a set of bridge domains:
   a. Enable configuration of Layer 2 learning and forwarding properties:

```
[edit logical-systems logical-system-name]
user@host# edit switch-options
```

   b. Configure Layer 2 learning and forwarding properties. For more information, see *Understanding Layer 2 Learning and Forwarding for Bridge Domains* .
5. Verify the configuration:

```
[edit logical-systems logical-system-name switch-options]
user@host# top
user@host# show logical-systems

logical-system-name {
```

```
    interfaces {
        ...interface-configurations...
    }
    routing-instances {
        instance-type virtual-switch;

    }
    bridge-domains{
        ...bridge-domain-configuration...
    }
    switch-options {
        interface logical-interface-name {
            ...layer-2-learning-and-forwarding-configuration...
        }
    }
    protocols {
        (rstp | mstp | vstp) {
            interface interface-name;
            ...spanning-tree-protocol-configuration...
        }
    }
}
```

SEE ALSO

Layer 2 Learning and Forwarding in a Logical System Overview | 71

Example: Configuring Layer 2 Learning and Forwarding and RSTP in a Logical System | 73

Example: Configuring Layer 2 Learning and Forwarding and RSTP in a Logical System

The following example configures a logical system and routing instance with its own bridge domain (**bd1**), switch options, and spanning- tree protocol (**rstp**).

```
[edit]
interfaces {
    ge-5/0/1 {
        flexible-vlan-tagging;
```

```
    }
}
logical-systems {
    logical-sys1 {
        interfaces {
            ge-5/0/1 {
                unit 0 {
                    family bridge {
                        interface-mode trunk;
                        vlan-id-list 1-5;
                    }
                }
                unit 3 {
                    family bridge {
                        interface-mode trunk;
                        vlan-id-list 11-15;
                    }
                }
            }
            ge-5/0/2 {
                unit 0 {
                    family bridge {
                        interface-mode trunk;
                        vlan-id-list 1-5;
                    }
                }
            }
        }
        routing-instances {
            routing-inst-1 {
                interface ge-5/0/2;
                instance-type virtual-switch;
                bridge-domains {
                    vlan-id 1;
                }
                protocols {
                    rstp {
                        interface ge-5/0/2;
                    }
                }
            }
        }
        bridge-domains {
```

```
            bd-1 {
                vlan-id 1;
            }
        }
        switch-options {
            interface ge-5/0/1.3 {
                interface-mac-limit {
                    1400;
                    packet-action drop;
                }
            }
        }
        protocols {
            rstp {
                interface ge-5/0/1;
            }
        }
    }
}
```

NOTE: This is not a complete router configuration.

SEE ALSO

Layer 2 Learning and Forwarding in a Logical System Overview | 71

Enabling Layer 2 Learning and Forwarding in a Logical System | 72

Multichassis Link Aggregation on Logical Systems Overview

IN THIS SECTION

Sample Configuration Scenario for MC-LAG on Logical Systems | 78

Guidelines for Configuring MC-LAG on Logical Systems | 80

On MX Series routers, EX9200, and QFX10000 switches, multichassis link aggregation (MC-LAG) enables a device to form a logical LAG interface with two or more other devices. MC-LAG provides additional benefits over traditional LAG in terms of node-level redundancy, multihoming support, and a loop-free Layer 2 network without running Spanning Tree Protocol (STP). The MC-LAG devices use Inter-Chassis Control Protocol (ICCP) to exchange the control information between two MC-LAG network devices. Starting in Junos OS Release 14.1, you can configure MC-LAG interfaces on logical systems within a router. Starting with Junos OS Release 15.1, you can configure MC-LAG interfaces on logical systems on EX9200 switches.

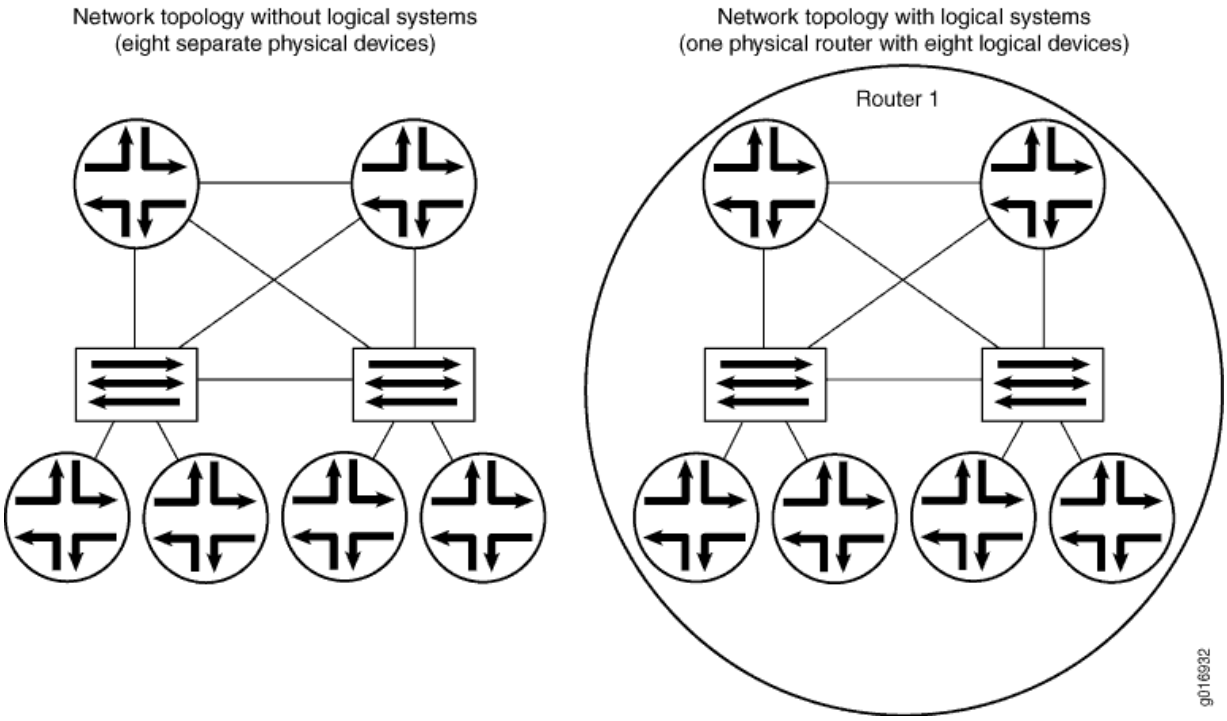
NOTE: On QFX10008 switches, Layer 2 and Layer 3 IRB interfaces are not supported under the `[edit logical-systems]` hierarchy.

To configure ICCP for MC-LAG interfaces on logical systems, include the `iccp` statement at the `[edit logical-systems` *`logical-system-name`* `protocols]` hierarchy level. To view ICCP information for MC-LAG on logical systems, use the `show iccp logical-system` *`logical-system-name`* command. To view ARP statistics or remote MAC addresses for the multichassis aggregated Ethernet nodes for all or specified redundancy groups on a logical system, use the `show l2-learning redundancy-groups` *`group-name`* `logical-system` *`logical-system-name`* `(arp-statistics | remote-macs)` command. To view neighbor discovery (ND) statistical details for multichassis aggregated Ethernet nodes on redundancy groups of a logical group, use the `show l2-learning redundancy-groups` *`group-name`* `logical-system` *`logical-system-name`* `nd-statistics` command.

Logical systems enable effective, optimal segregation of a single router or switch into multiple virtual partitions, which can be configured and managed by diversified entities. Logical systems perform a subset of the actions of a physical router or switch and have their own unique routing tables, interfaces, policies, and routing instances. A set of logical systems within a single router or switch can handle the functions previously performed by several small routers or switches. As shown on the right side of

Figure 10 on page 77, a set of logical systems within a single router can handle the functions previously performed by several small routers.

Figure 10: Comparison of Devices With and Without Logical Systems



In a network deployment that contains MC-LAG interfaces, you can configure such interfaces on logical systems contained within a router or switch. When you configure multichassis aggregated Ethernet interfaces on a logical system, you must ensure that these interfaces are added with the same multichassis aggregated Ethernet identification number and redundancy group identifier for the MC-LAG on both the peers or devices that are connected by the multichassis aggregated Ethernet interfaces. It is not necessary to specify the same logical system name on both the peers; however, you must ensure that ICCP to associate the routing or switching devices contained in a redundancy group is defined on both the peers within the logical systems of the devices. Such a configuration ensures that all the packets are transmitted using ICCP within the logical system network. The logical system information is added and removed by the ICCP process to prevent each packet from containing the logical system details. This behavior enables multiple disjoint users to employ MC-LAG capabilities within their networks transparently and seamlessly. A unique ICCP definition for a logical system is created, thereby enabling you to completely manage the ICCP parameters on one logical system without the need for access permissions to view other logical system networks on the same device. Configuration of MC-LAG interfaces on logical systems enables MC-LAG to be used across multiple routing tables and switch forwarding tables in active-active and active-standby modes of MC-LAG interfaces.

Because the Layer 2 address learning process supports logical systems, the ARP, neighbor discovery, and MAC synchronization packets that are traversing a multichassis aggregated Ethernet interface use the logical system:routing instance (LS:RI) combination to map the packets to the correct routing instance in a logical system. Link Aggregation Control Protocol (LACP) does not require the LS-RI combination to be identified because it operates on physical interfaces and is unique within a chassis. For a service, in the set of provider edge (PE) routers providing the service, the service ID distinguishes the routing instances in a logical system because it is unique for a logical system across a routing instance. MC-LAG is configured on the aggregated Ethernet (ae-) bundle interface. An ae- interface is a logical interface and is globally unique, which causes the MC-LAG configuration to be exclusive and separate for a router or switch. You can add ae- interfaces in an MC-LAG configuration to be part of a logical system and use it throughout that particular logical system.

Sample Configuration Scenario for MC-LAG on Logical Systems

Consider a sample scenario in which two MX Series routers, MX1 and MX2, are connected using an aggregated Ethernet interface that is enabled with MC-LAG. The peers in an MC-LAG use an interchassis link-protection link (ICL-PL) to replicate forwarding information across the peers. Additionally, ICCP propagates the operational state of MC-LAG members through the ICL-PL. The two PE devices, MX1 and MX2, each have a LAG connected to the CE devices, CE1 and CE2. Four logical systems are defined on each of the PE devices, MX1 and MX2. CE-1 and CE-2 can be part of the same VLAN with the same VLAN ID and located in the same IP subnet for MC-LAG in two different logical systems. All four logical system entities can work independently in MX1 and MX2.

The ICCP process can manage multiple client-server connections with its peer ICCP instances based on the ICCP configuration for the logical system:routing instance (LS-RI) combinations. Each ICCP connection is associated with an LS-RI combination. For example, with two routing instances, IP1 and IP2, on each of the logical systems, LS1 and LS2, the following mapping is performed for ICCP settings:

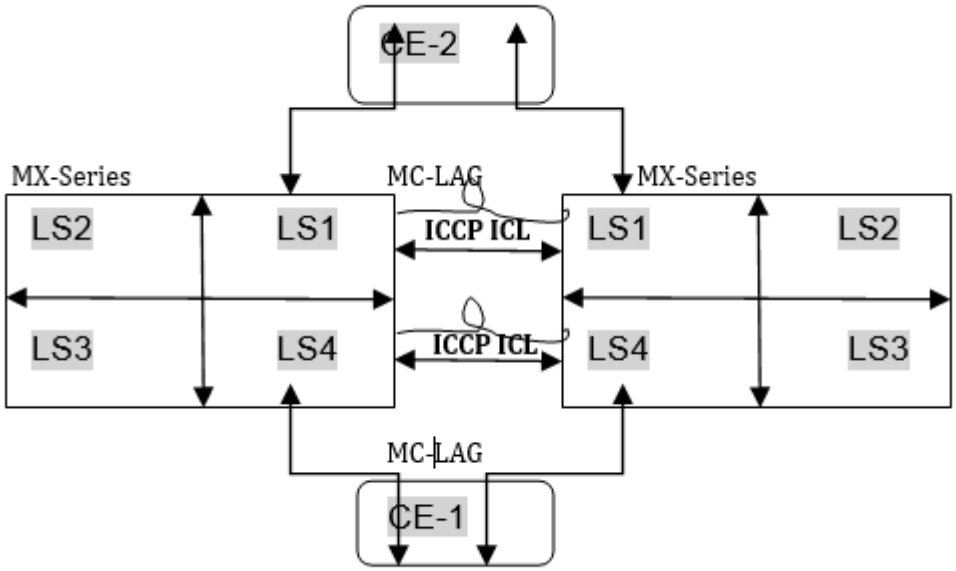
[ICCP] (LS1) (IP1) < = = > (IP2) (LS1) [ICCP] within LS1 network.

[ICCP] (LS2) (IP1) < = = > (IP2) (LS2) [ICCP] within LS2 network.

An ICCP instance in a logical system is linked with the ICCP instance of the peer logical system. The ICCP application transmits the relevant routing index depending on the LS:RI combination to the BFD process, when BFD is configured in your topology.

Figure 11 on page 79 shows the interconnection among logical systems on MX Series routers configured with MC-LAG.

Figure 11: Logical Systems with MC-LAG



The Layer 2 address learning process (l2ald) transmits and receives Address Learning Protocol (ARP), neighbor discovery, and MAC synchronization packets with the LS-RI information. When the peer MAC synchronization packets are received, l2ald decodes the logical system details from the packet and determines whether an identical logical system has been previously created on the router. If a match is found for the logical system, the MAC forwarding entry for the corresponding bridge table for an interface bridge domain is created. If the logical system in the received packet does not match the defined logical system on the device, for the MAC synchronization packet, the default logical instance is used for processing. Similarly, upon receipt of the ARP and neighbor discovery packets, l2ald decapsulates the logical system information from the packets and determines if the corresponding logical instance has been previously created. If a match is found for the logical system, the ARP and neighbor discovery packets are processed according to the Layer 3 index that is unique in the system. The programming kernel entry might not require any logical system information since it is programmed on a Layer 3 index which is unique in the system. If the logical system in the received packet does not match the defined logical system on the device, for the ARP and neighbor discovery packets, the default logical instance is used for processing. The routing instance is determined using the service ID attribute. The logical system information is forwarded to ICCP, which in turn identifies the appropriate ICCP interface for the logical system and sends packets over it.

Guidelines for Configuring MC-LAG on Logical Systems

Keep the following points in mind while configuring MC-LAG interfaces on logical systems:

- You cannot use a single chassis to function as a provider edge (PE) device and a customer edge (CE) device in different logical systems.
- You cannot use a single chassis to function as two PE devices by configuring logical systems on the chassis and ICCP. ICL links between the two logical systems because the multichassis aggregated Ethernet ID is unique in a router or switch.
- Logical interfaces (IFLs) on the same `mc-ae` interface cannot be configured across multiple logical systems. In other words, in a multichassis link aggregation (MC-LAG) with both logical systems and logical interfaces (such as `mc-ae ae0 unit 0`), the same logical interface cannot be shared between logical systems.
- IGMP snooping in MC-LAG topologies with logical systems is not supported.
- VPLS and VPN protocols with MC-LAG in active-standby mode is not supported.
- Logical system information is not communicated to the peer chassis because this detail is derived from an ICCP instance.

Release History Table

Release	Description
15.1	Starting with Junos OS Release 15.1, you can configure MC-LAG interfaces on logical systems on EX9200 switches.
14.1	Starting in Junos OS Release 14.1, you can configure MC-LAG interfaces on logical systems within a router.

VPNs and VPLS

IN THIS SECTION

Introduction to VPLS | 81

- Example: Using Logical Systems to Configure Provider Edge and Provider Routers in a Layer 3 VPN and VPLS Scenario | 82

A VPN is an encrypted connection over the Internet from a device to a network which prevents unauthorized access eavesdropping on the traffic, and allows the user to conduct work remotely. For more information, see the following topics:

Introduction to VPLS

VPLS is an Ethernet-based point-to-multipoint Layer 2 VPN. It allows you to connect geographically dispersed Ethernet local area networks (LAN) sites to each other across an MPLS backbone. For customers who implement VPLS, all sites appear to be in the same Ethernet LAN even though traffic travels across the service provider's network.

VPLS, in its implementation and configuration, has much in common with a Layer 2 VPN. In VPLS, a packet originating within a service provider customer's network is sent first to a customer edge (CE) device (for example, a router or Ethernet switch). It is then sent to a provider edge (PE) router within the service provider's network. The packet traverses the service provider's network over a MPLS label-switched path (LSP). It arrives at the egress PE router, which then forwards the traffic to the CE device at the destination customer site.

NOTE: In the VPLS documentation, the word *router* in terms such as *PE router* is used to refer to any device that provides routing functions.

The difference is that for VPLS, packets can traverse the service provider's network in point-to-multipoint fashion, meaning that a packet originating from a CE device can be broadcast to all the PE routers participating in a VPLS routing instance. In contrast, a Layer 2 VPN forwards packets in point-to-point fashion only.

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.

Example: Using Logical Systems to Configure Provider Edge and Provider Routers in a Layer 3 VPN and VPLS Scenario

IN THIS SECTION

- Requirements | 82
- Overview | 82
- Configuration | 84
- Verification | 120

This example provides step-by-step procedures to configure provider edge (PE) and provider (P) routers in a VPN and VPLS scenario using logical systems.

Requirements

In this example, no special configuration beyond device initialization is required.

Overview

IN THIS SECTION

- Topology | 82

In this example, VPNs are used to separate customer traffic across a provider backbone.

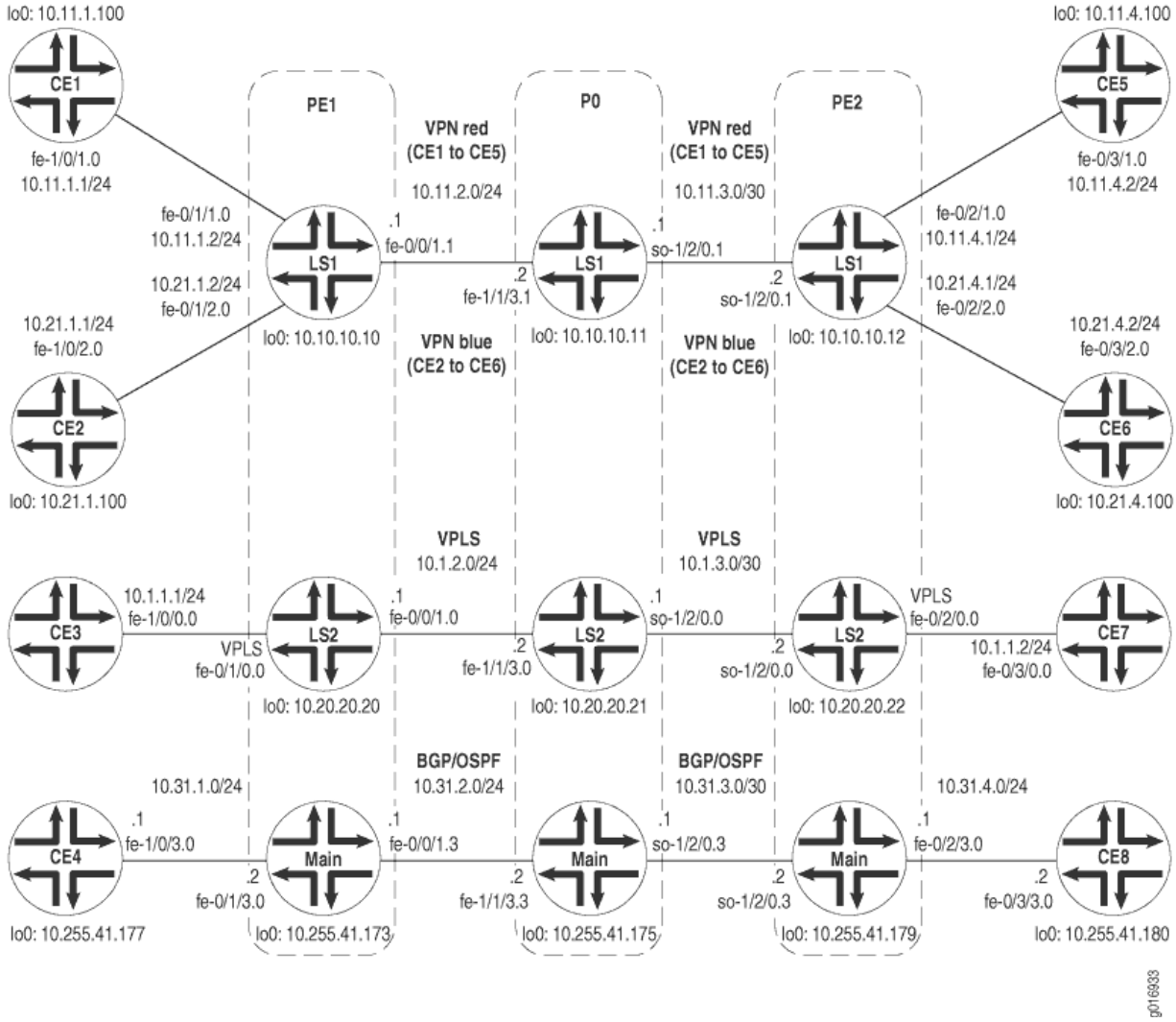
Topology

Figure 12 on page 83 shows four pairs of 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.
- Routers CE4 and CE8 are connected with standard protocols.

Two logical systems are configured on 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 Logical System LS2, and the remaining routers use the main router portion of routers PE1, P0, and PE2.

Figure 12: Provider Edge and Provider Logical System Topology Diagram



On Router PE1, two VPN routing and forwarding (VRF) routing instances are created in Logical System LS1. The routing instances are called red and blue. The example configures the customer edge (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. A logical interface at **fe-0/0/1.1** connects to Logical System LS1 on Router P0. A VPLS routing instance is in Logical System LS2. The logical interface is configured so that traffic from Router CE3 is sent into the VPLS domain. This logical interface connects to Logical System LS2 on Router P0. The example also contains an administrator for Logical System LS1. The logical

system administrator is responsible for the maintenance of this logical system. Finally, the example shows how to configure a logical interface to interconnect Router CE4 with the main router portion of Router PE1.

Router PE2 has the two VRF routing instances in Logical System LS1: red and blue. The CE-facing logical interfaces enable traffic from Router CE5 to be placed in the red VPN, and traffic from Router CE6 in the blue VPN. One logical interface on **so-1/2/0.1** connects to Logical System LS1 on Router P0. The VPLS routing instance is configured in Logical System LS2. A logical interface enables traffic from Router CE7 to be sent into the VPLS domain and connects to Logical System LS2 on Router P0. The example shows how to 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 Logical System LS1 and viewing privileges for Logical System LS2.

On Router P0, the example shows how to configure Logical Systems LS1, LS2, and the main router. You must configure physical interface properties at the main router `[edit interfaces]` hierarchy level. Next, the example shows how to configure protocols (such as RSVP, MPLS, BGP, and IS-IS), routing options, and policy options for the logical systems. Last, the example shows how to configure the same administrator for Logical System LS1 that is configured on Router PE1. This system administrator for Logical System LS2 has permission to view the LS2 configuration, but not change the configuration for Logical System LS2.

Logical System LS1 transports traffic for the red VPN that exists between routers CE1 and CE5. Logical System 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. The main router transports traffic between routers CE4 and CE8. The example shows how to configure the interfaces and routing protocols (OSPF, BGP) to connect to the main router portion of routers PE1 and PE2.

Configuration

IN THIS SECTION

- Configuring Interfaces on the Customer Edge Devices | **85**
- Configuring Router PE1 | **87**
- Configuring Router PE2 | **90**
- Configuring Router P0 | **93**
- Results | **95**

To configure the PE and P routers in logical systems involves performing the following tasks:

Configuring Interfaces on the Customer Edge Devices

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

1. On Router CE1, configure OSPF to connect to the red VPN in Logical System LS1 on Router PE1.

```
user@CE1# set interfaces fe-1/0/1 vlan-tagging
user@CE1# set interfaces fe-1/0/1 unit 0 description "routing-instance red CE"
user@CE1# set interfaces fe-1/0/1 unit 0 vlan-id 101
user@CE1# set interfaces fe-1/0/1 unit 0 family inet address 10.11.1.1/24
user@CE1# set interfaces lo0 unit 0 family inet address 10.11.1.100/32
user@CE1# set protocols ospf area 0.0.0.0 interface fe-1/0/1.0
user@CE1# set protocols ospf area 0.0.0.0 interface lo0.0
```

2. On Router CE2, configure BGP to connect to the blue VPN in Logical System LS1 on Router PE1.

```
user@CE2# set interfaces fe-1/0/2 vlan-tagging
user@CE2# set interfaces fe-1/0/2 unit 0 description "routing-instance blue CE"
user@CE2# set interfaces fe-1/0/2 unit 0 vlan-id 102
user@CE2# set interfaces fe-1/0/2 unit 0 family inet address 10.21.1.1/24
user@CE2# set interfaces lo0 unit 0 family inet address 10.21.1.100/32
user@CE2# set policy-options policy-statement export_loopback from route-filter
10.21.1.100/32 exact
user@CE2# set policy-options policy-statement export_loopback then accept
user@CE2# set protocols bgp export export_loopback
user@CE2# set protocols bgp group to_PE type external
user@CE2# set protocols bgp group to_PE local-address 10.21.1.1
user@CE2# set protocols bgp group to_PE peer-as 100
user@CE2# set protocols bgp group to_PE neighbor 10.21.1.2
user@CE2# set routing-options autonomous-system 200
```

3. 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.

```
user@CE3# set interfaces fe-1/0/0 vlan-tagging
user@CE3# set interfaces fe-1/0/0 unit 0 description "vpls interface"
```

```
user@CE3# set interfaces fe-1/0/0 unit 0 vlan-id 600
user@CE3# set interfaces fe-1/0/0 unit 0 family inet address 10.1.1.1/24
```

4. On Router CE4, configure the Fast Ethernet interface to connect with the main router at Router PE1.

```
user@CE4# set interfaces fe-1/0/3 vlan-tagging
user@CE4# set interfaces fe-1/0/3 unit 0 description "main router interface"
user@CE4# set interfaces fe-1/0/3 unit 0 vlan-id 103
user@CE4# set interfaces fe-1/0/3 unit 0 family inet address 10.31.1.1/24
user@CE4# set interfaces lo0 unit 0 family inet address 10.255.41.177/32
```

5. On Router CE5, configure OSPF to connect to the red VPN in Logical System LS1 on Router PE2.

```
user@CE5# set interfaces fe-0/3/1 vlan-tagging
user@CE5# set interfaces fe-0/3/1 unit 0 description "routing-instance red CE"
user@CE5# set interfaces fe-0/3/1 unit 0 vlan-id 101
user@CE5# set interfaces fe-0/3/1 unit 0 family inet address 10.11.4.2/24
user@CE5# set interfaces lo0 unit 0 family inet address 10.11.4.100/32
user@CE5# set protocols ospf area 0.0.0.0 interface fe-0/3/1.0
user@CE5# set protocols ospf area 0.0.0.0 interface lo0.0
user@CE5# set system login class LS1admin logical-system LS1
user@CE5# set system login class LS1admin permissions all
user@CE5# set system login class LS1onlooker logical-system LS2
user@CE5# set system login class LS1onlooker permissions view
user@CE5# set system login user LS1admin class LS1admin
```

6. On Router CE6, configure BGP to connect to the blue VPN in Logical System LS1 on Router PE2.

```
user@CE6# set interfaces fe-0/3/2 vlan-tagging
user@CE6# set interfaces fe-0/3/2 unit 0 description "routing-instance blue CE"
user@CE6# set interfaces fe-0/3/2 unit 0 vlan-id 102
user@CE6# set interfaces fe-0/3/2 unit 0 family inet address 10.21.4.2/24
user@CE6# set interfaces lo0 unit 0 family inet address 10.21.4.100/32
user@CE6# set routing-options autonomous-system 300
user@CE6# set protocols bgp export export_loopback
user@CE6# set protocols bgp group to_PE type external
user@CE6# set protocols bgp group to_PE local-address 10.21.4.2
user@CE6# set protocols bgp group to_PE peer-as 100
user@CE6# set protocols bgp group to_PE neighbor 10.21.4.1
user@CE6# set policy-options policy-statement export_loopback from route-filter
```

```
10.21.4.100/32 exact
user@CE6# set policy-options policy-statement export_loopback then accept
```

7. 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.

```
user@CE7# set interfaces fe-0/3/0 vlan-tagging
user@CE7# set interfaces fe-0/3/0 unit 0 description "vpls interface"
user@CE7# set interfaces fe-0/3/0 unit 0 vlan-id 600
user@CE7# set interfaces fe-0/3/0 unit 0 family inet address 10.1.1.2/24
```

8. On Router CE8, configure the Fast Ethernet interface to connect with the main router at Router PE2.

```
user@CE8# set interfaces fe-0/3/3 vlan-tagging
user@CE8# set interfaces fe-0/3/3 unit 0 description "main router interface"
user@CE8# set interfaces fe-0/3/3 unit 0 vlan-id 103
user@CE8# set interfaces fe-0/3/3 unit 0 family inet address 10.31.4.2/24
user@CE8# set interfaces lo0 unit 0 family inet address 10.255.41.180/32
```

Configuring Router PE1

Step-by-Step Procedure

1. Configure the main router on Router PE1.

```
user@PE1# set interfaces fe-0/0/1 vlan-tagging
user@PE1# set interfaces fe-0/0/1 unit 3 description "main router to P0"
user@PE1# set interfaces fe-0/0/1 unit 3 vlan-id 103
user@PE1# set interfaces fe-0/0/1 unit 3 family inet address 10.31.2.1/24
user@PE1# set interfaces fe-0/0/1 unit 3 family iso
user@PE1# set interfaces fe-0/0/1 unit 3 family mpls
user@PE1# set interfaces fe-0/1/0 vlan-tagging
user@PE1# set interfaces fe-0/1/0 encapsulation vlan-vpls
user@PE1# set interfaces fe-0/1/1 vlan-tagging
user@PE1# set interfaces fe-0/1/2 vlan-tagging
user@PE1# set interfaces fe-0/1/3 vlan-tagging
user@PE1# set interfaces fe-0/1/3 unit 0 description "main router to CE4"
user@PE1# set interfaces fe-0/1/3 unit 0 vlan-id 103
user@PE1# set interfaces fe-0/1/3 unit 0 family inet address 10.31.1.2/24
```

```
user@PE1# set interfaces lo0 unit 0 description "main router loopback"
user@PE1# set interfaces lo0 unit 0 family inet address 10.255.41.173/32
user@PE1# set protocols bgp group to_main_ls type internal
user@PE1# set protocols bgp group to_main_ls local-address 10.255.41.173
user@PE1# set protocols bgp group to_main_ls export export_address
user@PE1# set protocols bgp group to_main_ls neighbor 10.255.41.179
user@PE1# set protocols bgp group to_main_ls neighbor 10.255.41.175
user@PE1# set protocols ospf area 0.0.0.0 interface lo0.0
user@PE1# set protocols ospf area 0.0.0.0 interface fe-0/0/1.3
user@PE1# set routing-options static route 10.255.41.177/32 next-hop 10.31.1.1
user@PE1# set routing-options autonomous-system 500
user@PE1# set policy-options policy-statement export_address from route-filter
10.255.41.177/32 exact
user@PE1# set policy-options policy-statement export_address then accept
user@PE1# set system login class LS1-admin logical-system LS1
user@PE1# set system login class LS1-admin permissions all
user@PE1# set system login user LS1-admin class LS1-admin
user@PE1# set system login user LS1-admin authentication plain-text-password
New password:
Retype new password:
```

2. Configure Logical System LS1 on Router PE1.

```
user@PE1# set logical-systems LS1 interfaces fe-0/0/1 unit 1 description "LS1 interface"
user@PE1# set logical-systems LS1 interfaces fe-0/0/1 unit 1 vlan-id 101
user@PE1# set logical-systems LS1 interfaces fe-0/0/1 unit 1 family inet address 10.11.2.1/24
user@PE1# set logical-systems LS1 interfaces fe-0/0/1 unit 1 family iso
user@PE1# set logical-systems LS1 interfaces fe-0/0/1 unit 1 family mpls
user@PE1# set logical-systems LS1 interfaces fe-0/1/1 unit 0 description "routing-instance
red interface"
user@PE1# set logical-systems LS1 interfaces fe-0/1/1 unit 0 vlan-id 101
user@PE1# set logical-systems LS1 interfaces fe-0/1/1 unit 0 family inet address 10.11.1.2/24
user@PE1# set logical-systems LS1 interfaces fe-0/1/2 unit 0 description "routing-instance
blue interface"
user@PE1# set logical-systems LS1 interfaces fe-0/1/2 unit 0 vlan-id 102
user@PE1# set logical-systems LS1 interfaces fe-0/1/2 unit 0 family inet address 10.21.1.2/24
user@PE1# set logical-systems LS1 interfaces lo0 unit 1 description "LS1 loopback"
user@PE1# set logical-systems LS1 interfaces lo0 unit 1 family inet address 10.10.10.10/32
user@PE1# set logical-systems LS1 interfaces lo0 unit 1 family iso address
47.1111.1111.1111.1111.00
user@PE1# set logical-systems LS1 protocols rsvp interface all
user@PE1# set logical-systems LS1 protocols mpls label-switched-path to_10.10.10.12 to
```

```
10.10.10.12
user@PE1# set logical-systems LS1 protocols mpls interface all
user@PE1# set logical-systems LS1 protocols bgp group to_other_PE type internal
user@PE1# set logical-systems LS1 protocols bgp group to_other_PE local-address 10.10.10.10
user@PE1# set logical-systems LS1 protocols bgp group to_other_PE family inet-vpn any
user@PE1# set logical-systems LS1 protocols bgp group to_other_PE neighbor 10.10.10.12
user@PE1# set logical-systems LS1 protocols isis interface all
user@PE1# set logical-systems LS1 policy-options policy-statement from_bgp_to_ospf then accept
user@PE1# set logical-systems LS1 routing-instances blue instance-type vrf
user@PE1# set logical-systems LS1 routing-instances blue interface fe-0/1/2.0
user@PE1# set logical-systems LS1 routing-instances blue route-distinguisher 10.10.10.10:200
user@PE1# set logical-systems LS1 routing-instances blue vrf-target target:20:20
user@PE1# set logical-systems LS1 routing-instances blue protocols bgp group to_CE type
external
user@PE1# set logical-systems LS1 routing-instances blue protocols bgp group to_CE local-
address 10.21.1.2
user@PE1# set logical-systems LS1 routing-instances blue protocols bgp group to_CE peer-as 200
user@PE1# set logical-systems LS1 routing-instances blue protocols bgp group to_CE neighbor
10.21.1.1
user@PE1# set logical-systems LS1 routing-instances red instance-type vrf
user@PE1# set logical-systems LS1 routing-instances red interface fe-0/1/1.0
user@PE1# set logical-systems LS1 routing-instances red route-distinguisher 10.10.10.10:100
user@PE1# set logical-systems LS1 routing-instances red vrf-target target:10:10
user@PE1# set logical-systems LS1 routing-instances red protocols ospf export from_bgp_to_ospf
user@PE1# set logical-systems LS1 routing-instances red protocols ospf area 0.0.0.0 interface
all
user@PE1# set logical-systems LS1 routing-options autonomous-system 100
```

3. Configure Logical System LS2 on Router PE1.

```
user@PE1# set logical-systems LS2 interfaces fe-0/0/1 unit 0 description "core-facing LS2
interface"
user@PE1# set logical-systems LS2 interfaces fe-0/0/1 unit 0 vlan-id 100
user@PE1# set logical-systems LS2 interfaces fe-0/0/1 unit 0 family inet address 10.1.2.1/24
user@PE1# set logical-systems LS2 interfaces fe-0/0/1 unit 0 family iso
user@PE1# set logical-systems LS2 interfaces fe-0/0/1 unit 0 family mpls
user@PE1# set logical-systems LS2 interfaces fe-0/1/0 unit 0 description "vpls interface to
ce3"
user@PE1# set logical-systems LS2 interfaces fe-0/1/0 unit 0 encapsulation vlan-vpls
user@PE1# set logical-systems LS2 interfaces fe-0/1/0 unit 0 vlan-id 600
user@PE1# set logical-systems LS2 interfaces fe-0/1/0 unit 0 family vpls
user@PE1# set logical-systems LS2 interfaces lo0 unit 2 description "LS2 loopback"
```

```
user@PE1# set logical-systems LS2 interfaces lo0 unit 2 family inet address 10.20.20.20/32
user@PE1# set logical-systems LS2 interfaces lo0 unit 2 family iso address
47.2222.2222.2222.2222.00
user@PE1# set logical-systems LS2 protocols rsvp interface all
user@PE1# set logical-systems LS2 protocols mpls label-switched-path to_10.20.20.22 to
10.20.20.22
user@PE1# set logical-systems LS2 protocols mpls interface all
user@PE1# set logical-systems LS2 protocols bgp group to_VPLS_PE type internal
user@PE1# set logical-systems LS2 protocols bgp group to_VPLS_PE local-address 10.20.20.20
user@PE1# set logical-systems LS2 protocols bgp group to_VPLS_PE family l2vpn signaling
user@PE1# set logical-systems LS2 protocols bgp group to_VPLS_PE neighbor 10.20.20.22
user@PE1# set logical-systems LS2 protocols isis interface fe-0/0/1.0
user@PE1# set logical-systems LS2 protocols isis interface lo0.2
user@PE1# set logical-systems LS2 routing-instances new instance-type vpls
user@PE1# set logical-systems LS2 routing-instances new interface fe-0/1/0.0
user@PE1# set logical-systems LS2 routing-instances new route-distinguisher 10.20.20.20:100
user@PE1# set logical-systems LS2 routing-instances new vrf-target target:30:30
user@PE1# set logical-systems LS2 routing-instances new protocols vpls site-range 10
user@PE1# set logical-systems LS2 routing-instances new protocols vpls site newPE site-
identifier 1
user@PE1# set logical-systems LS2 routing-options autonomous-system 400
```

Configuring Router PE2

Step-by-Step Procedure

1. Configure the main router on Router PE2.

```
user@PE2# set interfaces fe-0/2/0 vlan-tagging
user@PE2# set interfaces fe-0/2/0 encapsulation vlan-vpls
user@PE2# set interfaces fe-0/2/1 vlan-tagging
user@PE2# set interfaces fe-0/2/2 vlan-tagging
user@PE2# set interfaces fe-0/2/3 vlan-tagging
user@PE2# set interfaces fe-0/2/3 unit 0 description "main router to CE8"
user@PE2# set interfaces fe-0/2/3 unit 0 vlan-id 103
user@PE2# set interfaces fe-0/2/3 unit 0 family inet address 10.31.4.1/24
user@PE2# set interfaces so-1/2/0 encapsulation frame-relay
user@PE2# set interfaces so-1/2/0 unit 3 description "main router to P0"
user@PE2# set interfaces so-1/2/0 unit 3 dlci 103
user@PE2# set interfaces so-1/2/0 unit 3 family inet address 10.31.3.2/24
user@PE2# set interfaces so-1/2/0 unit 3 family iso
```

```
user@PE2# set interfaces so-1/2/0 unit 3 family mpls
user@PE2# set interfaces lo0 unit 0 description "main router loopback"
user@PE2# set interfaces lo0 unit 0 family inet address 10.155.41.179/32
user@PE2# set protocols bgp group to_main_ls type internal
user@PE2# set protocols bgp group to_main_ls local-address 10.255.41.179
user@PE2# set protocols bgp group to_main_ls export export_address
user@PE2# set protocols bgp group to_main_ls neighbor 10.255.41.173
user@PE2# set protocols bgp group to_main_ls neighbor 10.255.41.175
user@PE2# set protocols ospf area 0.0.0.0 interface so-1/2/0.3
user@PE2# set protocols ospf area 0.0.0.0 interface fe-0/2/3.0
user@PE2# set protocols ospf area 0.0.0.0 interface lo0.0
user@PE2# set routing-options static route 10.255.41.180/32 next-hop 10.31.4.2
user@PE2# set routing-options autonomous-system 500
user@PE2# set policy-options policy-statement export_address from route-filter
10.255.41.180/32 exact
user@PE2# set policy-options policy-statement export_address then accept
user@PE2# set system login class LS1-admin logical-system LS1
user@PE2# set system login class LS1-admin permissions all
user@PE2# set system login class LS1-onlooker logical-system LS2
user@PE2# set system login class LS1-onlooker permissions view
user@PE2# set system login user LS1-admin class LS1-admin
```

2. Configure Logical System LS1 on Router PE2.

```
user@PE2# set logical-systems LS1 interfaces fe-0/2/0 unit 1 description "routing-instance
red interface connects to Router CE5"
user@PE2# set logical-systems LS1 interfaces fe-0/2/0 unit 1 vlan-id 101
user@PE2# set logical-systems LS1 interfaces fe-0/2/0 unit 1 family inet address 10.11.4.1/24
user@PE2# set logical-systems LS1 interfaces fe-0/2/0 unit 2 description "routing-instance
blue interface connects to Router CE6"
user@PE2# set logical-systems LS1 interfaces fe-0/2/0 unit 2 vlan-id 102
user@PE2# set logical-systems LS1 interfaces fe-0/2/0 unit 2 family inet address 10.21.4.1/24
user@PE2# set logical-systems LS1 interfaces so-1/2/0 unit 1 description "core-facing LS1
interface"
user@PE2# set logical-systems LS1 interfaces so-1/2/0 unit 1 dlci 101
user@PE2# set logical-systems LS1 interfaces so-1/2/0 unit 1 family inet address 10.11.3.2/24
user@PE2# set logical-systems LS1 interfaces so-1/2/0 unit 1 family iso
user@PE2# set logical-systems LS1 interfaces so-1/2/0 unit 1 family mpls
user@PE2# set logical-systems LS1 interfaces lo0 unit 1 description "LS1 loopback"
user@PE2# set logical-systems LS1 interfaces lo0 unit 1 family inet address 10.10.10.12/32
user@PE2# set logical-systems LS1 interfaces lo0 unit 1 family iso address
47.1111.1111.1111.1113.00
```

```
user@PE2# set logical-systems LS1 protocols rsvp interface all
user@PE2# set logical-systems LS1 protocols mpls label-switched-path to_10.10.10.10 to
10.10.10.10
user@PE2# set logical-systems LS1 protocols mpls interface all
user@PE2# set logical-systems LS1 protocols bgp group to_other_PE type internal
user@PE2# set logical-systems LS1 protocols bgp group to_other_PE local-address 10.10.10.12
user@PE2# set logical-systems LS1 protocols bgp group to_other_PE family inet any
user@PE2# set logical-systems LS1 protocols bgp group to_other_PE family inet-vpn any
user@PE2# set logical-systems LS1 protocols bgp group to_other_PE neighbor 10.10.10.10
user@PE2# set logical-systems LS1 protocols isis interface all
user@PE2# set logical-systems LS1 policy-options policy-statement from_bgp_to_ospf then accept
user@PE2# set logical-systems LS1 routing-instances blue instance-type vrf
user@PE2# set logical-systems LS1 routing-instances blue interface fe-0/2/2.0
user@PE2# set logical-systems LS1 routing-instances blue route-distinguisher 10.10.10.12:200
user@PE2# set logical-systems LS1 routing-instances blue vrf-target target:20:20
user@PE2# set logical-systems LS1 routing-instances blue protocols bgp group to_CE local-
address 10.21.4.1
user@PE2# set logical-systems LS1 routing-instances blue protocols bgp group to_CE peer-as 300
user@PE2# set logical-systems LS1 routing-instances blue protocols bgp group to_CE neighbor
10.21.4.2
user@PE2# set logical-systems LS1 routing-instances red instance-type vrf
user@PE2# set logical-systems LS1 routing-instances red interface fe-0/2/1.0
user@PE2# set logical-systems LS1 routing-instances red route-distinguisher 10.10.10.12:100
user@PE2# set logical-systems LS1 routing-instances red vrf-target target:10:10
user@PE2# set logical-systems LS1 routing-instances red protocols ospf export from_bgp_to_ospf
user@PE2# set logical-systems LS1 routing-instances red protocols ospf area 0.0.0.0 interface
all
user@PE2# set logical-systems LS1 routing-options autonomous-system 100
```

3. Configure Logical System LS2 on Router PE2.

```
user@PE2# set logical-systems LS2 interfaces fe-0/2/0 unit 0 description "vpls interface
connects to Router CE7"
user@PE2# set logical-systems LS2 interfaces fe-0/2/0 unit 0 encapsulation vlan-vpls
user@PE2# set logical-systems LS2 interfaces fe-0/2/0 unit 0 vlan-id 600
user@PE2# set logical-systems LS2 interfaces fe-0/2/0 unit 0 family vpls
user@PE2# set logical-systems LS2 interfaces so-1/2/0 unit 0 description "core-facing LS2
interface"
user@PE2# set logical-systems LS2 interfaces so-1/2/0 unit 0 dlci 100
user@PE2# set logical-systems LS2 interfaces so-1/2/0 unit 0 family inet address 10.1.3.2/24
user@PE2# set logical-systems LS2 interfaces so-1/2/0 unit 0 family iso
user@PE2# set logical-systems LS2 interfaces so-1/2/0 unit 0 family mpls
```

```
user@PE2# set logical-systems LS2 interfaces lo0 unit 2 description "LS2 loopback"
user@PE2# set logical-systems LS2 interfaces lo0 unit 2 family inet address 10.20.20.22/32
user@PE2# set logical-systems LS2 interfaces lo0 unit 2 family iso address
47.2222.2222.2222.2224.00
user@PE2# set logical-systems LS2 protocols rsvp interface all
user@PE2# set logical-systems LS2 protocols mpls label-switched-path to_10.20.20.20 to
10.20.20.20
user@PE2# set logical-systems LS2 protocols mpls interface all
user@PE2# set logical-systems LS2 protocols bgp group to_VPLS_PE type internal
user@PE2# set logical-systems LS2 protocols bgp group to_VPLS_PE local-address 10.20.20.22
user@PE2# set logical-systems LS2 protocols bgp group to_VPLS_PE family l2vpn signaling
user@PE2# set logical-systems LS2 protocols bgp group to_VPLS_PE neighbor 10.20.20.20
user@PE2# set logical-systems LS2 protocols isis interface so-1/2/0.0
user@PE2# set logical-systems LS2 protocols isis interface lo0.2
user@PE2# set logical-systems LS2 routing-instances new instance-type vpls
user@PE2# set logical-systems LS2 routing-instances new interface fe-0/2/0.0
user@PE2# set logical-systems LS2 routing-instances new route-distinguisher 10.20.20.22:100
user@PE2# set logical-systems LS2 routing-instances new vrf-target target:30:30
user@PE2# set logical-systems LS2 routing-instances new protocols vpls site-range 10
user@PE2# set logical-systems LS2 routing-instances new protocols vpls site newPE site-
identifier 2
user@PE2# set logical-systems LS2 routing-options autonomous-system 400
```

Configuring Router P0

Step-by-Step Procedure

1. Configure the main router on Router P0.

```
user@P0# set interfaces fe-1/1/3 vlan-tagging
user@P0# set interfaces fe-1/1/3 unit 3 description "connects to the main router on pe1"
user@P0# set interfaces fe-1/1/3 unit 3 vlan-id 103
user@P0# set interfaces fe-1/1/3 unit 3 family inet address 10.31.2.2/24
user@P0# set interfaces fe-1/1/3 unit 3 family iso
user@P0# set interfaces fe-1/1/3 unit 3 family mpls
user@P0# set interfaces so-1/2/0 dce
user@P0# set interfaces so-1/2/0 encapsulation frame-relay
user@P0# set interfaces so-1/2/0 unit 3 description "connects to the main router on pe2"
user@P0# set interfaces so-1/2/0 unit 3 dlci 103
user@P0# set interfaces so-1/2/0 unit 3 family inet address 10.31.3.1/24
user@P0# set interfaces so-1/2/0 unit 3 family iso
```

```
user@P0# set interfaces so-1/2/0 unit 3 family mpls
user@P0# set interfaces lo0 unit 0 description "main router loopback"
user@P0# set interfaces lo0 unit 0 family inet address 10.255.41.175/32
user@P0# set routing-options autonomous-system 500
user@P0# set protocols bgp group to_main_ls type internal
user@P0# set protocols bgp group to_main_ls local-address 10.255.41.175
user@P0# set protocols bgp group to_main_ls neighbor 10.255.41.179
user@P0# set protocols bgp group to_main_ls neighbor 10.255.41.173
user@P0# set protocols ospf area 0.0.0.0 interface lo0.0
user@P0# set protocols ospf area 0.0.0.0 interface fe-1/1/3.3
user@P0# set protocols ospf area 0.0.0.0 interface so-1/2/0.3
user@P0# set system login class LS1-admin logical-system LS1
user@P0# set system login class LS1-admin permissions all
user@P0# set system login class LS1-onlooker logical-system LS2
user@P0# set system login class LS1-onlooker permissions view
user@P0# set system login user LS1-admin class LS1-admin
```

2. Configure Logical System LS1 on Router P0.

```
user@P0# set logical-systems LS1 interfaces fe-1/1/3 unit 1 description "LS1 interface
connects to LS1 on pe1"
user@P0# set logical-systems LS1 interfaces fe-1/1/3 unit 1 vlan-id 101
user@P0# set logical-systems LS1 interfaces fe-1/1/3 unit 1 family inet address 10.11.2.2/24
user@P0# set logical-systems LS1 interfaces fe-1/1/3 unit 1 family iso
user@P0# set logical-systems LS1 interfaces fe-1/1/3 unit 1 family mpls
user@P0# set logical-systems LS1 interfaces so-1/2/0 unit 1 description "LS1 interface
connects to LS1 on pe2"
user@P0# set logical-systems LS1 interfaces so-1/2/0 unit 1 dlci 101
user@P0# set logical-systems LS1 interfaces so-1/2/0 unit 1 family inet address 10.11.3.1/24
user@P0# set logical-systems LS1 interfaces so-1/2/0 unit 1 family iso
user@P0# set logical-systems LS1 interfaces so-1/2/0 unit 1 family mpls
user@P0# set logical-systems LS1 interfaces lo0 unit 1 description "LS1 loopback"
user@P0# set logical-systems LS1 interfaces lo0 unit 1 family inet address 10.10.10.11/32
user@P0# set logical-systems LS1 interfaces lo0 unit 1 family iso address
47.1111.1111.1111.1112.00
user@P0# set logical-systems LS1 protocols rsvp interface all
user@P0# set logical-systems LS1 protocols mpls interface all
user@P0# set logical-systems LS1 protocols isis interface all
```

3. Configure Logical System LS2 on Router P0.

```
user@P0# set logical-systems LS2 interfaces fe-1/1/3 unit 0 description "LS2 interface
connects to LS2 on pe1"
user@P0# set logical-systems LS2 interfaces fe-1/1/3 unit 0 vlan-id 100
user@P0# set logical-systems LS2 interfaces fe-1/1/3 unit 0 family inet address 10.1.2.2/24
user@P0# set logical-systems LS2 interfaces fe-1/1/3 unit 0 family iso
user@P0# set logical-systems LS2 interfaces fe-1/1/3 unit 0 family mpls
user@P0# set logical-systems LS2 interfaces so-1/2/0 unit 0 description "LS2 interface
connects to LS2 on pe2"
user@P0# set logical-systems LS2 interfaces so-1/2/0 unit 0 dlci 100
user@P0# set logical-systems LS2 interfaces so-1/2/0 unit 0 family inet address 10.1.3.1/24
user@P0# set logical-systems LS2 interfaces so-1/2/0 unit 0 family iso
user@P0# set logical-systems LS2 interfaces so-1/2/0 unit 0 family mpls
user@P0# set logical-systems LS2 interfaces lo0 unit 2 description "LS2 loopback"
user@P0# set logical-systems LS2 interfaces lo0 unit 2 family inet address 10.20.20.21/32
user@P0# set logical-systems LS2 interfaces lo0 unit 2 family iso address
47.2222.2222.2222.2223.00
user@P0# set logical-systems LS2 protocols rsvp interface all
user@P0# set logical-systems LS2 protocols mpls interface all
user@P0# set logical-systems LS2 protocols isis interface fe-1/1/3.0
user@P0# set logical-systems LS2 protocols isis interface so-1/2/0.0
user@P0# set logical-systems LS2 protocols isis interface lo0.2
```

Results

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 an administrator for Logical System 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 administrator for Logical System LS1 that you configured on Router PE1. Configure this same administrator for Logical System 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. Logical System 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 Logical System LS1 and viewing privileges for Logical System 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;
        }
    }
}
```

Verification

IN THIS SECTION

- Router CE1 Status | **121**
- Router CE2 Status | **122**
- Router CE3 Status | **122**
- Router PE1 Status: Main Router | **123**
- Router PE1 Status: Logical System LS1 | **124**
- Router PE1 Status: Logical System LS2 | **126**
- Router P0 Status: Main Router | **127**
- Router P0 Status: Main Router | **127**
- Router P0 Status: Logical System LS1 | **128**
- Router P0 Status: Logical System LS2 | **129**
- Router PE2 Status: Main Router | **130**
- Router PE2 Status: Logical System LS1 | **132**
- Router PE2 Status: Logical System LS2 | **134**
- Router CE5 Status | **136**
- Router CE6 Status | **136**
- Router CE7 Status | **137**
- Logical System Administrator Verification Output | **137**

Confirm that the configuration is working properly by running these 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 | ospf3) neighbor** (**logical-system** *logical-system-name*)
- **show route** (**logical-system** *logical-system-name*)
- **show route protocol** (**logical-system** *logical-system-name*)
- **show rsvp session** (**logical-system** *logical-system-name*)

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

Router CE1 Status

Purpose

Verify connectivity.

Action

```
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

Purpose

Verify connectivity.

Action

```
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

Purpose

Verify connectivity.

Action

```
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

Purpose

Verify BGP operation.

Action

```
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: Logical System LS1

Purpose

Verify BGP operation.

Action

```
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
```

Red VPN

The primary administrator or logical system 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
```

Blue VPN

The primary administrator or logical system 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
```

Router PE1 Status: Logical System LS2

Purpose

Verify VPLS operation.

Action

```
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

Purpose

Verify connectivity.

Action

```
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: Main Router

Purpose

Verify routing protocols operation.

Action

```
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: Logical System LS1

Purpose

Verify routing protocols operation.

Action

```
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: Logical System LS2

Purpose

Verify routing protocols operation.

Action

```
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

Purpose

Verify routing protocols operation.

Action

```
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: Logical System LS1

Purpose

Verify routing protocols operation.

Action

```
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: Logical System LS2

Purpose

Verify routing protocols operation.

Action

```
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

Purpose

Verify connectivity.

Action

```
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

Purpose

Verify connectivity.

Action

```
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

Purpose

Verify connectivity.

Action

```
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

Purpose

Because logical system administrators only have access to the configuration information of the logical systems to which they are assigned, the verification output is limited to these logical systems as well. The following output shows what the logical system administrator **LS1-admin** in this example configuration would see.

To verify that each pair of CE routers has end-to-end connectivity, issue the `ping` command on Routers CE1, CE2, and CE3:

Action

From CE1, ping CE5 (the Red VPN).

From CE2, ping CE6 (the Blue VPN).

From CE3, ping CE7 (the VPLS).

```
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
```

```
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
```

```
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
```

SEE ALSO

Junos OS VPNs Library for Routing Devices

RELATED DOCUMENTATION

Layer 2 Learning and Forwarding on Logical Systems | 71

Virtualized Data Center Using Logical Systems

IN THIS SECTION

- Two-Tiered Virtualized Data Center Solution for Large Enterprise Networks | 139
- Requirements of a Two-Tiered Virtualized Data Center for Large Enterprise Networks | 142
- Example: Configuring a Two-Tiered Virtualized Data Center for Large Enterprise Networks | 145

Logical Systems enables VDC design uses virtualization technologies such as virtual LANs (VLANs), virtual routers, virtual route forwarders, inter-virtual route forwarding to provide flexible traffic isolation. For more information, see the following topics:

Two-Tiered Virtualized Data Center Solution for Large Enterprise Networks

IN THIS SECTION

- Network Traffic Segmentation | 140
- Flexibility | 140
- Security | 140
- Access and Availability | 141

Cost-Effective Incremental Scaling | 141
Orchestration and Automation | 142

The following describes a Juniper Networks two-tiered, high-speed, multiservice virtualized data center (VDC). A two-tiered architecture meets the low latency requirements of a virtualized server environment and supports the overlying security mandate to maintain controlled segmentation between various business units.

Network Traffic Segmentation

Juniper Networks VDC design uses virtualization technologies such as virtual LANs (VLANs), virtual routers, virtual route forwarders, inter-virtual route forwarding, and logical systems to provide flexible traffic isolation.

A fully redundant two-tiered data center design consists of Juniper Networks EX Series Ethernet Switches at the access layer for server connectivity, MX Series 5G Universal Routing Platforms as a collapsed LAN aggregation/core layer, and clustered SRX Series Services Gateways to provide firewall security services across the data center trust boundaries.

Flexibility

The Juniper Networks VDC design uses 802.1Q VLANs, MPLS, BGP, Virtual Router Redundancy Protocol (VRRP), Traffic Engineering, and Fast Reroute to provide design flexibility while maintaining a standards-based approach. The design can also support a virtual private LAN service (VPLS).

Security

The Juniper Networks VDC design uses security zones to implement the policy enforcement points. The SRX cluster is responsible for all stateful packet inspection for traffic that crosses business unit trust boundaries as well as all ingress and egress traffic for the data center.

The Juniper Networks Junos operating system is configured with different administrator accounts for each logical system that supports confined access to network resources and can be customized for individual business units.

Access and Availability

In the Juniper Networks VDC design, described in "Example: Configuring a Two-Tiered Virtualized Data Center for Large Enterprise Networks" on page 145, top-of-rack (TOR) EX Series switches provide access to the servers and provide redundancy.

All uplinks from the TOR switches are 802.1Q trunk links that are terminated directly into each of the MX Series devices that make up the Point of Delivery (POD) at the aggregation/core layer.

A VRRP instance is defined on each VLAN within the MX Series device to act as the default router for all server hosts in a given VLAN. To allow for VRRP to work properly, each bridge domain is extended between each MX Series device through an interconnection link. The MX Series device uses an integrated routing and bridging (IRB) interface as the Layer 3 interface for each bridge domain, with VRRP configured for redundancy.

A pair of 802.3ad aggregated Ethernet bundles are used between the MX Series devices. Each MX Series device is divided into a number of Logical Systems. Logical systems in the MX Series device are used to define logical trust boundaries within the data center itself and between respective business units.

A clustered pair of SRX Series devices acting as firewalls provide security services across the data center trust boundaries. Virtual routers on the SRX Series devices act as customer edge (CE) routers for each business unit.

A single redundancy group for the data plane is defined on the SRX Series Services Gateways with two redundant Ethernet interfaces as member interfaces. This redundancy group handles the data plane failover of the SRX Series firewall and is configured such that any loss of either northbound or southbound SRX Series interfaces forces a full failover to the secondary node. This failover is essentially a Layer 1 failover, which means that it occurs quickly and does not disrupt the routing topology above it.

Cost-Effective Incremental Scaling

The Juniper Networks VDC design supports incremental scaling of the network. This allows the VDC to be created with minimum cost to meet the current need.

The access layer can be expanded by adding EX Series switches at the top of rack.

The aggregation/core layer can be expanded by adding additional MX Series devices within a given POD.

The security services can be expanded by adding 4-port 10-Gigabit Ethernet I/O cards (IOCs) and services processing cards (SPCs) in the SRX Series devices. The addition of IOCs increases the 10-Gigabit Ethernet port density. The addition of each SPC card to the chassis adds another 10 Gbps (5 Gbps Internet mix (IMIX)), 2 million sessions, and 100,000 connections per second (CPS) up to a maximum rated capacity for the platform of 150 Gbps (47.5 Gbps IMIX), 10 million sessions, and 350,000 CPS (as measured in Junos OS Release 10.2).

Orchestration and Automation

The Juniper Networks VDC design uses the Juniper Networks Junos Space management platform. Junos Space includes a portfolio of applications for scaling services, simplifying network operations, and automating support for complex network environments.

In addition, the network devices are configured to support background Secure Copy Protocol (SCP) file transfers, commit scripts, and a file archive site.

SEE ALSO

Requirements of a Two-Tiered Virtualized Data Center for Large Enterprise Networks | 142

Example: Configuring a Two-Tiered Virtualized Data Center for Large Enterprise Networks | 145

Requirements of a Two-Tiered Virtualized Data Center for Large Enterprise Networks

IN THIS SECTION

- Network Traffic Segmentation | 143
- Flexibility | 143
- Security | 143
- Access and Availability | 144
- Cost-Effective Incremental Scaling | 144
- Orchestration and Automation | 144

Large enterprises have certain specific needs for the hosting environment that the design of their data center must meet. This section describes the requirements of a company that operates as a service provider to its individual business units (BUs).

One of the primary requirements of a virtualized data center (VDC) for a large enterprise is the ability to segment the network by business unit. This includes traffic segmentation and administrative control segmentation.

Other requirements include security controls between business units, security controls between the company and the outside world, flexibility to grow and adapt the network, and a robust and cost-effective way to manage the entire network.

Network Traffic Segmentation

The requirement described here is for network resources to be isolated in several ways. Traffic must be segmented by business units. Traffic flows between network segments must be prohibited except where specifically allowed. Traffic isolation must be controlled at designated policy enforcement points. Network resources must be dedicated to a segment, but the network must have the flexibility to change the allocation of resources.

Segmented resources must be logically grouped according to policies. For example, test traffic must be isolated from production traffic. Traffic must also be isolated according to business entities, contractual requirements, legal or regulatory requirements, risk rating, and corporate standards.

The network segmentation design must not be disruptive to the business, must be integrated with the larger data center and cloud network design, must allow business units to access network resources globally, and must support new business capabilities.

Flexibility

The network design must be flexible enough to react to business and environment changes with minimal design and re-engineering efforts. The VDC design must be flexible in terms of isolating business unit workloads from other business units and general data center services and applications. The network solution must ensure that the business is minimally impacted when network and segmentation changes take place.

The VDC must be flexible enough to be implemented:

- Within a single data center
- Within a data hall
- Across two or more data centers
- Across two or more data halls within or between data centers
- Between a data center and an external cloud service provider

Security

The network design must allow business units to be isolated within the hosting environment. In the event of a network security incident, business units must be isolated from the hosting environment and other business units.

Traffic flow between business unit segments must be denied by default and must be explicitly permitted only at policy enforcement points owned and controlled by the data center service provider.

The policy enforcement point must include access control capabilities and might include threat protection capabilities.

Access and Availability

The VDC must provide access to common data center services such as computation, storage, security, traffic management, operations, and applications. The network must operate across multiple global service providers and must deliver optimal, predictable, and consistent performance across the network. The VDC must be implemented across data center business units.

The network solution must meet business unit availability requirements as defined in service-level agreements.

Cost-Effective Incremental Scaling

The VDC design must be cost effective for the business to run and must enable new business capabilities. It must be possible to implement the network solution in an incremental manner with minimal impact to the business.

Orchestration and Automation

The VDC design must include a management system that supports automation for provisioning, availability and workload monitoring, and reporting. Workload and availability reports must be available by business unit.

SEE ALSO

Two-Tiered Virtualized Data Center Solution for Large Enterprise Networks | **139**

Example: Configuring a Two-Tiered Virtualized Data Center for Large Enterprise Networks | **145**

Example: Configuring a Two-Tiered Virtualized Data Center for Large Enterprise Networks

IN THIS SECTION

- Requirements | 145
- Configuring a Two-Tiered Virtualized Data Center Overview | 145
- Configuring the Access Layer | 148
- Configuring the Aggregation Layer in the Trusted Logical Systems | 154
- Configuring the Core Layer in the Untrusted Logical Systems | 164
- Configuring the Security Device | 171

This example provides a step-by-step procedure for configuring a two-tiered virtualized data center for large enterprise networks.

Requirements

This example uses the following hardware and software components:

- Two MX Series 5G Universal Routing Platforms running Junos OS Release 10.2 or later
- Six EX Series Ethernet Switches running Junos OS Release 10.2 or later
- Two SRX Series Services Gateways running Junos OS Release 10.4 or later

Configuring a Two-Tiered Virtualized Data Center Overview

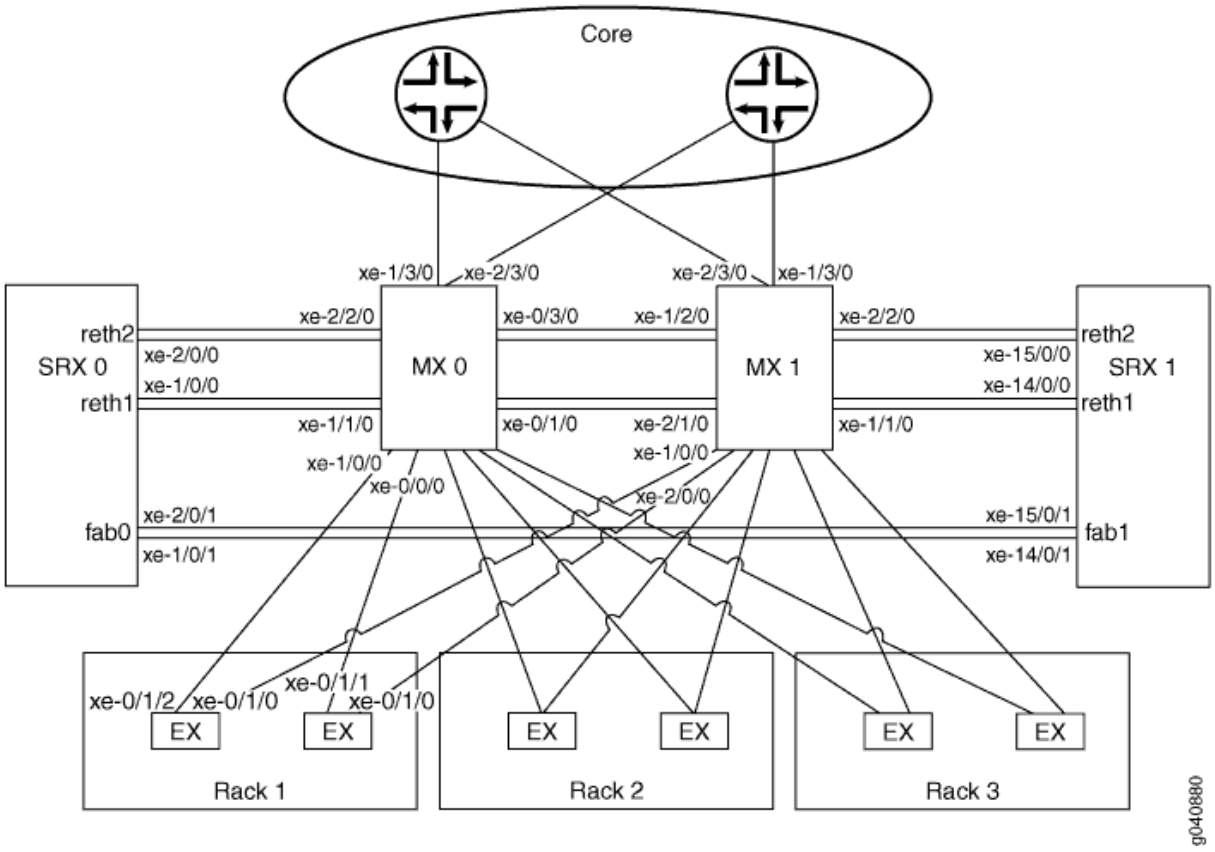
This example provides a step-by-step procedure for configuring a two-tiered virtualized data center for large enterprises. The steps in the example follow the data path from an interface connected to a server in BU2 using VLAN 17, to Logical System Trust1, through Virtual Router MX-VR2, through Virtual Router SRX-VR2, through VRF2 in the Logical System Untrust, and out to the core network.

The core network in this example simultaneously supports IP-based routing and MPLS-based label switching. The virtual routers on the SRX Series device perform the functions of customer edge (CE) routers. The VPN routing and forwarding (VRF) routing instances on the MX Series devices perform the functions of service provider edge (PE) routers. The OSPF protocol serves as the interior gateway protocol to carry routes to the PE router loopback addresses that are used as the BGP next-hop address for the IP-based and MPLS-based networks supported by this example.

NOTE: The steps in this example are representative of the entire network configuration. The example does not show every step for every virtual device.

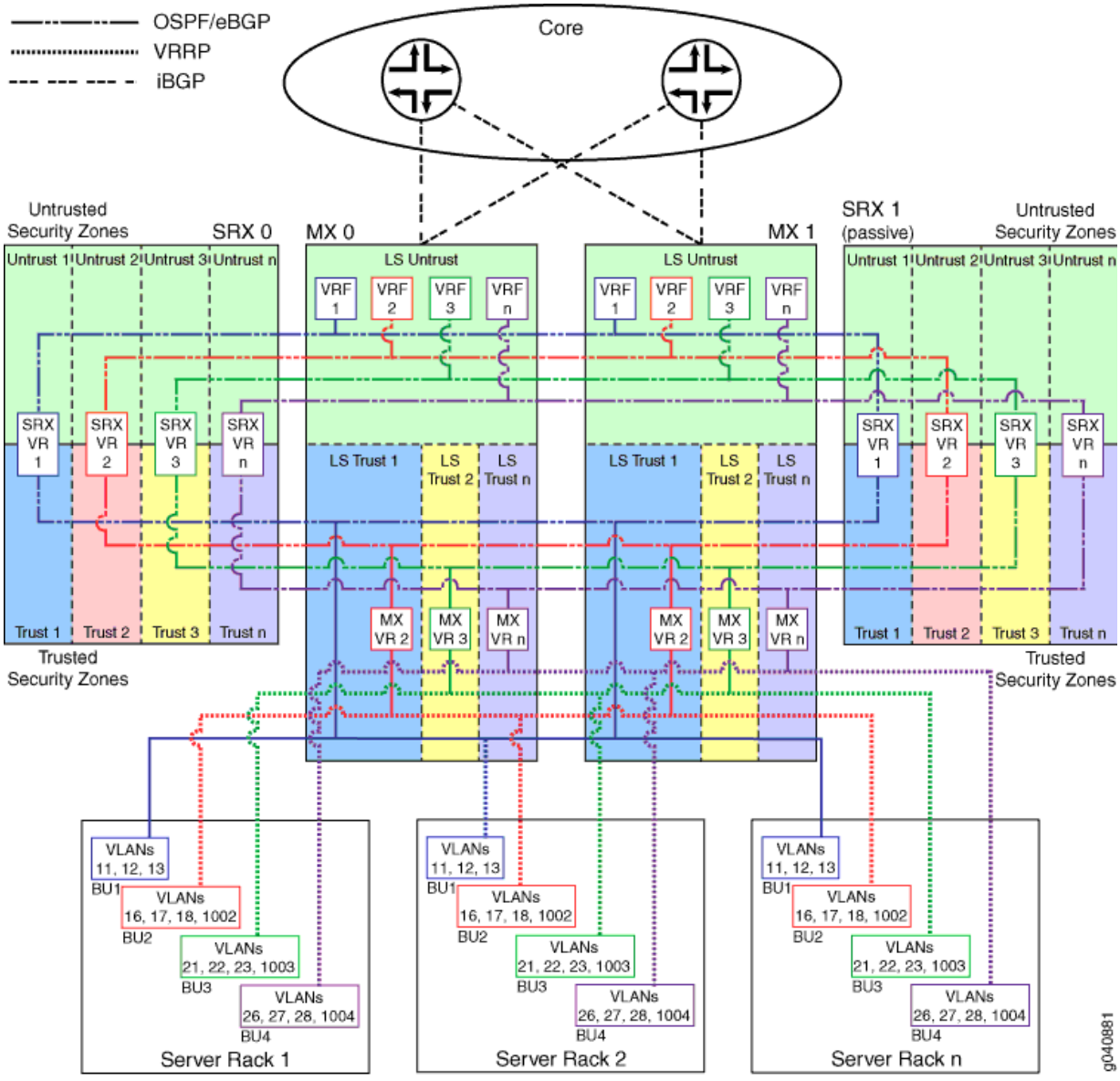
The physical connections used in this example are shown in Figure 13 on page 146.

Figure 13: Virtualized Data Center Physical Topology



The logical connections used in this example are shown in Figure 14 on page 147.

Figure 14: Virtualized Data Center Logical Topology



In the logical topology illustration:

- Users access the data center across the enterprise core network shown at the top.
- Virtual routers configured in Logical System Untrust on the MX Series devices forward the traffic to separate virtual routers configured in the Untrusted security zone on the SRX Series devices. These virtual routers act as edge routers for the various business units.

- Virtual routers configured on the active SRX Series device forward the traffic to the Trusted security zones.
- Virtual routers configured in separate logical systems on the MX Series devices forward the traffic to a bridge domain of VLANs configured on the EX Series devices.
- Business unit 1 requires additional separation. In this case, the virtual router (VR) configured on the SRX Series device forwards the traffic directly to the bridge domain on the EX Series devices.
- The EX Series devices switch the traffic to the data center server.
- The SRX Series devices apply security policy to all traffic traversing the untrust to trust boundary and all traffic forwarded between logical systems.
- The SRX Series devices are configured in an active/passive cluster so that only one node in the cluster is active on the data forwarding plane at a time.
- The SRX Series devices are configured with a single redundancy group for the data plane. The redundancy group uses two Ethernet interfaces (`reth1` and `reth2` in Figure 13 on page 146) as member interfaces.

Configuring the Access Layer

IN THIS SECTION

- Configuring Interfaces | **148**
- Configuring VLANs in the Access Layer | **150**
- Configuring a Redundant Trunk Group and Disabling the Spanning Tree Protocol for the Trunk Interfaces | **151**
- Configuring Management Automation | **152**

Configure the access layer by doing the following:

Configuring Interfaces

Step-by-Step Procedure

This procedure explains how to configure the physical, logical, and network management interfaces for the access layer devices. This procedure shows a representative sample of the configuration. The example does not show the configuration for every interface.

1. Configure the access layer server-facing 10-Gigabit Ethernet interfaces.

   This example configures the `ge-0/0/17` interface with VLAN ID `17`.

   Include the `member` statement and specify VLAN ID `17` at the `[edit interfaces ge-0/0/17 unit 0 family ethernet-switching vlan]` hierarchy level.

   ```
   [edit interfaces ge-0/0/17 unit 0]
   user@ex# set family ethernet-switching vlan members 17
   ```

   Repeat this step for every server-facing interface by using the appropriate interface name and VLAN number.

2. Configure the 10-Gigabit Ethernet trunk interfaces from the EX Series device to the two MX Series devices.

   This example configures the `xe-0/1/2` and `xe-0/1/0` interfaces.

   Include the `port-mode` statement and specify the `trunk` option at the `[edit interfaces xe-0/1/2 unit 0 family ethernet-switching]` and `[edit interfaces xe-0/1/0 unit 0 family ethernet-switching]` hierarchy levels.

   Include the `members` statement and specify the `all` option at the `[edit interfaces xe-0/1/2 unit 0 family ethernet-switching vlan]` and `[edit interfaces xe-0/1/0 unit 0 family ethernet-switching]` hierarchy levels.

   ```
   [edit interfaces xe-0/1/2 unit 0]
   user@ex# set family ethernet-switching port-mode trunk
   user@ex# set family ethernet-switching vlan members all
   ```

   ```
   [edit interfaces xe-0/1/0 unit 0]
   user@ex# set family ethernet-switching port-mode trunk
   user@ex# set family ethernet-switching vlan members all
   ```

   Repeat this step for every 10-Gigabit Ethernet trunk interface by using the appropriate interface name.

3. Enable the IPv4 address family for the loopback logical interface.

Include the `family` statement and specify the `inet` option to enable IPv4 at the `[edit interfaces lo0 unit 0]` hierarchy level.

```
[edit interfaces lo0 unit 0]
user@ex# set family inet
```

Repeat this step for every EX Series device by using the appropriate address for that device.

4. Configure the EX Series device management Ethernet interface.

   This example configures the `unit 0` logical interface.

   Include the `family` statement and specify the `inet` option at the `[edit me0 unit 0]` hierarchy level.

   Include the `address` statement and specify `10.8.108.19/24` as the IPv4 address at the `[edit interfaces me0 unit 0 family inet]` hierarchy level.

```
[edit interfaces me0 unit 0]
user@ex# set family inet address 10.8.108.19/24
```

   Repeat this step for every EX Series device by using the appropriate management interface address for that device.

Configuring VLANs in the Access Layer

Step-by-Step Procedure

This procedure explains how to configure the VLAN names and tag IDs and associate trunk interfaces with one of the access layer devices. This procedure shows a representative sample of the configuration. The example does not show the configuration for every VLAN.

1. Configure the VLAN name and tag ID (number) for each VLAN on the EX Series device.

   This example configures a VLAN with the name `vlan17` and tag ID `17`.

   Include the `vlan-id` statement and specify `17` as the VLAN tag ID at the `[edit vlans vlan17]` hierarchy level.

```
[edit vlans vlan17]
user@ex# set vlan-id 17
```

   Repeat this step for every VLAN on each EX Series device by using the appropriate VLAN names and tag IDs.

2. Associate the logical trunk interfaces with each VLAN on the EX Series device.

   This example associates logical interfaces `xe-0/1/0.0` and `xe-0/1/2.0` with `vlan17`.

   Include the `interface` statement and specify `xe-0/1/0.0` at the `[edit vlans vlan17]` hierarchy level.

   Include the `interface` statement and specify `xe-0/1/2.0` at the `[edit vlans vlan17]` hierarchy level.

```
[edit vlans vlan17]
user@ex# set interface xe-0/1/0.0
user@ex# set interface xe-0/1/2.0
```

   Repeat this step for every VLAN on each EX Series device by using the appropriate trunk interface names.

Configuring a Redundant Trunk Group and Disabling the Spanning Tree Protocol for the Trunk Interfaces

Step-by-Step Procedure

This procedure explains how to configure a redundant trunk group and disable the Rapid Spanning Tree Protocol (RSTP) on the trunk interfaces.

1. Configure the trunk interfaces as a redundant trunk group.

   This example configures the `xe-0/1/0.0` and `xe-0/1/2.0` trunk interfaces in a redundant trunk group named `rtgroup1`.

   Include the `interface` statement at the `[edit ethernet-switching-options redundant-trunk-group group rtgroup1]` hierarchy level and specify each trunk interface name.

   Include the `primary` statement at the `[edit ethernet-switching-options redundant-trunk-group group rtgroup1 xe-0/1/2.0]` hierarchy level.

```
[edit ethernet-switching-options redundant-trunk-group group rtgroup1]
user@ex# set interface xe-0/1/0.0
user@ex# set interface xe-0/1/2.0 primary
```

   Repeat this step for every redundant trunk group by using the appropriate interface names.

2. Disable RSTP on the trunk interfaces.

   On an EX Series device, RSTP is enabled by default. RSTP cannot be enabled on the same interface as routing.

This example disables RSTP on the `xe-0/1/0.0` and `xe-0/1/2.0` trunk interfaces.

Include the `disable` statement at the `[edit protocols rstp interface xe-0/1/0.0]` and `[edit protocols rstp interface xe-0/1/2.0]` hierarchy levels.

```
[edit protocols rstp]
user@ex# set interface xe-0/1/0.0 disable
user@ex# set interface xe-0/1/2.0 disable
```

Repeat this step for every core-facing trunk interface by using the appropriate interface name.

Configuring Management Automation

Step-by-Step Procedure

This procedure explains how to configure static routes to the management network, a known host to support background Secure Copy Protocol (SCP) file transfers, a commit script, and an event archive site.

1. Configure static routes so the Ethernet management interface can reach the management network.

   Include the `route` statement, and specify `10.8.0.0/16` as the IPv4 subnet address of the management network at the `[edit routing-options static]` hierarchy level.

   Include the `next-hop` statement, and specify the IPv4 host address of the next-hop router at the `[edit routing-options static route 10.8.0.0/16]` hierarchy level.

   ```
   [edit routing-options static]
   user@ex# set route 10.8.0.0/16 next-hop 10.8.108.254
   ```

   Repeat this step for every Ethernet management interface on the EX Series devices.

2. Configure an SSH known host.

   Include the `host` statement, and specify the IPv4 address and RSA host key options for trusted servers at the `[edit security ssh-known-hosts]` hierarchy level. In this example, the RSA host key is truncated to make it easier to read.

   ```
   [edit security ssh-known-hosts]
   user@ex# set host 127.0.0.1 rsa-key AAAAB3NzaC1yc2
   ```

   Repeat this step for every EX Series device.

3. Configure outbound SSH to support Juniper Message Bundle (JMB) transfers to Juniper Support Systems (JSS).

   In this example, the client ID is configured as `00187D0B670D`.

   Include the `client` statement, specify `00187D0B670D` as the client ID, and specify `10.8.7.32` as the IPv4 address at the `[edit system services outbound-ssh]` hierarchy level.

   Include the `port` statement and specify `7804` as the TCP port at the `[edit system services outbound-ssh client 00187D0B670D 10.8.7.32]` hierarchy level.

   Include the `device-id` statement and specify `FA022D` as the device ID at the `[edit system services outbound-ssh client 00187D0B670D]` hierarchy level.

   Include the `secret` statement at the `[edit system services outbound-ssh client 00187D0B670D ]` hierarchy level.

   Include the `services` statement and specify `netconf` as the available service at the `[edit system services outbound-ssh client 00187D0B670D ]` hierarchy level.

   ```
   [edit system services outbound-ssh client 00187D0B670D]
   user@ex# set 10.8.7.32 port 7804
   user@ex# set device-id FA022D
   user@ex# set secret "$ABC123"
   user@ex# set services netconf
   ```

   Repeat this step for every EX Series device.

4. Configure a commit script.

   In this example the script file name is `jais-activate-scripts.slax`.

   Include the `allow-transients` statement at the `[edit system scripts commit]` hierarchy level.

   Include the `optional` statement at the `[edit system scripts commit file jais-activate-scripts.slax]` hierarchy level.

   ```
   [edit system scripts commit]
   user@ex# set allow-transients
   user@ex# set file jais-activate-scripts.slax optional
   ```

   Repeat this step for every EX Series devices.

5. Configure an event archive site.

In this example, the archive URL is the local `/var/tmp/` directory, and the name given to the destination is `juniper-aim`.

Include the `archive-sites` statement and specify the archive URL at the `[edit event-options destinations juniper-aim]` hierarchy level.

```
[edit event-options destinations juniper-aim]
user@ex# set archive-sites "scp://admin@127.0.0.1://var/tmp" password “12345”
```

Repeat this step for every EX Series device.

Configuring the Aggregation Layer in the Trusted Logical Systems

IN THIS SECTION

- Configuring Interfaces in the Trusted Logical Systems | **154**
- Configuring VLANs in the Aggregation Layer | **157**
- Configuring the Virtual Router Routing Instance | **159**
- Configuring Management Interfaces | **161**
- Configuring Logical System Administrator Accounts | **162**
- Configuring Management Automation | **163**

Configure the aggregation layer by doing the following:

Configuring Interfaces in the Trusted Logical Systems

Step-by-Step Procedure

This procedure explains how to configure the physical, logical, and Layer 3 routing interfaces for the logical system in the trusted security zone of the aggregation layer. This procedure shows a representative sample of the configuration. The example does not show the configuration for every interface.

1. Enable flexible VLAN tagging on the physical interfaces.

   This example configures physical interface `xe-1/0/0`.

   Include the `encapsulation` statement and specify the `flexible-ethernet-services` option at the `[edit interfaces xe-1/0/0]` hierarchy level.

Include the `flexible-vlan-tagging` statement at the `[edit interfaces xe-1/0/0]` hierarchy level.

```
[edit interfaces xe-1/0/0]
user@mx# set encapsulation flexible-ethernet-services
user@mx# set flexible-vlan-tagging
```

Repeat this step for every physical interface connected to the EX series, SRX Series, and MX Series devices using the appropriate interface name.

2. Configure the 10-Gigabit Ethernet interfaces connected to the EX Series access layer device.

   This example configures logical interface `17`on the `xe-1/0/0` interface under the logical system named `Trust1`.

   Include the `encapsulation` statement and specify the `vlan-bridge` option at the `[edit logical-systems Trust1 interfaces xe-1/0/0 unit 17]` hierarchy level.

   Include the `vlan-id` statement and specify `17` as the VLAN ID at the `[edit logical-systems Trust1 interfaces xe-1/0/0 unit 17]` hierarchy level.

   ```
   [edit logical-systems Trust1 interfaces xe-1/0/0 unit 17]
   user@mx# set encapsulation vlan-bridge
   user@mx# set vlan-id 17
   ```

   Repeat this step for every interface connected to the access layer devices by using the appropriate interface name, logical interface number, VLAN ID, and logical system name.

3. Configure the 10-Gigabit Ethernet interfaces connected to the other MX Series device shown in Figure 13 on page 146.

   This example configures logical interface `17` on the `xe-0/1/0` interface.

   Include the `encapsulation` statement and specify the `vlan-bridge` option at the `[edit logical-systems Trust1 interfaces xe-0/1/0 unit 17]` hierarchy level.

   Include the `vlan-id` statement and specify `17` as the VLAN tag ID at the `[edit logical-systems Trust1 interfaces xe-0/1/0 unit 17]` hierarchy level.

   ```
   [edit logical-systems Trust1 interfaces xe-0/1/0 unit 17]
   user@mx# set encapsulation vlan-bridge
   user@mx# set vlan-id 17
   ```

Repeat this step for every interface connected to the other MX Series device shown in Figure 13 on page 146 by using the appropriate interface name, logical interface number, VLAN ID, and logical system name.

4. Configure the 10-Gigabit Ethernet interface connected to the SRX Series device.

   This example configures logical interface `15` on the `xe-1/1/0` interface. Include the `encapsulation` statement and specify the `vlan-bridge` option at the `[edit logical-systems Trust1 interfaces xe-1/1/0 unit 15]` hierarchy level.

   Include the `vlan-id` statement and specify `15` as the VLAN tag ID at the `[edit logical-systems Trust1 interfaces xe-1/1/0 unit 15]` hierarchy level.

   ```
   [edit logical-systems Trust1 interfaces xe-1/1/0 unit 15]
   user@mx# set encapsulation vlan-bridge
   user@mx# set vlan-id 15
   ```

   Repeat this step for every interface connected to the SRX Series device by using the appropriate interface name, logical interface number, VLAN ID, and logical system name.

5. Configure the Layer 3 integrated routing and bridging (IRB) interface address.

   This example configures the `unit 17` logical interface with `10.17.2.2/24` as the IPv4 address under the logical system named `Trust1`. Include the `address` statement and specify `10.17.2.2/24` as the IPv4 address at the `[edit logical-systems Trust1 interfaces irb unit 17 family inet]` hierarchy level.

   ```
   [edit logical-systems Trust1 interfaces irb unit 17 family inet]
   user@mx# set address 10.17.2.2/24
   ```

   Repeat this step for every Layer 3 IBR by using the appropriate logical interface name and IPv4 address.

6. Configure the IRB interface to participate in Virtual Router Redundancy Protocol (VRRP).

   This example configures the `unit 17` logical interface with `17` as the VRRP group name.

   Include the `virtual-address` statement and specify `10.17.2.1` as the IPv4 address of the virtual router at the `[edit logical-systems Trust1 interfaces irb unit 17 family inet address 10.17.2.2/24 vrrp-group 17]` hierarchy level.

Include the accept-data statement at the [edit logical-systems Trust1 interfaces irb unit 17 family inet address 10.17.2.2/24 vrrp-group 17] hierarchy level so the interface will accept packets destined for the virtual IP address.

Include the priority statement and specify 200 as the router's priority at the [edit logical-systems Trust1 interfaces irb unit 17 family inet address 10.17.2.2/24 vrrp-group 17] hierarchy level.

Include the fast-interval statement and specify 200 as the interval between VRRP advertisements at the [edit logical-systems Trust1 interfaces irb unit 17 family inet address 10.17.2.2/24 vrrp-group 17] hierarchy level.

Include the preempt statement at the [edit logical-systems Trust1 interfaces irb unit 17 family inet address 10.17.2.2/24 vrrp-group 17] hierarchy level.

```
[edit logical-systems Trust1 interfaces irb unit 17 family inet address 10.17.2.2/24 vrrp-
group 17]
user@mx# set virtual-address 10.17.2.1
user@mx# set accept-data
user@mx# set priority 200
user@mx# set fast-interval 200
user@mx# set preempt
```

Repeat this step for every Layer 3 IBR interface by using the appropriate logical interface name, IPv4 address, VRRP group name, and priority.

Configuring VLANs in the Aggregation Layer

Step-by-Step Procedure

This procedure explains how to configure the VLAN names and tag IDs and associate trunk interfaces and Layer 3 routing interfaces with each VLAN. This procedure shows a representative sample of the configuration. The example does not show the configuration for every VLAN.

1. Configure the VLAN name and tag ID (number) for each VLAN on the MX Series device.

   This example configures a VLAN with the name vlan17 and tag ID 17 in the Logical System Trust1. Include the vlan-id statement and specify 17 as the VLAN ID at the [edit logical-systems Trust1 bridge-domains vlan17] hierarchy level.

```
[edit logical-systems Trust1 bridge-domains vlan17]
user@mx# set vlan-id 17
```

Repeat this step for every VLAN on each MX Series device by using the appropriate VLAN names and tag IDs.

2. Associate the logical trunk interfaces with each VLAN on the MX Series device.

   This example associates logical interface `xe-1/0/0.17` that is connected to the EX Series device and logical interface `xe-0/1/0.17` that is connected to the other MX Series device with `vlan17`.

   Include the `interface` statement and specify `xe-1/0/0.17` at the `[edit logical-systems Trust1 bridge-domains vlan17]` hierarchy level.

   Include the `interface` statement and specify `xe-0/1/0.17` at the `[edit logical-systems Trust1 bridge-domains vlan17]` hierarchy level.

   ```
   [edit logical-systems Trust1 bridge-domains vlan17]
   user@mx# set interface xe-1/0/0.17
   user@mx# set interface xe-0/1/0.17
   ```

   Repeat this step for every server-facing VLAN on each MX Series device by using the appropriate trunk interface names.

3. Associate a Layer 3 interface with each VLAN on the MX Series device.

   This example associates the `irb.17` interface with `vlan17`.

   Include the `routing-interface` statement and specify `irb.17` at the `[edit logical-systems Trust1 bridge-domains vlan17]` hierarchy level.

   ```
   [edit logical-systems Trust1 bridge-domains vlan17]
   user@mx# set routing-interface irb.17
   ```

   Repeat this step for every server-facing VLAN on each MX Series device by using the appropriate Layer 3 interface name.

4. Associate the logical interfaces with each interconnection VLAN on the MX Series device.

   This example associates logical interface `xe-1/1/0.15` that is connected to the SRX Series device and logical interface `xe-0/1/0.15` that is connected to the other MX Series device with `vlan15`.

   Include the `interface` statement and specify `xe-1/1/0.15` at the `[edit logical-systems Trust1 bridge-domains vlan15]` hierarchy level.

Include the `interface` statement and specify `xe-0/1/0.15` at the `[edit logical-systems Trust1 bridge-domains vlan15]` hierarchy level.

```
[edit logical-systems Trust1 bridge-domains vlan15]
user@mx# set interface xe-1/1/0.15
user@mx# set interface xe-0/1/0.15
```

Repeat this step for every interconnect VLAN on each MX Series device by using the appropriate interconnect interface names.

5. Associate a Layer 3 interface with each interconnection VLAN on the MX Series device to support active participation in the OSPF protocol.

   This example associates the `irb.15` interface with `vlan15`.

   Include the `routing-interface` statement and specify `irb.15` at the `[edit logical-systems Trust1 bridge-domains vlan15]` hierarchy level.

```
[edit logical-systems Trust1 bridge-domains vlan15]
user@mx# set routing-interface irb.15
```

   Repeat this step for every server-facing VLAN on each MX Series device by using the appropriate Layer 3 interface name.

Configuring the Virtual Router Routing Instance

Step-by-Step Procedure

This procedure explains how to configure a single virtual router routing instance. This procedure shows a representative sample of the example configuration. The example does not show the configuration for every device.

1. Configure the routing instance type.

   This example configures the routing instance with the name `MX-VR2`. Include the `instance-type` statement and specify `virtual-router` as the type at the `[edit logical-systems Trust1 routing-instances MX-VR2]` hierarchy level.

```
[edit logical-systems Trust1 routing-instances MX-VR2]
user@mx# set instance-type virtual-router
```

Repeat this step for every virtual router in each MX Series device by using the appropriate virtual router name.

2. Add the IRB interfaces used by the virtual router routing instance.

   Include the `interface` statement and specify the name of each IRB interface at the `[edit routing-instances MX-VR2]` hierarchy level.

```
[edit logical-systems Trust1 routing-instances MX-VR2]
user@mx# set interface irb.15
user@mx# set interface irb.16
user@mx# set interface irb.17
user@mx# set interface irb.18
user@mx# set interface irb.1002
```

   Repeat this step for every virtual router in each MX Series device by using the appropriate interface names.

3. Configure the IGP protocol active interface used by the virtual router routing instance so the routing tables can be populated with the routes to the servers.

   This example configures one IRB interface to actively participate in the OSPF protocol area `0.0.0.0`.

   Include the `interface` statement and specify the name of the IRB interface at the `[edit logical-systems Trust1 routing-instances MX-VR2 protocols ospf area 0.0.0.0]` hierarchy level.

```
[edit routing-instances MX-VR2 protocols ospf area 0.0.0.0]
user@mx# set interface irb.15
```

   Repeat this step for every virtual router in each MX Series device by using the appropriate virtual router name.

4. Configure the interior gateway protocol passive interfaces that are associated with each VLAN within the virtual router routing instance.

   This example configures the IRB interfaces to passively participate in the OSPF protocol area `0.0.0.0`.

   Include the `passive` statement at the `[edit logical-systems Trust1 routing-instances MX-VR2 protocols ospf area 0.0.0.0 interface` *`irb-name`*`]` hierarchy level.

```
[edit logical-systems Trust1 routing-instances MX-VR2 protocols ospf area 0.0.0.0]
user@mx# set interface irb.16 passive
user@mx# set interface irb.17 passive
```

```
user@mx# set interface irb.18 passive
user@mx# set interface irb.1002 passive
```

Repeat this step for every virtual router in each MX Series device by using the appropriate virtual router name.

5. Configure the logical system router identifier.

   Include the `router-id` statement and specify `10.200.11.101` as the router identifier at the `[edit logical-systems Trust1 routing-instances MX-VR2 routing-options]` hierarchy level.

```
[edit logical-systems Trust1 routing-instances MX-VR2 routing-options]
user@mx# set router-id 10.200.11.101
```

   Repeat this step for every virtual router in each MX Series device by using the appropriate router identifier.

Configuring Management Interfaces

Step-by-Step Procedure

This procedure explains how to configure static routes to the management network and the IPv4 address family for the loopback logical interface. This procedure shows a representative sample of the configuration. The example does not show the configuration for every interface.

1. Configure static routes so the Ethernet management interface can reach the management network.

   Include the `route` statement and specify `10.0.0.0/8` as the IPv4 subnet address of the management network at the `[edit routing-options static]` hierarchy level.

   Include the `next-hop` statement, and specify the IPv4 host address of the next-hop router at the `[edit routing-options static route 10.0.0.0/8]` hierarchy level.

   Include the `retain` and `no-readvertise` statements at the `[edit routing-options static route 10.0.0.0/8]` hierarchy level.

```
[edit routing-options static]
user@mx# set route 10.0.0.0/8 next-hop 10.8.3.254
user@mx# set route 10.0.0.0/8 retain
user@mx# set route 10.0.0.0/8 no-readvertise
```

   Repeat this step for every MX Series device.

2. Configure the MX Series device management Ethernet interface. This example configures the `unit 0` logical interface.

   Include the `family` statement and specify the `inet` option at the `[edit fxp0 unit 0]` hierarchy level.

   Include the `address` statement and specify `10.8.3.212/24` as the IPv4 address at the `[edit interfaces fxp0 unit 0]` hierarchy level.

   ```
   [edit interfaces fxp0 unit 0]
   user@mx# set family inet address 10.8.3.212/24
   ```

   Repeat this step for every MX Series device by using the appropriate management interface address for that device.

3. Configure the loopback logical interface.

   Include the `family` statement and specify the `inet` option at the `[edit interfaces lo0 unit 0]` hierarchy level.

   ```
   [edit interfaces lo0 unit 0]
   user@mx# set family inet
   ```

   Repeat this step for every MX Series device.

Configuring Logical System Administrator Accounts

Step-by-Step Procedure

This procedure explains how to configure administrator account classes that are confined to the context of the logical system to which they are assigned and administrator accounts for each logical system.

1. Create administrator account classes.

   In this example, the `trust1-admin` user class is created with `all` permissions for the `Trust1` logical system.

   Include the `class` statement and specify `trust1-admin` as the class name at the `[edit system login]` hierarchy level.

   Include the `logical-system` statement and specify `Trust1` as the logical system name at the `[edit system login class trust1-admin]` hierarchy level.

Include the `permissions` statement and specify the `all` option at the `[edit system login class trust1-admin]` hierarchy level.

```
[edit system]
user@mx# set login class trust1-admin logical-system Trust1
user@mx# set login class trust1-admin permissions all
```

Repeat this step for the trust2-admin and untrust-admin classes on each MX Series device by using the appropriate logical-system name.

2. Create administrator accounts that correspond to each logical system in the MX Series device.

   In this example, the `trust1` user account is created and assigned the `trust1-admin` class.

   Include the `class` statement and specify `trust1-admin` as the user class at the `[edit system login user trust1]` hierarchy level.

   Include the `encrypted-password` statement and enter the encrypted password string at the `[edit system login user trust1 authentication]` hierarchy level.

```
[edit system]
user@mx# set login user trust1 class trust1-admin
user@mx# set login user trust1 authentication encrypted-password 12345
```

   Repeat this step for the trust2 and untrust user accounts on each MX Series device.

Configuring Management Automation

Step-by-Step Procedure

This procedure explains how to configure a known host to support background SCP file transfers, a commit script, and an archive site.

1. Configure a commit script.

   In this example, the script file name is `jais-activate-scripts.slax`.

   Include the `allow-transients` statement at the `[edit system scripts commit]` hierarchy level.

Include the `optional` statement at the `[edit system scripts commit file jais-activate-scripts.slax]` hierarchy level.

```
[edit system scripts commit]
user@mx# set allow-transients
user@mx# set file jais-activate-scripts.slax optional
```

2. Configure an event archive site.

   In this example the archive URL is the local `/var/tmp/` directory, and the name given to the destination is `juniper-aim`.

   Include the `archive-sites` statement and specify the archive URL at the `[edit event-options destinations juniper-aim]` hierarchy level.

```
[edit event-options destinations juniper-aim]
user@mx# set archive-sites "scp://admin@127.0.0.1://var/tmp" password “12345”
```

Configuring the Core Layer in the Untrusted Logical Systems

IN THIS SECTION

- Configuring Interfaces in the Untrusted Logical Systems | **164**
- Configuring VLANs in the Core Layer | **166**
- Configuring Protocols in the Untrusted Logical System | **168**

Configure the core layer by doing the following:

Configuring Interfaces in the Untrusted Logical Systems

Step-by-Step Procedure

This procedure explains how to configure the physical, logical, and Layer 3 routing interfaces for the logical system in the untrusted security zone of the core layer. This procedure shows a representative sample of the configuration. The example does not show the configuration for every interface.

1. Configure the 10-Gigabit redundant Ethernet interfaces connected to the other MX Series device shown in Figure 13 on page 146.

This example configures logical interface 19 on the xe-0/3/0 interface under the logical system named Untrust to participate in VLAN 19. Include the encapsulation statement and specify the vlan-bridge option at the [edit logical-systems Untrust interfaces xe-0/3/0 unit 19] hierarchy level.

Include the vlan-id statement and specify 19 as the VLAN tag ID at the [edit logical-systems Untrust interfaces xe-0/3/0 unit 19] hierarchy level.

```
[edit logical-systems Untrust interfaces xe-0/3/0 unit 19]
user@mx# set encapsulation vlan-bridge
user@mx# set vlan-id 19
```

Repeat this step for every redundant Ethernet interface connected to the other MX Series device by using the appropriate interface name, logical interface number, VLAN ID, and logical system name.

2. Configure the 10-Gigabit Ethernet interfaces connected to the SRX Series device.

   This example configures logical interface 19 on the xe-2/2/0 interface under the logical system named Untrust to participate in VLAN 19.

   Include the encapsulation statement and specify the vlan-bridge option at the [edit logical-systems Untrust interfaces xe-2/2/0 unit 19] hierarchy level.

   Include the vlan-id statement and specify 19 as the VLAN tag ID at the [edit logical-systems Untrust interfaces xe-2/2/0 unit 19] hierarchy level.

   ```
   [edit logical-systems Untrust interfaces xe-2/2/0 unit 19]
   user@mx# set encapsulation vlan-bridge
   user@mx# set vlan-id 19
   ```

   Repeat this step for every redundant Ethernet interface connected to the SRX Series device by using the appropriate interface name, logical interface number, VLAN ID, and logical system name.

3. Configure the 10-Gigabit Ethernet interfaces connected to the IP-based/MPLS-based core network.

   This example configures logical interface 0 on the xe-1/3/0 interface under the logical system named Untrust.

   Include the address statement and specify 10.200.4.1/30 as the IPv4 address at the [edit logical-systems Untrust interfaces xe-1/3/0 unit 0 family inet] hierarchy level.

Include the `family` statement and specify the `mpls` option at the `[edit logical-systems Untrust interfaces xe-1/3/0 unit 0]` hierarchy level.

```
[edit logical-systems Untrust interfaces xe-1/3/0 unit 0]
user@mx# set family inet address 10.200.4.1/30
user@mx# set family mpls
```

Repeat this step for every 10-Gigabit Ethernet interface connected to the service provider network by using the appropriate interface name, logical interface number, IPv4 address, and logical system name.

4. Configure the Layer 3 IRB interface address.

   This example configures the `unit 19` logical interface that participates in VLAN 19 with `10.19.2.1/24` as the IPv4 address under the logical system named `Untrust`.

   Include the `address` statement and specify `10.19.2.1/24` as the IPv4 address at the `[edit logical-systems Untrust interfaces irb unit 19 family inet]` hierarchy level.

   ```
   [edit logical-systems Untrust interfaces irb unit 19 family inet]
   user@mx# set address 10.19.2.1/24
   ```

   Repeat this step for every Layer 3 IRB interface by using the appropriate logical interface name and IPv4 address.

5. Configure an IP address for the loopback logical interface of the Logical System `Untrust`.

   Include the `address` statement and specify `10.200.11.1/32` as the IPv4 address at the `[edit logical-systems Untrust interfaces lo0 unit 1 family inet]` hierarchy level.

   ```
   [edit logical-systems Untrust interfaces lo0 unit 1 family inet]
   user@mx# set address 10.200.11.1/32
   ```

   Repeat this step for every MX Series device by using the appropriate IPv4 address.

Configuring VLANs in the Core Layer

Step-by-Step Procedure

This procedure explains how to configure the VLAN names and tag IDs and associate interfaces and Layer 3 routing interfaces with each core interconnect VLAN. This procedure shows a representative sample of the configuration. The example does not show the configuration for every VLAN.

1. Configure the VLAN name and tag ID (number) for each core interconnect VLAN on the MX Series device.

   This example configures a VLAN with the name `vlan14` and tag ID `14` in the Logical System `Untrust`.

   Include the `vlan-id` statement and specify `14` as the VLAN ID at the `[edit logical-systems Untrust bridge-domains vlan14]` hierarchy level.

   ```
   [edit logical-systems Untrust bridge-domains vlan14]
   user@mx# set vlan-id 14
   ```

   Repeat this step for every VLAN on each MX Series device by using the appropriate VLAN names and tag IDs.

2. Associate the logical interfaces with each VLAN on the MX Series device.

   This example associates logical interface `xe-0/3/0.14` that is connected to the other MX Series device and `xe-2/2/0.14` that is connected to the SRX Series device with `vlan14`.

   Include the `interface` statement and specify `xe-0/3/0.14` at the `[edit logical-systems Untrust bridge-domains vlan14]` hierarchy level.

   Include the `interface` statement and specify `xe-2/2/0.14` at the `[edit logical-systems Untrust bridge-domains vlan14]` hierarchy level.

   ```
   [edit logical-systems Untrust bridge-domains vlan14]
   user@mx# set interface xe-0/3/0.14
   user@mx# set interface xe-2/2/0.14
   ```

   Repeat this step for every core interconnect VLAN on each MX Series device by using the appropriate interface names.

3. Associate a Layer 3 interface with each VLAN on the MX Series device.

   This example associates the `irb.14` interface with `vlan14`.

   Include the `routing-interface` statement and specify `irb.14` at the `[edit logical-systems Untrust bridge-domains vlan14]` hierarchy level.

   ```
   [edit logical-systems Untrust bridge-domains vlan14]
   user@mx# set routing-interface irb.14
   ```

   Repeat this step for every core interconnect VLAN on each MX Series device by using the appropriate Layer 3 interface name.

Configuring Protocols in the Untrusted Logical System

Step-by-Step Procedure

This procedure explains how to configure the BGP, MPLS, RSVP, and OSPF protocols for the Logical System Untrust. This procedure shows a representative sample of the configuration. The example does not show the configuration for every device.

1. Add interfaces to the OSPF protocol on the MX Series device.

   This example adds logical interfaces `xe-1/3/0.0` and `lo0.1` to the OSPF protocol used in the core network.

   Include the `interface` statement and specify the `xe-1/3/0.0` and `lo0.1` interfaces at the `[edit logical-systems Untrust protocols ospf area 0.0.0.0]` hierarchy level.

   ```
   [edit logical-systems Untrust protocols ospf area 0.0.0.0]
   user@mx# set interface xe-1/3/0.0
   user@mx# set interface lo0.1
   ```

   Repeat this step for every 10-Gigabit Ethernet interface connected to the core layer devices by using the appropriate interface name.

2. Configure the Generic Router Encapsulation (GRE) tunnel.

   This example enables a dynamic GRE tunnel named `GRE1`.

   Include the `gre` statement to specify the tunnel type at the `[edit logical-systems Untrust routing-options dynamic-tunnel GRE1]` hierarchy level.

   Include the `source-address` statement and specify `10.200.11.1` as the IPv4 source address at the `[edit logical-systems Untrust routing-options dynamic-tunnel GRE1]` hierarchy level.

   Include the `destination-networks` statement and specify `0.0.0.0/0` as the destination prefix at the `[edit logical-systems Untrust routing-options dynamic-tunnel GRE1]` hierarchy level.

   ```
   [edit logical-systems Untrust routing-options dynamic-tunnel GRE1]
   user@mx# set source-address 10.200.11.1
   user@mx# set gre
   user@mx# set destination-networks 0.0.0.0/0
   ```

   Repeat this step for each MX Series device by using the appropriate source address.

3. Configure the Logical System local autonomous system number and router identifier.

Include the autonomous-system statement and specify 64500 as the autonomous system number at the [edit logical-systems Untrust routing-options] hierarchy level.

Include the router-id statement and specify 10.200.11.101 as the router identifier at the [edit logical-systems Untrust routing-options] hierarchy level.

```
[edit logical-systems Untrust]
user@mx# set routing-options autonomous-system 64500
user@mx# set routing-options router-id 10.200.11.101
```

Repeat this step for each MX Series device by using the appropriate router identifier and autonomous system number 64500.

4. Configure the internal BGP peer group.

Include the type statement and specify the internal option at the [edit logical-systems Untrust protocols bgp group int] hierarchy level.

Include the local-address statement and specify the router ID (10.200.11.1) of Logical System Untrust as the local address at the [edit logical-systems Untrust protocols bgp group int] hierarchy level.

Include the unicast statement at the [edit logical-systems Untrust protocols bgp group int family inet] and [edit logical-systems Untrust protocols bgp group int family inet-vpn] hierarchy levels.

Include the local-as statement and specify 64500 as the local autonomous system number at the [edit logical-systems Untrust protocols bgp group int] hierarchy level.

Include the peer-as statement and specify 64500 as the peer autonomous system number at the [edit logical-systems Untrust protocols bgp group int] hierarchy level.

Include the neighbor statement and specify the neighbor IPv4 addresses at the [edit logical-systems Untrust protocols bgp group int] hierarchy level.

The neighbor addresses are the router ID addresses of the other MX Series device in the local data center, MX Series devices in a remote data center, and routers located in the IP-based/MPLS-based core network.

```
[edit logical-systems Untrust protocols bgp group int]
user@mx# set type internal
user@mx# set local-address 10.200.11.1
user@mx# set family inet unicast
user@mx# set family inet-vpn unicast
user@mx# set local-as 64500
user@mx# set peer-as 64500
user@mx# set neighbor 10.200.11.2
```

```
user@mx# set neighbor 10.200.11.3
user@mx# set neighbor 10.200.11.4
```

Repeat this step for every MX Series device.

5. Add interfaces to the MPLS protocol used in the service provider core network.

   This example adds the `xe-1/3/0.0` and `xe-2/3/0.0` interfaces that are connected to the service provider core network.

   Include the `interface` statement and specify the `xe-1/3/0.0` and `xe-2/3/0.0` interfaces at the `[edit logical-systems Untrust protocols mpls]` hierarchy level.

```
[edit logical-systems Untrust protocols mpls]
user@mx# set interface xe-1/3/0.0
user@mx# set interface xe-2/3/0.0
```

   Repeat this step for every MX Series device.

6. Create an MPLS LSP to the router that is located in the MPLS-based core network.

   This example creates an LSP named `to-core-router`.

   Include the `to` statement and specify `10.200.11.3` as the IPv4 address of the core router at the `[edit logical-systems Untrust protocols mpls label-switched-path to-core-router]` hierarchy level.

   Include the `no-cspf` statement at the `[edit logical-systems Untrust protocols mpls]` hierarchy level.

```
[edit logical-systems Untrust protocols mpls]
user@mx# set label-switched-path to-core-router to 10.200.11.3
user@mx# set no-cspf
```

   Repeat this step for every MX Series device.

7. Add interfaces to the RSVP protocol used in the MPLS-based core network.

   Include the `interface` statement and specify the `xe-1/3/0.0` and `xe-2/3/0.0` interfaces at the `[edit logical-systems Untrust protocols rsvp]` hierarchy level.

```
[edit logical-systems Untrust protocols rsvp]
user@mx# set interface xe-1/3/0.0
user@mx# set interface xe-2/3/0.0
```

   Repeat this step for every MX Series device.

Configuring the Security Device

IN THIS SECTION

- Configuring the Redundant Ethernet Interface Link Aggregation Group | 171
- Configuring the SRX Series Cluster | 173
- Creating Security Zones and Configuring the In-Bound Traffic Policy Action | 175
- Configuring the Security Zone Policies | 176
- Creating the Routing Policies | 178
- Configuring the Virtual Router Routing Instance | 181
- Results | 184

The following procedures explain how to configure the redundant Ethernet interfaces, node cluster, security zones, security policies, and routing policies for the trusted security zone of the access layer.

Configuring the Redundant Ethernet Interface Link Aggregation Group

Step-by-Step Procedure

This procedure explains how to configure the redundant Ethernet interface link aggregation group. This procedure shows a representative sample of the configuration. The example does not show the configuration for every interface.

1. Configure the number of aggregated Ethernet interfaces supported on the node.

   This example enables support for two interfaces.

   Include the `device-count` statement and specify `2` as the number of interfaces supported at the `[edit chassis aggregated-devices ethernet]` hierarchy level.

   ```
   [edit chassis aggregated-devices ethernet]
   user@srx# set device-count 2
   ```

   Repeat this step for every SRX Series device by using the appropriate device count.

2. Assign 10-Gigabit Ethernet child interfaces to the redundant Ethernet (reth) parent interface.

   This example assigns the `xe-1/0/0` 10-Gigabit Ethernet child interface to the `reth1` parent interface on Node0.

Include the `redundant-parent` statement and specify `reth1` as the parent interface at the `[edit interfaces xe-1/0/0 gigether-options]` hierarchy level.

```
[edit interfaces xe-1/0/0 gigether-options]
user@srx# set redundant-parent reth1
```

Repeat this step for every redundant Ethernet interface by using the appropriate interface name and redundant parent name.

3. Configure the redundant Ethernet parent interface options.

   This example configures the `reth1` redundant parent interface.

   Include the `redundancy-group` statement and specify `1` as the group number at the `[edit interfaces reth1 redundant-ether-options]` hierarchy level.

   Include the `vlan-tagging` statement at the `[edit interfaces reth1]` hierarchy level.

   ```
   [edit interfaces reth1]
   user@srx# set redundant-ether-options redundancy-group 1
   user@srx# set vlan-tagging
   ```

   Repeat this step for every redundant parent interface by using the appropriate redundant parent name and redundancy group number.

4. Configure the redundant Ethernet parent logical interfaces.

   This example configures the `unit 15` logical interface.

   Include the `address` statement and specify `10.15.2.2/24` as the IPv4 address at the `[edit interfaces reth1 unit 15 family inet]` hierarchy level.

   Include the `vlan-id` statement and specify `15` as the VLAN identifier at the `[edit interfaces reth1 unit 15]` hierarchy level.

   ```
   [edit interfaces reth1 unit 15]
   user@srx# set family inet address 10.15.2.2/24
   user@srx# set vlan-id 15
   ```

   Repeat this step for every redundant parent interface by using the appropriate redundant parent name, IPv4 address, and VLAN identifier.

Step-by-Step Procedure

This procedure explains how to configure fabric connections between the nodes in the cluster. This procedure shows a representative sample of the configuration. The example does not show the configuration for every interface.

1. Configure the 10-Gigabit Ethernet interface to serve as the fabric between the cluster nodes.

   This example configures `xe-1/0/1` as the child fabric interface and `fab0` as the parent fabric interface. The connection is from SRX0 to SRX1.

   Include the `member-interfaces` statement and specify the `xe-1/0/1` interface at the `[edit interfaces fab0 fabric-options]` hierarchy level.

   ```
   [edit interfaces fab0 fabric-options]
   user@srx# set member-interfaces xe-1/0/1
   ```

   Repeat this step for every 10-Gigabit Ethernet interface that is part of the cluster fabric by using the appropriate child interface name and parent interface name.

2. Configure the number of redundant Ethernet interfaces that the cluster supports.

   This example configures `4` as the number of interfaces.

   Include the `reth-count` statement and specify `4` as the number of interfaces at the `[edit chassis cluster]` hierarchy level.

   ```
   [edit chassis cluster]
   user@srx# set reth-count 4
   ```

   Repeat this step for every SRX Series device in the cluster.

3. Configure the node priority for the redundancy group to determine which node is primary and which is secondary.

   This example configures `node 0` with a higher priority.

   Include the `priority` statement and specify `200` at the `[edit chassis cluster redundancy-group 1 node 0]` hierarchy level.

Include the `priority` statement and specify `100` at the `[edit chassis cluster redundancy-group 1 node 1]` hierarchy level.

```
[edit chassis cluster redundancy-group 1]
user@srx# set node 0 priority 200
user@srx# set node 1 priority 100
```

Repeat this step for every redundancy group on every SRX Series device in the cluster.

4. Allow a node with a higher priority to initiate a failover to become the primary node for the redundancy group.

   Include the `preempt` statement at the `[edit chassis cluster redundancy-group 1]` hierarchy level.

   ```
   [edit chassis cluster redundancy-group 1]
   user@srx# set preempt
   ```

   Repeat this step for every redundancy group on every SRX Series device in the cluster.

5. Enable control link recovery to be done automatically.

   Include the `control-link-recovery` statement at the `[edit chassis cluster]` hierarchy level.

   ```
   [edit chassis cluster]
   user@srx# set control-link-recovery
   ```

   Repeat this step for every redundancy group on every SRX Series device in the cluster.

6. Enable interface monitoring to monitor the health of the interfaces and trigger redundancy group failover.

   This example configures the `xe-1/0/0` interface with a weight of `255`.

   Include the `weight` statement at the `[edit chassis cluster redundancy-group 1 interface-monitor xe-1/0/0]` hierarchy level.

   ```
   [edit chassis cluster redundancy-group 1 interface-monitor xe-1/0/0]
   user@srx# set weight 255
   ```

   Repeat this step for every redundancy group interface on every SRX Series device in the cluster.

Creating Security Zones and Configuring the In-Bound Traffic Policy Action

Step-by-Step Procedure

This procedure explains how to configure the trusted and untrusted security zones on the SRX Series device. This procedure shows a representative sample of the configuration. The example does not show the configuration for every zone.

1. Assign a redundant Ethernet logical interface to a trusted zones.

   This example assigns the `reth1.15` interface to the `Trust2` zone.

   Include the `interfaces` statement and specify `reth1.15` as the interface in the zone at the `[edit security zones security-zone Trust2]` hierarchy level.

   ```
   [edit security zones security-zone Trust2]
   user@srx# set interfaces reth1.15
   ```

   Repeat this step for every trusted security zone by using the appropriate zone name and redundant Ethernet logical interface name.

2. Assign a redundant Ethernet logical interface to the untrusted zones.

   This example assigns the `reth2.19` interface to the `Untrust2` zone.

   Include the `interfaces` statement and specify `reth2.19` as the interface in the zone at the `[edit security zones security-zone Untrust2]` hierarchy level.

   ```
   [edit security zones security-zone Untrust2]
   user@srx# set interfaces reth2.19
   ```

   Repeat this step for every untrusted security zone by using the appropriate zone name and redundant Ethernet logical interface name.

3. Enable all inbound system services traffic in the trusted security zone.

   This example enables all services for the `Trust2` zone.

   Include the `system-services` statement and specify the `all` option at the `[edit security zones security-zone Trust2 host-inbound-traffic]` hierarchy level.

   ```
   [edit security zones security-zone Trust2 host-inbound-traffic]
   user@srx# set system-services all
   ```

Repeat this step for every security zone on the SRX Series device where system services are allowed.

4. Enable all protocols for inbound traffic in the trusted security zone.

   This example enables all protocols for the `Trust2` zone.

   Include the `protocols` statement and specify the `all` option at the `[edit security zones security-zone Trust2 host-inbound-traffic]` hierarchy level.

```
[edit security zones security-zone Trust2 host-inbound-traffic]
user@srx# set protocols all
```

   Repeat this step for every security zone on the SRX Series device where all protocols are allowed for inbound traffic.

Configuring the Security Zone Policies

Step-by-Step Procedure

This procedure explains how to configure the security zone policies on the SRX Series device. This procedure shows a representative sample of the configuration. The example does not show the configuration for every policy.

1. Define which zone traffic is coming from and which zone traffic is going to for the policy being created.

   This example defines the from zone as `Trust2` and the to zone as `Untrust2`.

   On a single command line, include the `from-zone` statement and specify `Trust2`, include the `to-zone` statement and specify `Untrust2`, include the `policy` statement and specify `denyftp` as the policy name, and included the `match` statement at the `[edit security policies]` hierarchy level.

```
[edit security policies]
user@srx# set from-zone Trust2 to-zone Untrust2 policy denyftp match
```

   Repeat this step for every policy that controls traffic between zones.

2. Configure the policy match criteria for denying traffic.

   This example matches the Junos OS FTP application from any source to any destination address in a policy named `denyftp`.

   Include the `source-address` statement and specify `any` as the IPv4 address at the `[edit security policies from-zone Trust2 to-zone Untrust2 policy denyftp match]` hierarchy level.

Include the destination-address statement and specify any as the IPv4 address at the [edit security policies from-zone Trust2 to-zone Untrust2 policy denyftp match] hierarchy level.

Include the application statement and specify junos-ftp as the application at the [edit security policies from-zone Trust2 to-zone Untrust2 policy denyftp match] hierarchy level.

```
[edit security policies from-zone Trust2 to-zone Untrust2 policy denyftp match]
user@srx# set source-address any
user@srx# set destination-address any
user@srx# set application junos-ftp
```

Repeat this step for every protocol matching policy by using the correct protocol.

3. Block specific applications from passing from the Trust2 zone to the Untrust2 zone.

   This example denies the Junos OS FTP application from the Trust2 zone to the Untrust2 zone.

   Include the deny statement at the [edit security policies from-zone Trust2 to-zone Untrust2 policy denyftp then] hierarchy level.

   ```
   [edit security policies from-zone Trust2 to-zone Untrust2 policy denyftp then]
   user@srx# set deny
   ```

   Repeat this step for every deny policy.

4. Configure the policy match criteria for allowing traffic.

   This example matches any application from any source to any destination address in a policy named allow_all.

   Include the source-address statement and specify any as the IPv4 address at the [edit security policies from-zone Trust2 to-zone Untrust2 policy allow_all match] hierarchy level.

   Include the destination-address statement and specify any as the IPv4 address at the [edit security policies from-zone Trust2 to-zone Untrust2 policy allow_all match] hierarchy level.

   Include the application statement and specify any as the application at the [edit security policies from-zone Trust2 to-zone Untrust2 policy allow_all match] hierarchy level.

   ```
   [edit security policies from-zone Trust2 to-zone Untrust2 policy allow_all match]
   user@srx# set source-address any
   user@srx# set destination-address any
   user@srx# set application any
   ```

Repeat this step for every application matching policy.

5. Permit any application traffic to pass from the Trust2 zone to the Untrust2 zone.

   This example allows any application traffic from the `Trust2` zone to the `Untrust2` zone.

   Include the `permit` statement at the `[edit security policies from-zone Trust2 to-zone Untrust2 policy allow_all then]` hierarchy level.

```
[edit security policies from-zone Trust2 to-zone Untrust2 policy allow_all then]
user@srx# set permit
```

   Repeat this step for every permit policy.

Creating the Routing Policies

Step-by-Step Procedure

This procedure explains how to creates the routing policies on the SRX Series device that can be applied to the appropriate routing instances. This procedure shows a representative sample of the configuration. The example does not show the configuration for every policy.

1. Create a policy to set the local preference for BGP routes to 120.

   This example creates a policy named `local-pref-120` that sets the BGP local preference value for received routes advertised by BGP to `120`.

   Include the `protocol` statement and specify `bgp` as the value at the `[edit policy-options policy-statement local-pref-120 term term1 from]` hierarchy level.

   Include the `local-preference` statement and specify `120` as the value at the `[edit policy-options policy-statement local-pref-120 term term1 then]` hierarchy level.

```
[edit policy-options policy-statement local-pref-120]
user@srx# set term term1 from protocol bgp
user@srx# set term term1 then local-preference 120
```

   Repeat this step for each SRX Series device.

2. Configure the match criteria for a policy named `default-ospf` to accept all aggregate (generated) routes.

   Include the `protocol` statement and specify `aggregate` as the protocol to match at the `[edit policy-options policy-statement default-ospf term term1 from]` hierarchy level.

Include the route-filter statement and specify 0.0.0.0/0 exact as the match criteria at the [edit policy-options policy-statement default-ospf term term1 from] hierarchy level.

```
[edit policy-options policy-statement default-ospf term term1 from]
user@srx# set protocol aggregate
user@srx# set route-filter 0.0.0.0/0 exact
```

Repeat this step for each SRX Series device.

3. Configure the action for a policy to set the metric to 0, and set the external route type to 1.

   This example configures a policy named default-ospf that sets the metric to 0, sets the external route to type 1, and accepts aggregate routes into the routing table.

   Include the metric statement and specify 0 as the external type at the [edit policy-options policy-statement default-ospf term term1 then] hierarchy level.

   Include the type statement and specify 1 as the external route type at the [edit policy-options policy-statement default-ospf term term1 then external] hierarchy level.

   Include the accept statement at the [edit policy-options policy-statement default-ospf term term1 then] hierarchy level.

   ```
   [edit policy-options policy-statement default-ospf term term1 then]
   user@srx# set metric 0
   user@srx# set external type 1
   user@srx# set accept
   ```

   Repeat this step for each SRX Series device.

4. Create a policy that accepts OSPF routes with specified prefixes.

   This example creates a policy named trust2-ebgp-out that accepts OSPF routes with the route prefixes that correspond to the subnets for each trust VLAN.

   Include the protocol statement and specify ospf as the protocol at the [edit policy-options policy-statement trust2-ebgp-out term term1 from] hierarchy level.

   Include the route-filter statement and specify the VLAN subnet addresses and the exact match keyword at the [edit policy-options policy-statement trust2-ebgp-out term term1 from] hierarchy level.

Include the accept statement at the [edit policy-options policy-statement trust2-ebgp-out term term1 then] hierarchy level.

```
[edit policy-options policy-statement trust2-ebgp-out term term1]
user@srx# set from protocol ospf
user@srx# set from route-filter 10.16.2.0/24 exact
user@srx# set from route-filter 10.17.2.0/24 exact
user@srx# set from route-filter 10.18.2.0/24 exact
user@srx# set then accept
```

Repeat this step for each SRX Series device.

5. Create a policy that accepts BGP routes if the route type is external.

   This example creates a policy named check-bgp-routes that accepts BGP routes only if the route type is external.

   Include the protocol statement and specify bgp as the protocol at the [edit policy-options policy-statement check-bgp-routes term term1 from] hierarchy level.

   Include the route-type statement and specify the external option at the [edit policy-options policy-statement check-bgp-routes term term1 from] hierarchy level.

   Include the accept statement at the [edit policy-options policy-statement check-bgp-routes term term1 then] hierarchy level.

   ```
   [edit policy-options policy-statement check-bgp-routes term term1]
   user@srx# set from protocol bgp
   user@srx# set from route-type external
   user@srx# set then accept
   ```

   Repeat this step for each SRX Series device.

6. Create a policy that accepts routes from other virtual router routing instances.

   This example creates a policy named from_srx_vr1 that accepts routes from routing instance SRX-VR1.

   Include the instance statement and specify SRX-VR1 as the routing instance name at the [edit policy-options policy-statement from_srx_vr1 term term1 from] hierarchy level.

Include the `accept` statement at the `[edit policy-options policy-statement from_srx_vr1 term term1 then]` hierarchy level.

```
[edit policy-options policy-statement from_srx_vr1 term term1]
user@srx# set from instance SRX-VR1
user@srx# set then accept
```

Repeat this step for each virtual router in each SRX Series device.

Configuring the Virtual Router Routing Instance

Step-by-Step Procedure

This procedure explains how to configure a single virtual router routing instance. This procedure shows a representative sample of the example configuration. The example does not show the configuration for every virtual router routing instance.

1. Configure the routing instance type.

   This example configures the routing instance with the name SRX-VR2.

   Include the `instance-type` statement and specify `virtual-router` as the type at the `[edit routing-instances SRX-VR2]` hierarchy level.

   ```
   [edit routing-instances SRX-VR2]
   user@srx# set instance-type virtual-router
   ```

   Repeat this step for every virtual router in each SRX Series device by using the appropriate virtual router name.

2. Add the redundant Ethernet interfaces used by the virtual router routing instance.

   This example adds `reth1.15` and `reth2.19` interfaces to the `SRX-VR2` routing instance.

   Include the `interface` statement and specify the name of the redundant Ethernet interface at the `[edit routing-instances SRX-VR2]` hierarchy level.

   ```
   [edit routing-instances SRX-VR2]
   user@srx# set interface reth1.15
   user@srx# set interface reth2.19
   ```

Repeat this step for every virtual router in each SRX Series device by using the appropriate virtual router name and interface names.

3. Configure the routing options used by the virtual router routing instance.

   This example configures the autonomous system number and enables the graceful restart feature on the SRX-VR2 routing instance.

   Include the autonomous-system statement and specify 65019 as the autonomous system number at the [edit routing-instances SRX-VR2 routing-options] hierarchy level.

   Include the graceful-restart statement at the [edit routing-instances SRX-VR2 routing-options] hierarchy level.

```
[edit routing-instances SRX-VR2 routing-options]
user@srx# set autonomous-system 65019
user@srx# set graceful-restart
```

   Repeat this step for every virtual router in each SRX Series device by using the appropriate virtual router name and interface names.

4. Apply the routing policy that accepts external BGP routes and uses them as generated routes for the routing instance.

   This example applies the policy named check-bgp-routes to the SRX-VR2 routing instance.

   Include the policy statement and specify check-bgp-routes at the [edit routing-instances SRX-VR2 routing-options generate route 0.0.0.0/0] hierarchy level.

   Include the graceful-restart statement at the [edit routing-instances SRX-VR2 routing-options] hierarchy level.

```
[edit routing-instances SRX-VR2 routing-options]
user@srx# set generate route 0.0.0.0/0 policy
user@srx# set graceful-restart
```

   Repeat this step for every virtual router in each SRX Series device by using the appropriate virtual router name and interface names.

5. Apply the routing policy that accepts routes from other routing instances.

   This example applies the policy named from_srx_vr1 to the SRX-VR2 routing instance.

Include the instance-import statement and specify from_srx_vr1 at the [edit routing-instances SRX-VR2 routing-options] hierarchy level.

```
[edit routing-instances SRX-VR2 routing-options]
user@srx# set instance-import from_srx_vr1
```

Repeat this step for every virtual router in each SRX Series device except the SRX-VR1 instance.

6. Configure the IGP protocol export policy used by the virtual router routing instance in the trusted security zone.

   This example configures the default-ospf policy.

   Include the export statement and specify default-ospf as the policy name at the [edit routing-instances SRX-VR2 protocols ospf] hierarchy level.

   ```
   [edit routing-instances SRX-VR2 protocols ospf]
   user@srx# set export default-ospf
   ```

   Repeat this step for every virtual router in each SRX Series device by using the appropriate virtual router name and policy name.

7. Configure the IGP protocol active and passive interfaces used by the virtual router routing instance in the trusted security zone.

   This example configures the reth1.15 redundant Ethernet interface to actively participate in the OSPF protocol area 0.0.0.0, and the reth2.19 redundant Ethernet interface to passively participate.

   Include the interface statement and specify reth1.15 at the [edit routing-instances SRX-VR2 protocols ospf area 0.0.0.0] hierarchy level.

   Include the interface statement and specify reth2.19 at the [edit routing-instances SRX-VR2 protocols ospf area 0.0.0.0] hierarchy level.

   Include the passive statement at the [edit routing-instances SRX-VR2 protocols ospf area 0.0.0.0 reth2.19] hierarchy level.

   ```
   [edit routing-instances SRX-VR2 protocols ospf area 0.0.0.0]
   user@srx# set interface reth1.15
   user@srx# set interface reth2.19 passive
   ```

   Repeat this step for every virtual router in each SRX Series device by using the appropriate virtual router name and interface names.

8. Configure the BGP protocol peer groups used by the virtual router routing instance in the untrusted security zone.

   Include the `type` statement and specify the `external` option at the `[edit routing-instances SRX-VR2 protocols bgp group MX0-vrf]` hierarchy level.

   Include the `peer-as` statement and specify `64500` as the peer autonomous system number at the `[edit routing-instances SRX-VR2 protocols bgp group MX0-vrf]` hierarchy level.

   Include the `neighbor` statement and specify `10.19.2.1` as the IPv4 neighbor address at the `[edit routing-instances SRX-VR2 protocols bgp group MX0-vrf]` hierarchy level. The neighbor address is the IRB Logical interface address of the VRF routing instance on the MX Series device.

   ```
   [edit routing-instances SRX-VR2 protocols bgp group MX0-vrf]
   user@srx# set type external
   user@srx# set peer-as 64500
   user@srx# set neighbor 10.19.2.1
   ```

   Repeat this step for every virtual router in each SRX Series device by using the appropriate virtual router name, instance type, neighbor address, and peer AS number.

9. Configure the BGP protocol peer groups export and import policies used by the virtual router routing instance in the untrusted security zone.

   Include the `export` statement and specify `trust2-ebgp-out` as the export policy name at the `[edit routing-instances SRX-VR2 protocols bgp group MX0-vrf]` hierarchy level.

   Include the `import` statement and specify `local-pref-120` as the import policy name at the `[edit routing-instances SRX-VR2 protocols bgp group MX0-vrf]` hierarchy level.

   ```
   [edit routing-instances SRX-VR2 protocols bgp group MX0-vrf]
   user@srx# set export trust2-ebgp-out
   user@srx# set import local-pref-120
   ```

   Repeat this step for every virtual router in each SRX Series device by using the appropriate virtual router name, export policy, and import policy.

Results

The configuration steps of this example have been completed. The following section is for your reference.

The relevant sample configuration for the EX Series device follows.

EX Series Device

```
system {
    scripts {
        commit {
            allow-transients;
            file jais-activate-scripts.slax {
                optional;
            }
        }
    }
    services {
        ftp;
        ssh;
        telnet;
        outbound-ssh {
            client 00187D0B670D {
                device-id FA022D;
                secret "$ABC123"; ## SECRET-DATA
                services netconf;
                10.8.7.32 port 7804;
            }
        }
    }
}
interfaces {
    ge-0/0/17 {
        unit 0 {
            family ethernet-switching {
                vlan {
                    members 17;
                }
            }
        }
    }
    xe-0/1/0 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
                vlan {
                    members all;
                }
```

```
            }
        }
    }
    xe-0/1/2 {
        unit 0 {
            enable;
            family ethernet-switching {
                port-mode trunk;
                vlan {
                    members all;
                }
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 127.0.0.1/32;
            }
        }
    }
    me0 {
        unit 0 {
            family inet {
                address 10.8.108.19/24;
            }
        }
    }
}
event-options {
    destinations {
        juniper-aim {
            archive-sites {
                "scp://admin@127.0.0.1://var/tmp" password "$ABC123"; ## SECRET-DATA
            }
        }
    }
}
routing-options {
    static {
        route 10.8.0.0/16 next-hop 10.8.108.254;
    }
}
```

```
protocols {
    rstp {
        interface xe-0/1/0.0 {
            disable;
        }
        interface xe-0/1/2.0 {
            disable;
        }
    }
}
security {
    ssh-known-hosts {
        host 127.0.0.1 {
            rsa-key AAAAB3NzaC1yc2;
        }
    }
}
ethernet-switching-options {
    redundant-trunk-group {
        group rtgroup1 {
            interface xe-0/1/0.0;
            interface xe-0/1/2.0 {
                primary;
            }
        }
    }
}
vlans {
    vlan17 {
        vlan-id 17;
        interface {
            xe-0/1/0.0;
            xe-0/1/2.0;
        }
    }
}
```

The relevant sample configuration for the MX Series device follows.

MX Series Device

```
groups {
    re0 {
        system {
            host-name MX0;
        }
    }
    re1 {
        system {
            host-name MX0re1;
        }
    }
}
apply-groups [re0 re1 ];
system {
    scripts {
        commit {
            allow-transients;
            file jais-activate-scripts.slax {
                optional;
            }
        }
    }
    login {
        class trust1-admin {
            logical-system Trust1;
            permissions all;
        }
        user trust1 {
            uid 2000;
            class trust1-admin;
            authentication {
                encrypted-password 12345; ## SECRET-DATA
            }
        }
    }
}
logical-systems {
    Trust1 {
        interfaces {
            xe-0/1/0 {
```

```
                unit 17 {
                    encapsulation vlan-bridge;
                    vlan-id 17;
                }
            }
            xe-1/0/0 {
                unit 17 {
                    encapsulation vlan-bridge;
                    vlan-id 17;
                }
            }
            xe-1/1/0 {
                unit 15 {
                    encapsulation vlan-bridge;
                    vlan-id 15;
                }
            }
            irb {
                unit 15 {
                    family inet {
                        address 10.15.2.3/24;
                    }
                }
                unit 17 {
                    family inet {
                        address 10.17.2.2/24 {
                            vrrp-group 17 {
                                virtual-address 10.17.2.1;
                                priority 200;
                                fast-interval 200;
                                preempt;
                                accept-data;
                            }
                        }
                    }
                }
            }
        }
        routing-instances {
            MX-VR2 {
                instance-type virtual-router;
                interface irb.15;
                interface irb.16;
```

```
                interface irb.17;
                interface irb.18;
                interface irb.1002;
                protocols {
                    ospf {
                        area 0.0.0.0 {
                            interface irb.16 {
                                passive;
                            }
                            interface irb.17 {
                                passive;
                            }
                            interface irb.18 {
                                passive;
                            }
                            interface irb.1002 {
                                passive;
                            }
                        }
                    }
                }
            }
        }
        bridge-domains {
            vlan15 {
                vlan-id 15;
                interface xe-1/1/0.15;
                interface xe-0/1/0.15; ## 'xe-0/1/0.15' is not defined
                routing-interface irb.15;
            }
            vlan17 {
                vlan-id 17;
                interface xe-1/0/0.17;
                interface xe-0/1/0.17;
                routing-interface irb.17;
            }
        }
    }
    Untrust {
        interfaces {
            xe-0/3/0 {
                unit 19 {
                    encapsulation vlan-bridge;
```

```
                vlan-id 19;
            }
        }
        xe-1/3/0 {
            unit 0 {
                family inet {
                    address 10.200.4.1/30;
                }
                family mpls;
            }
        }
        xe-2/2/0 {
            unit 19 {
                encapsulation vlan-bridge;
                vlan-id 19;
            }
        }
        irb {
            unit 19 {
                family inet {
                    address 10.19.2.1/24;
                }
            }
        }
        lo0 {
            unit 1 {
                family inet {
                    address 10.200.11.1/32;
                }
            }
        }
    }
    protocols {
        rsvp {
            interface xe-1/3/0.0;
            interface xe-2/3/0.0;
        }
        mpls {
            no-cspf;
            label-switched-path to-core-router {
                to 10.200.11.3;
            }
            interface xe-1/3/0.0;
```

```
            interface xe-2/3/0.0;
        }
        bgp {
            group int {
                type internal;
                local-address 10.200.11.1;
                family inet {
                    unicast;
                }
                family inet-vpn {
                    unicast;
                }
                peer-as 64500;
                local-as 64500;
                neighbor 10.200.11.2;
                neighbor 10.200.11.3;
                neighbor 10.200.11.4;
            }
        }
        ospf {
            area 0.0.0.0 {
                interface xe-1/3/0.0;
                interface lo0.1;
            }
        }
    }
    routing-options {
        router-id 10.200.11.101;
        autonomous-system 64500;
        dynamic-tunnels {
            GRE1 {
                source-address 10.200.11.1;
                gre;
                destination-networks {
                    0.0.0.0/0;
                }
            }
        }
    }
    bridge-domains {
        vlan14 {
            vlan-id 14;
            interface xe-0/3/0.14;
```

```
                interface xe-2/2/0.14;
                routing-interface irb.14;
            }
        }
    }
}
interfaces {
    xe-0/1/0 {
        flexible-vlan-tagging;
        encapsulation flexible-ethernet-services;
    }
    xe-0/3/0 {
        flexible-vlan-tagging;
        encapsulation flexible-ethernet-services;
    }
    xe-1/0/0 {
        flexible-vlan-tagging;
        encapsulation flexible-ethernet-services;
    }
    xe-1/1/0 {
        flexible-vlan-tagging;
        encapsulation flexible-ethernet-services;
    }
    xe-2/2/0 {
        flexible-vlan-tagging;
        encapsulation flexible-ethernet-services;
    }
    fxp0 {
        unit 0 {
            family inet {
                address 10.8.3.212/24;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet;
        }
    }
}
event-options {
    destinations {
        juniper-aim {
```

```
            archive-sites {
                "scp://admin@127.0.0.1://var/tmp" password "$ABC123"; ## SECRET-DATA
            }
        }
    }
}
routing-options {
    static {
        route 10.0.244.8/30 next-hop 10.0.134.10;
        route 10.0.0.0/8 {
            next-hop 10.8.3.254;
            retain;
            no-readvertise;
        }
    }
}
```

The relevant sample configuration for the SRX Series device follows.

SRX Series Device

```
system {
    host-name srx0;
    chassis {
        cluster {
            control-link-recovery;
            reth-count 4;
            redundancy-group 1 {
                node 0 priority 200;
                node 1 priority 100;
                preempt;
                interface-monitor {
                    xe-1/0/0 weight 255;
                }
            }
        }
    }
}
interfaces {
    xe-1/0/0 {
        gigether-options {
            redundant-parent reth1;
```

```
        }
    }
    fab0 {
        fabric-options {
            member-interfaces {
                xe-1/0/1;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 127.0.0.1/32;
            }
        }
    }
    reth1 {
        vlan-tagging;
        redundant-ether-options {
            redundancy-group 1;
        }
        unit 15 {
            family inet {
                address 10.15.2.2/24;
            }
            vlan-id 15;
        }
    }
}
policy-options {
    policy-statement check-bgp-routes {
        term term1 {
            from {
                protocol bgp;
                route-type external;
            }
            then accept;
        }
    }
    policy-statement default-ospf {
        term term1 {
            from {
                protocol aggregate;
```

```
                route-filter 0.0.0.0/0 exact;
            }
            then {
                metric 0;
                external {
                    type 1;
                }
                accept;
            }
        }
    }
    policy-statement from_srx_vr1 {
        term term1 {
            from instance SRX-VR1;
            then accept;
        }
    }
    policy-statement local-pref-120 {
        term term1 {
            from protocol bgp;
            then {
                local-preference 120;
            }
        }
    }
    policy-statement trust2-ebgp-out {
        term term1 {
            from {
                protocol ospf;
                route-filter 10.16.2.0/24 exact;
                route-filter 10.17.2.0/24 exact;
                route-filter 10.18.2.0/24 exact;
            }
            then accept;
        }
    }
}
security {
    policies {
        from-zone Trust2 to-zone Untrust2 {
            policy denyftp {
                match {
                    source-address any;
```

```
                    destination-address any;
                    application junos-ftp;
                }
                then {
                    deny;
                }
            }
            policy allow_all {
                match {
                    source-address any;
                    destination-address any;
                    application any;
                }
                then {
                    permit;
                }
            }
        }
    }
    zones {
        security-zone Trust2 {
            host-inbound-traffic {
                system-services {
                    all;
                }
                protocols {
                    all;
                }
            }
            interfaces {
                reth1.15;
            }
        }
        security-zone Untrust2 {
            interfaces reth2.19;
        }
    }
}
routing-instances {
    SRX-VR2 {
        instance-type virtual-router;
        interface reth1.15;
        interface reth2.19;
```

```
        routing-options {
            graceful-restart;
            autonomous-system 65019;
            instance-import from_srx_vr1;
        }
        protocols {
            bgp {
                group MX0-vrf {
                    import local-pref-120;
                    export trust2-ebgp-out;
                }
            }
            ospf {
                export default-ospf;
                area 0.0.0.0 {
                    interface reth1.15;
                    interface reth2.19 {
                        passive;
                    }
                }
            }
        }
    }
}
```

SEE ALSO

- Requirements of a Two-Tiered Virtualized Data Center for Large Enterprise Networks | **142**
- Two-Tiered Virtualized Data Center Solution for Large Enterprise Networks | **139**

RELATED DOCUMENTATION

- Monitoring Logical Systems for Routers and Switches | **368**

CHAPTER 3

Routing Protocols and Routing Policies on Logical Systems

BGP on Logical Systems | 200

IS-IS on Logical Systems | 265

OSPF on Logical Systems | 289

RSVP-Signaled Point-to-Multipoint LSPs on Logical Systems | 333

BGP on Logical Systems

IN THIS SECTION

- Example: Configuring Internal BGP Peering Sessions on Logical Systems | 200
- Example: Configuring External BGP on Logical Systems with IPv6 Interfaces | 214
- Example: Configuring BFD on Internal BGP Peer Sessions | 237
- Example: Configuring EBGP Multihop Sessions on Logical Systems | 250

Logical Systems enable you to configure the internal BGP sessions. For more information, see the following topics:

Example: Configuring Internal BGP Peering Sessions on Logical Systems

IN THIS SECTION

- Requirements | 200
- Overview | 200
- Configuration | 201
- Verification | 210

This example shows how to configure internal BGP peer sessions on logical systems.

Requirements

In this example, no special configuration beyond device initialization is required.

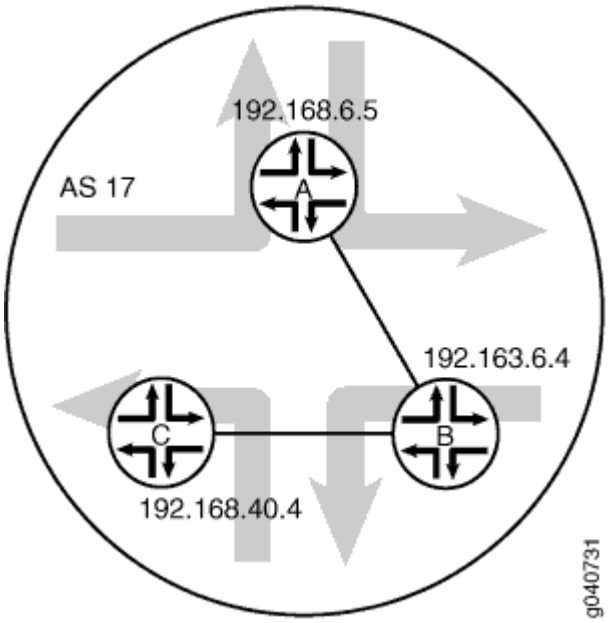
Overview

In this example, you configure internal BGP (IBGP) peering sessions.

In the sample network, the devices in AS 17 are fully meshed in the group **internal-peers**. The devices have loopback addresses 192.168.6.5, 192.163.6.4, and 192.168.40.4.

Figure 15 on page 201 shows a typical network with internal peer sessions.

Figure 15: Typical Network with IBGP Sessions



Configuration

IN THIS SECTION

- CLI Quick Configuration | **202**
- Device A | **203**

CLI Quick Configuration

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

```
set logical-systems A interfaces lt-0/1/0 unit 1 description to-B
set logical-systems A interfaces lt-0/1/0 unit 1 encapsulation ethernet
set logical-systems A interfaces lt-0/1/0 unit 1 peer-unit 2
set logical-systems A interfaces lt-0/1/0 unit 1 family inet address 10.10.10.1/30
set logical-systems A interfaces lo0 unit 1 family inet address 192.168.6.5/32
set logical-systems A protocols bgp group internal-peers type internal
set logical-systems A protocols bgp group internal-peers local-address 192.168.6.5
set logical-systems A protocols bgp group internal-peers export send-direct
set logical-systems A protocols bgp group internal-peers neighbor 192.163.6.4
set logical-systems A protocols bgp group internal-peers neighbor 192.168.40.4
set logical-systems A protocols ospf area 0.0.0.0 interface lo0.1 passive
set logical-systems A protocols ospf area 0.0.0.0 interface lt-0/1/0.1
set logical-systems A policy-options policy-statement send-direct term 2 from protocol direct
set logical-systems A policy-options policy-statement send-direct term 2 then accept
set logical-systems A routing-options router-id 192.168.6.5
set logical-systems A routing-options autonomous-system 17
set logical-systems B interfaces lt-0/1/0 unit 2 description to-A
set logical-systems B interfaces lt-0/1/0 unit 2 encapsulation ethernet
set logical-systems B interfaces lt-0/1/0 unit 2 peer-unit 1
set logical-systems B interfaces lt-0/1/0 unit 2 family inet address 10.10.10.2/30
set logical-systems B interfaces lt-0/1/0 unit 5 description to-C
set logical-systems B interfaces lt-0/1/0 unit 5 encapsulation ethernet
set logical-systems B interfaces lt-0/1/0 unit 5 peer-unit 6
set logical-systems B interfaces lt-0/1/0 unit 5 family inet address 10.10.10.5/30
set logical-systems B interfaces lo0 unit 2 family inet address 192.163.6.4/32
set logical-systems B protocols bgp group internal-peers type internal
set logical-systems B protocols bgp group internal-peers local-address 192.163.6.4
set logical-systems B protocols bgp group internal-peers export send-direct
set logical-systems B protocols bgp group internal-peers neighbor 192.168.40.4
set logical-systems B protocols bgp group internal-peers neighbor 192.168.6.5
set logical-systems B protocols ospf area 0.0.0.0 interface lo0.2 passive
set logical-systems B protocols ospf area 0.0.0.0 interface lt-0/1/0.2
set logical-systems B protocols ospf area 0.0.0.0 interface lt-0/1/0.5
set logical-systems B policy-options policy-statement send-direct term 2 from protocol direct
set logical-systems B policy-options policy-statement send-direct term 2 then accept
set logical-systems B routing-options router-id 192.163.6.4
```

```
set logical-systems B routing-options autonomous-system 17
set logical-systems C interfaces lt-0/1/0 unit 6 description to-B
set logical-systems C interfaces lt-0/1/0 unit 6 encapsulation ethernet
set logical-systems C interfaces lt-0/1/0 unit 6 peer-unit 5
set logical-systems C interfaces lt-0/1/0 unit 6 family inet address 10.10.10.6/30
set logical-systems C interfaces lo0 unit 3 family inet address 192.168.40.4/32
set logical-systems C protocols bgp group internal-peers type internal
set logical-systems C protocols bgp group internal-peers local-address 192.168.40.4
set logical-systems C protocols bgp group internal-peers export send-direct
set logical-systems C protocols bgp group internal-peers neighbor 192.163.6.4
set logical-systems C protocols bgp group internal-peers neighbor 192.168.6.5
set logical-systems C protocols ospf area 0.0.0.0 interface lo0.3 passive
set logical-systems C protocols ospf area 0.0.0.0 interface lt-0/1/0.6
set logical-systems C policy-options policy-statement send-direct term 2 from protocol direct
set logical-systems C policy-options policy-statement send-direct term 2 then accept
set logical-systems C routing-options router-id 192.168.40.4
set logical-systems C routing-options autonomous-system 17
```

Device A

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure internal BGP peer sessions on Device A:

1. Configure the interfaces.

```
[edit logical-systems A interfaces lt-0/1/0 unit 1]
user@R1# set description to-B
user@R1# set encapsulation ethernet
user@R1# set peer-unit 2
user@R1# set family inet address 10.10.10.1/30
user@R1# set family inet address 192.168.6.5/32
user@R1# up
user@R1# up
[edit  logical-systems A interfaces]
user@R1# set lo0 unit 1 family inet address 192.168.6.5/32
user@R1# exit
[edit]
```

```
user@R1# edit logical-systems B interfaces lt-0/1/0
[edit logical-systems B interfaces lt-0/1/0]
user@R1# set unit 2 description to-A
user@R1# set unit 2 encapsulation ethernet
user@R1# set unit 2 peer-unit 1
user@R1# set unit 2 family inet address 10.10.10.2/30
user@R1# set unit 5 description to-C
user@R1# set unit 5 encapsulation ethernet
user@R1# set unit 5 peer-unit 6
user@R1# set family inet address 10.10.10.5/30
user@R1# up
[edit logical-systems B interfaces]
user@R1# set lo0 unit 2 family inet address 192.163.6.4/32
user@R1# exit
[edit]
user@R1# edit logical-systems C interfaces lt-0/1/0 unit 6
[edit logical-systems C interfaces lt-0/1/0 unit 6]
set description to-B
set encapsulation ethernet
set peer-unit 5
set family inet address 10.10.10.6/30
user@R1# up
user@R1# up
[edit  logical-systems C interfaces]
set lo0 unit 3 family inet address 192.168.40.4/32
```

2. Configure BGP.

   On Logical System A, the `neighbor` statements are included for both Device B and Device C, even though Logical System A is not directly connected to Device C.

```
[edit logical-systems A protocols bgp group internal-peers]
user@R1# set type internal
user@R1# set local-address 192.168.6.5
user@R1# set export send-direct
user@R1# set neighbor 192.163.6.4
user@R1# set neighbor 192.168.40.4
[edit logical-systems B protocols bgp group internal-peers]
user@R1# set type internal
user@R1# set local-address 192.163.6.4
user@R1# set export send-direct
user@R1# set neighbor 192.168.40.4
```

```
user@R1# set neighbor 192.168.6.5
[edit logical-systems C protocols bgp group internal-peers]
user@R1# set type internal
user@R1# set local-address 192.168.40.4
user@R1# set export send-direct
user@R1# set neighbor 192.163.6.4
user@R1# set neighbor 192.168.6.5
```

3. Configure OSPF.

```
[edit logical-systems A protocols ospf area 0.0.0.0]
user@R1# set interface lo0.1 passive
user@R1# set interface lt-0/1/0.1
[edit logical-systems A protocols ospf area 0.0.0.0]
user@R1# set interface lo0.2 passive
user@R1# set interface lt-0/1/0.2
user@R1# set interface lt-0/1/0.5
[edit logical-systems A protocols ospf area 0.0.0.0]
user@R1# set interface lo0.3 passive
user@R1# set interface lt-0/1/0.6
```

4. Configure a policy that accepts direct routes.

   Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit logical-systems A policy-options policy-statement send-direct term 2]
user@R1# set from protocol direct
user@R1# set then accept
[edit logical-systems B policy-options policy-statement send-direct term 2]
user@R1# set from protocol direct
user@R1# set then accept
[edit logical-systems C policy-options policy-statement send-direct term 2]
user@R1# set from protocol direct
user@R1# set then accept
```

5. Configure the router ID and the autonomous system (AS) number.

```
[edit logical-systems A routing-options]
user@R1# set router-id 192.168.6.5
```

```
user@R1# set autonomous-system 17
[edit logical-systems B routing-options]
user@R1# set router-id 192.163.6.4
user@R1# set autonomous-system 17
[edit logical-systems C routing-options]
user@R1# set router-id 192.168.40.4
user@R1# set autonomous-system 17
```

Results

From configuration mode, confirm your configuration by entering the `show logical-systems` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@R1# show logical-systems
A {
    interfaces {
        lt-0/1/0 {
            unit 1 {
                description to-B;
                encapsulation ethernet;
                peer-unit 2;
                family inet {
                    address 10.10.10.1/30;
                }
            }
        }
        lo0 {
            unit 1 {
                family inet {
                    address 192.168.6.5/32;
                }
            }
        }
    }
    protocols {
        bgp {
            group internal-peers {
                type internal;
                local-address 192.168.6.5;
                export send-direct;
```

```
                neighbor 192.163.6.4;
                neighbor 192.168.40.4;
            }
        }
        ospf {
            area 0.0.0.0 {
                interface lo0.1 {
                    passive;
                }
                interface lt-0/1/0.1;
            }
        }
    }
    policy-options {
        policy-statement send-direct {
            term 2 {
                from protocol direct;
                then accept;
            }
        }
    }
    routing-options {
        router-id 192.168.6.5;
        autonomous-system 17;
    }
}
B {
    interfaces {
        lt-0/1/0 {
            unit 2 {
                description to-A;
                encapsulation ethernet;
                peer-unit 1;
                family inet {
                    address 10.10.10.2/30;
                }
            }
            unit 5 {
                description to-C;
                encapsulation ethernet;
                peer-unit 6;
                family inet {
                    address 10.10.10.5/30;
```

```
            }
        }
    }
    lo0 {
        unit 2 {
            family inet {
                address 192.163.6.4/32;
            }
        }
    }
}
protocols {
    bgp {
        group internal-peers {
            type internal;
            local-address 192.163.6.4;
            export send-direct;
            neighbor 192.168.40.4;
            neighbor 192.168.6.5;
        }
    }
    ospf {
        area 0.0.0.0 {
            interface lo0.2 {
                passive;
            }
            interface lt-0/1/0.2;
            interface lt-0/1/0.5;
        }
    }
}
policy-options {
    policy-statement send-direct {
        term 2 {
            from protocol direct;
            then accept;
        }
    }
}
routing-options {
    router-id 192.163.6.4;
    autonomous-system 17;
}
```

```
}
C {
    interfaces {
        lt-0/1/0 {
            unit 6 {
                description to-B;
                encapsulation ethernet;
                peer-unit 5;
                family inet {
                    address 10.10.10.6/30;
                }
            }
        }
        lo0 {
            unit 3 {
                family inet {
                    address 192.168.40.4/32;
                }
            }
        }
    }
    protocols {
        bgp {
            group internal-peers {
                type internal;
                local-address 192.168.40.4;
                export send-direct;
                neighbor 192.163.6.4;
                neighbor 192.168.6.5;
            }
        }
        ospf {
            area 0.0.0.0 {
                interface lo0.3 {
                    passive;
                }
                interface lt-0/1/0.6;
            }
        }
    }
    policy-options {
        policy-statement send-direct {
            term 2 {
```

```
                from protocol direct;
                then accept;
            }
        }
    }
    routing-options {
        router-id 192.168.40.4;
        autonomous-system 17;
    }
}
```

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

Verification

IN THIS SECTION

- Verifying BGP Neighbors | **210**
- Verifying BGP Groups | **212**
- Verifying BGP Summary Information | **213**
- Verifying That BGP Routes Are Installed in the Routing Table | **213**

Confirm that the configuration is working properly.

Verifying BGP Neighbors

Purpose

Verify that BGP is running on configured interfaces and that the BGP session is active for each neighbor address.

Action

From the operational mode, enter the `show bgp neighbor` command.

```
user@R1> show bgp neighbor logical-system A
Peer: 192.163.6.4+179 AS 17    Local: 192.168.6.5+58852 AS 17
  Type: Internal    State: Established    Flags: <Sync>
```

```
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ send-direct ]
  Options: <Preference LocalAddress Refresh>
  Local Address: 192.168.6.5 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 192.163.6.4     Local ID: 192.168.6.5       Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 0
  BFD: disabled, down
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-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-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 17)
  Peer does not support Addpath
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            3
    Accepted prefixes:            3
    Suppressed due to damping:    0
    Advertised prefixes:          2
  Last traffic (seconds): Received 16   Sent 1    Checked 63
  Input messages:  Total 15713  Updates 4       Refreshes 0     Octets 298622
  Output messages: Total 15690  Updates 2       Refreshes 0     Octets 298222
  Output Queue[0]: 0

Peer: 192.168.40.4+179 AS 17   Local: 192.168.6.5+56466 AS 17
  Type: Internal    State: Established    Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ send-direct ]
  Options: <Preference LocalAddress Refresh>
  Local Address: 192.168.6.5 Holdtime: 90 Preference: 170
  Number of flaps: 0
```

```
  Peer ID: 192.168.40.4    Local ID: 192.168.6.5       Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 1
  BFD: disabled, down
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-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-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 17)
  Peer does not support Addpath
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            2
    Accepted prefixes:            2
    Suppressed due to damping:    0
    Advertised prefixes:          2
  Last traffic (seconds): Received 15   Sent 22   Checked 68
  Input messages:  Total 15688  Updates 2       Refreshes 0     Octets 298111
  Output messages: Total 15688  Updates 2       Refreshes 0     Octets 298184
  Output Queue[0]: 0
```

Verifying BGP Groups

Purpose

Verify that the BGP groups are configured correctly.

Action

From the operational mode, enter the `show bgp group` command.

```
user@A> show bgp group logical-system A
Group Type: Internal    AS: 17                    Local AS: 17
```

```
  Name: internal-peers  Index: 0                   Flags: <Export Eval>
  Export: [ send-direct ]
  Holdtime: 0
  Total peers: 2        Established: 2
  192.163.6.4+179
  192.168.40.4+179
  inet.0: 0/5/5/0

Groups: 1  Peers: 2    External: 0    Internal: 2    Down peers: 0   Flaps: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 5          0          0          0          0          0
```

Verifying BGP Summary Information

Purpose

Verify that the BGP configuration is correct.

Action

From the operational mode, enter the `show bgp summary` command.

```
user@A> show bgp summary logical-system A
Groups: 1 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 5          0          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
192.163.6.4              17      15723      15700       0       0 4d 22:13:15
0/3/3/0              0/0/0/0
192.168.40.4             17      15698      15699       0       0 4d 22:13:11
0/2/2/0              0/0/0/0
```

Verifying That BGP Routes Are Installed in the Routing Table

Purpose

Verify that the export policy configuration is working.

Action

From the operational mode, enter the `show route protocol bgp` command.

```
user@A> show route protocol bgp logical-system A
inet.0: 7 destinations, 12 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.10.0/30       [BGP/170] 4d 11:05:55, localpref 100, from 192.163.6.4
                      AS path: I
                    > to 10.10.10.2 via lt-0/1/0.1
10.10.10.4/30       [BGP/170] 4d 11:05:55, localpref 100, from 192.163.6.4
                      AS path: I
                    > to 10.10.10.2 via lt-0/1/0.1
                    [BGP/170] 4d 11:03:10, localpref 100, from 192.168.40.4
                      AS path: I
                    > to 10.10.10.2 via lt-0/1/0.1
192.163.6.4/32      [BGP/170] 4d 11:05:55, localpref 100, from 192.163.6.4
                      AS path: I
                    > to 10.10.10.2 via lt-0/1/0.1
192.168.40.4/32     [BGP/170] 4d 11:03:10, localpref 100, from 192.168.40.4
                      AS path: I
                    > to 10.10.10.2 via lt-0/1/0.1
```

SEE ALSO

- Understanding Internal BGP Peering Sessions
- BGP Configuration Overview

Example: Configuring External BGP on Logical Systems with IPv6 Interfaces

IN THIS SECTION

Requirements | 215

- Overview | 215
- Configuration | 218
- Verification | 230

This example shows how to configure external BGP (EBGP) point-to-point peer sessions on logical systems with IPv6 interfaces.

Requirements

In this example, no special configuration beyond device initialization is required.

Overview

IN THIS SECTION

- Topology | 217

Junos OS supports EBGP peer sessions by means of IPv6 addresses. An IPv6 peer session can be configured when an IPv6 address is specified in the `neighbor` statement. This example uses EUI-64 to generate IPv6 addresses that are automatically applied to the interfaces. An EUI-64 address is an IPv6 address that uses the IEEE EUI-64 format for the interface identifier portion of the address (the last 64 bits).

NOTE: Alternatively, you can configure EBGP sessions using manually assigned 128-bit IPv6 addresses.

If you use 128-bit link-local addresses for the interfaces, you must include the `local-interface` statement. This statement is valid only for 128-bit IPv6 link-local addresses and is mandatory for configuring an IPv6 EBGP link-local peer session.

Configuring EBGP peering using link-local addresses is only applicable for directly connected interfaces. There is no support for multihop peering.

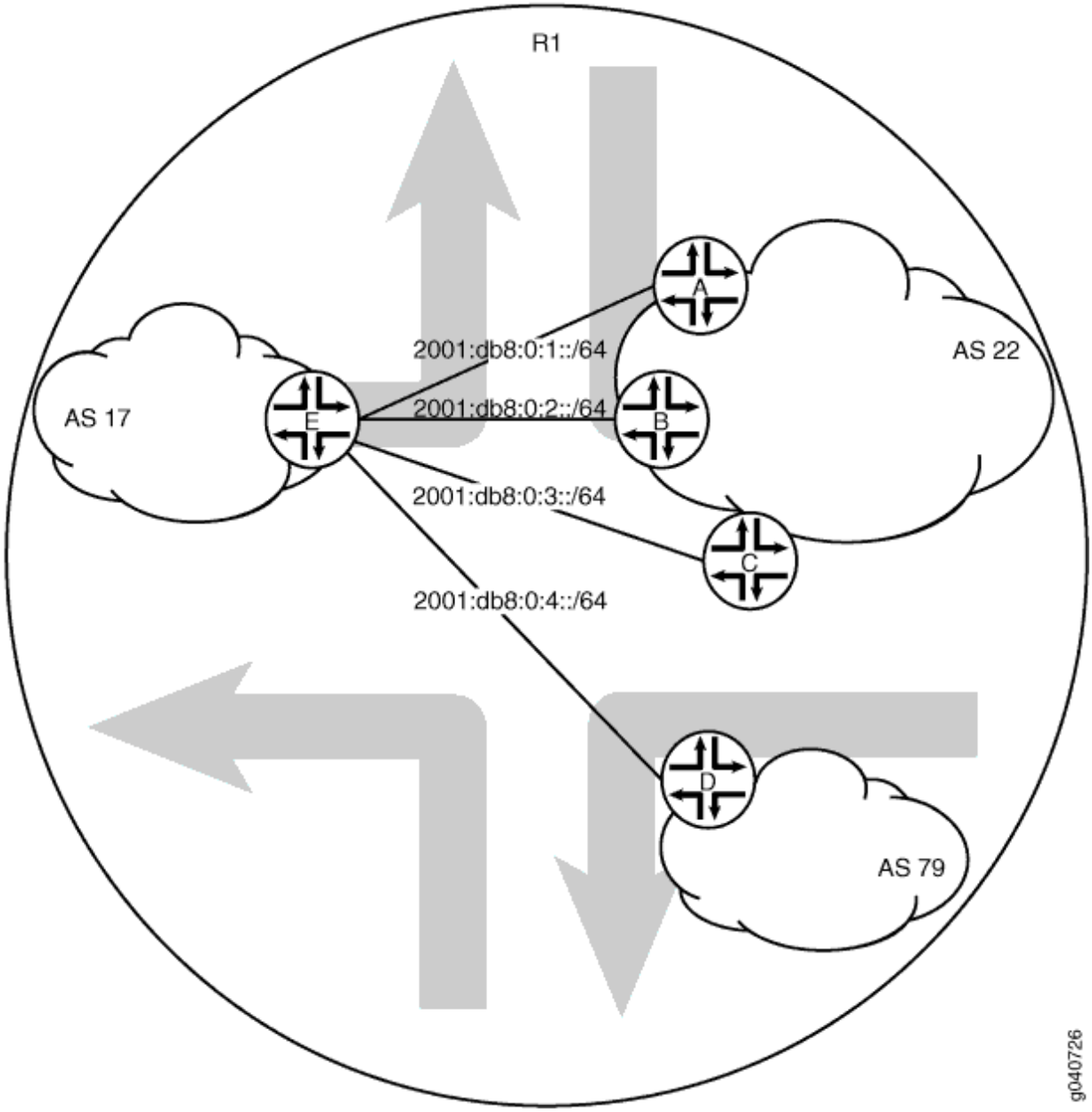
After your interfaces are up, you can use the `show interfaces terse` command to view the EUI-64-generated IPv6 addresses on the interfaces. You must use these generated addresses in the BGP `neighbor` statements. This example demonstrates the full end-to-end procedure.

In this example, Frame Relay interface encapsulation is applied to the logical tunnel (**lt**) interfaces. This is a requirement because only Frame Relay encapsulation is supported when IPv6 addresses are configured on the **lt** interfaces.

Figure 16 on page 217 shows a network with BGP peer sessions. In the sample network, Router R1 has five logical systems configured. Device E in autonomous system (AS) 17 has BGP peer sessions to a group of peers called **external-peers**. Peers A, B, and C reside in AS 22. This example shows the step-by-step configuration on Logical System A and Logical System E.

Topology

Figure 16: Typical Network with BGP Peer Sessions



Configuration

IN THIS SECTION

- Procedure | 218
- Configuring the External BGP Sessions | 223

Procedure

CLI Quick Configuration

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

Device A

```
set logical-systems A interfaces lt-0/1/0 unit 1 description to-E
set logical-systems A interfaces lt-0/1/0 unit 1 encapsulation frame-relay
set logical-systems A interfaces lt-0/1/0 unit 1 dlci 1
set logical-systems A interfaces lt-0/1/0 unit 1 peer-unit 25
set logical-systems A interfaces lt-0/1/0 unit 1 family inet6 address 2001:db8:0:1::/64 eui-64
set logical-systems A interfaces lo0 unit 1 family inet6 address 2001:db8::1/128
set logical-systems A protocols bgp group external-peers type external
set logical-systems A protocols bgp group external-peers peer-as 17
set logical-systems A protocols bgp group external-peers neighbor 2001:db8:0:1:2a0:a502:0:19da
set logical-systems A routing-options router-id 172.16.1.1
set logical-systems A routing-options autonomous-system 22
```

Device B

```
set logical-systems B interfaces lt-0/1/0 unit 6 description to-E
set logical-systems B interfaces lt-0/1/0 unit 6 encapsulation frame-relay
set logical-systems B interfaces lt-0/1/0 unit 6 dlci 6
set logical-systems B interfaces lt-0/1/0 unit 6 peer-unit 5
set logical-systems B interfaces lt-0/1/0 unit 6 family inet6 address 2001:db8:0:2::/64 eui-64
set logical-systems B interfaces lo0 unit 2 family inet6 address 2001:db8::2/128
set logical-systems B protocols bgp group external-peers type external
```

```
set logical-systems B protocols bgp group external-peers peer-as 17
set logical-systems B protocols bgp group external-peers neighbor 2001:db8:0:2:2a0:a502:0:5da
set logical-systems B routing-options router-id 172.16.2.2
set logical-systems B routing-options autonomous-system 22
```

Device C

```
set logical-systems C interfaces lt-0/1/0 unit 10 description to-E
set logical-systems C interfaces lt-0/1/0 unit 10 encapsulation frame-relay
set logical-systems C interfaces lt-0/1/0 unit 10 dlci 10
set logical-systems C interfaces lt-0/1/0 unit 10 peer-unit 9
set logical-systems C interfaces lt-0/1/0 unit 10 family inet6 address 2001:db8:0:3::/64 eui-64
set logical-systems C interfaces lo0 unit 3 family inet6 address 2001:db8::3/128
set logical-systems C protocols bgp group external-peers type external
set logical-systems C protocols bgp group external-peers peer-as 17
set logical-systems C protocols bgp group external-peers neighbor 2001:db8:0:3:2a0:a502:0:9da
set logical-systems C routing-options router-id 172.16.3.3
set logical-systems C routing-options autonomous-system 22
```

Device D

```
set logical-systems D interfaces lt-0/1/0 unit 7 description to-E
set logical-systems D interfaces lt-0/1/0 unit 7 encapsulation frame-relay
set logical-systems D interfaces lt-0/1/0 unit 7 dlci 7
set logical-systems D interfaces lt-0/1/0 unit 7 peer-unit 21
set logical-systems D interfaces lt-0/1/0 unit 7 family inet6 address 2001:db8:0:4::/64 eui-64
set logical-systems D interfaces lo0 unit 4 family inet6 address 2001:db8::4/128
set logical-systems D protocols bgp group external-peers type external
set logical-systems D protocols bgp group external-peers peer-as 17
set logical-systems D protocols bgp group external-peers neighbor 2001:db8:0:4:2a0:a502:0:15da
set logical-systems D routing-options router-id 172.16.4.4
set logical-systems D routing-options autonomous-system 79
```

Device E

```
set logical-systems E interfaces lt-0/1/0 unit 5 description to-B
set logical-systems E interfaces lt-0/1/0 unit 5 encapsulation frame-relay
set logical-systems E interfaces lt-0/1/0 unit 5 dlci 6
set logical-systems E interfaces lt-0/1/0 unit 5 peer-unit 6
set logical-systems E interfaces lt-0/1/0 unit 5 family inet6 address 2001:db8:0:2::/64 eui-64
set logical-systems E interfaces lt-0/1/0 unit 9 description to-C
```

```
set logical-systems E interfaces lt-0/1/0 unit 9 encapsulation frame-relay
set logical-systems E interfaces lt-0/1/0 unit 9 dlci 10
set logical-systems E interfaces lt-0/1/0 unit 9 peer-unit 10
set logical-systems E interfaces lt-0/1/0 unit 9 family inet6 address 2001:db8:0:3::/64 eui-64
set logical-systems E interfaces lt-0/1/0 unit 21 description to-D
set logical-systems E interfaces lt-0/1/0 unit 21 encapsulation frame-relay
set logical-systems E interfaces lt-0/1/0 unit 21 dlci 7
set logical-systems E interfaces lt-0/1/0 unit 21 peer-unit 7
set logical-systems E interfaces lt-0/1/0 unit 21 family inet6 address 2001:db8:0:4::/64 eui-64
set logical-systems E interfaces lt-0/1/0 unit 25 description to-A
set logical-systems E interfaces lt-0/1/0 unit 25 encapsulation frame-relay
set logical-systems E interfaces lt-0/1/0 unit 25 dlci 1
set logical-systems E interfaces lt-0/1/0 unit 25 peer-unit 1
set logical-systems E interfaces lt-0/1/0 unit 25 family inet6 address 2001:db8:0:1::/64 eui-64
set logical-systems E interfaces lo0 unit 5 family inet6 address 2001:db8::5/128
set logical-systems E protocols bgp group external-peers type external
set logical-systems E protocols bgp group external-peers peer-as 22
set logical-systems E protocols bgp group external-peers neighbor 2001:db8:0:1:2a0:a502:0:1da
set logical-systems E protocols bgp group external-peers neighbor 2001:db8:0:2:2a0:a502:0:6da
set logical-systems E protocols bgp group external-peers neighbor 2001:db8:0:3:2a0:a502:0:ada
set logical-systems E protocols bgp group external-peers neighbor 2001:db8:0:4:2a0:a502:0:7da
peer-as 79
set logical-systems E routing-options router-id 172.16.5.5
set logical-systems E routing-options autonomous-system 17
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the BGP peer sessions:

1. Run the `show interfaces terse` command to verify that the physical router has a logical tunnel (**lt**) interface.

```
user@R1> show interfaces terse
Interface               Admin Link Proto    Local                 Remote
...
lt-0/1/0                up    up
...
```

2. On Logical System A, configure the interface encapsulation, peer-unit number, and DLCI to reach Logical System E.

```
user@R1> set cli logical-system A
Logical system: A
[edit]
user@R1:A> edit
Entering configuration mode
[edit]
user@R1:A# edit interfaces
[edit interfaces]
user@R1:A# set lt-0/1/0 unit 1 encapsulation frame-relay
user@R1:A# set lt-0/1/0 unit 1 dlci 1
user@R1:A# set lt-0/1/0 unit 1 peer-unit 25
```

3. On Logical System A, configure the network address for the link to Peer E, and configure a loopback interface.

```
[edit interfaces]
user@R1:A# set lt-0/1/0 unit 1 description to-E
user@R1:A# set lt-0/1/0 unit 1 family inet6 address 2001:db8:0:1::/64 eui-64
user@R1:A# set lo0 unit 1 family inet6 address 2001:db8::1/128
```

4. On Logical System E, configure the interface encapsulation, peer-unit number, and DLCI to reach Logical System A.

```
user@R1> set cli logical-system E
Logical system: E
[edit]
user@R1:E> edit
Entering configuration mode
[edit]
user@R1:E# edit interfaces
[edit interfaces]
user@R1:E# set lt-0/1/0 unit 25 encapsulation frame-relay
user@R1:E# set lt-0/1/0 unit 25 dlci 1
user@R1:E# set lt-0/1/0 unit 25 peer-unit 1
```

5. On Logical System E, configure the network address for the link to Peer A, and configure a loopback interface.

```
[edit interfaces]
user@R1:E# set lt-0/1/0 unit 25 description to-A
user@R1:E# set lt-0/1/0 unit 25 family inet6 address 2001:db8:0:1::/64 eui-64
user@R1:E# set lo0 unit 5 family inet6 address 2001:db8::5/128
```

6. Run the `show interfaces terse` command to see the IPv6 addresses that are generated by EUI-64.

   The 2001 addresses are used in this example in the BGP `neighbor` statements.

> **NOTE**: The fe80 addresses are link-local addresses and are not used in this example.

```
user@R1:A> show interfaces terse
Interface               Admin Link Proto    Local                 Remote
Logical system: A

betsy@tp8:A> show interfaces terse
Interface               Admin Link Proto    Local                 Remote
lt-0/1/0
lt-0/1/0.1              up    up   inet6    2001:db8:0:1:2a0:a502:0:1da/64
                                            fe80::2a0:a502:0:1da/64
lo0
lo0.1                   up    up   inet6    2001:db8::1
                                            fe80::2a0:a50f:fc56:1da
```

```
user@R1:E> show interfaces terse
Interface               Admin Link Proto    Local                 Remote
lt-0/1/0
lt-0/1/0.25             up    up   inet6    2001:db8:0:1:2a0:a502:0:19da/64
                                            fe80::2a0:a502:0:19da/64
lo0
lo0.5                   up    up   inet6    2001:db8::5
                                            fe80::2a0:a50f:fc56:1da
```

7. Repeat the interface configuration on the other logical systems.

Configuring the External BGP Sessions

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the BGP peer sessions:

1. On Logical System A, create the BGP group, and add the external neighbor address.

```
[edit protocols bgp group external-peers]
user@R1:A# set neighbor 2001:db8:0:1:2a0:a502:0:19da
```

2. On Logical System E, create the BGP group, and add the external neighbor address.

```
[edit protocols bgp group external-peers]
user@R1:E# set neighbor 2001:db8:0:1:2a0:a502:0:1da
```

3. On Logical System A, specify the autonomous system (AS) number of the external AS.

```
[edit protocols bgp group external-peers]
user@R1:A# set peer-as 17
```

4. On Logical System E, specify the autonomous system (AS) number of the external AS.

```
[edit protocols bgp group external-peers]
user@R1:E# set peer-as 22
```

5. On Logical System A, set the peer type to EBGP.

```
[edit protocols bgp group external-peers]
user@R1:A# set type external
```

6. On Logical System E, set the peer type to EBGP.

```
[edit protocols bgp group external-peers]
user@R1:E# set type external
```

7. On Logical System A, set the autonomous system (AS) number and router ID.

```
[edit routing-options]
user@R1:A# set router-id 172.16.1.1
user@R1:A# set autonomous-system 22
```

8. On Logical System E, set the AS number and router ID.

```
[edit routing-options]
user@R1:E# set router-id 172.16.5.5
user@R1:E# set autonomous-system 17
```

9. Repeat these steps for Peers A, B, C, and D.

Results

From configuration mode, confirm your configuration by entering the `show logical-systems` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@R1# show logical-systems
A {
    interfaces {
        lt-0/1/0 {
            unit 1 {
                description to-E;
                encapsulation frame-relay;
                dlci 1;
                peer-unit 25;
                family inet6 {
                    address 2001:db8:0:1::/64 {
                        eui-64;
                    }
```

```
            }
        }
    }
    lo0 {
        unit 1 {
            family inet6 {
                address 2001:db8::1/128;
            }
        }
    }
}
protocols {
    bgp {
        group external-peers {
            type external;
            peer-as 17;
            neighbor 2001:db8:0:1:2a0:a502:0:19da;
        }
    }
    routing-options {
        router-id 172.16.1.1;
        autonomous-system 22;
    }
}
B {
    interfaces {
        lt-0/1/0 {
            unit 6 {
                description to-E;
                encapsulation frame-relay;
                dlci 6;
                peer-unit 5;
                family inet6 {
                    address 2001:db8:0:2::/64 {
                        eui-64;
                    }
                }
            }
        }
        lo0 {
            unit 2 {
                family inet6 {
                    address 2001:db8::2/128;
```

```
                }
            }
        }
    }
    protocols {
        bgp {
            group external-peers {
                type external;
                peer-as 17;
                neighbor 2001:db8:0:2:2a0:a502:0:5da;
            }
        }
        routing-options {
            router-id 172.16.2.2;
            autonomous-system 22;
        }
    }
    C {
        interfaces {
            lt-0/1/0 {
                unit 10 {
                    description to-E;
                    encapsulation frame-relay;
                    dlci 10;
                    peer-unit 9;
                    family inet6 {
                        address 2001:db8:0:3::/64 {
                            eui-64;
                        }
                    }
                }
            }
            lo0 {
                unit 3 {
                    family inet6 {
                        address 2001:db8::3/128;
                    }
                }
            }
        }
        protocols {
            bgp {
                group external-peers {
```

```
                    type external;
                    peer-as 17;
                    neighbor 2001:db8:0:3:2a0:a502:0:9da;
                }
            }
        }
        routing-options {
            router-id 172.16.3.3;
            autonomous-system 22;
        }
    }
    D {
        interfaces {
            lt-0/1/0 {
                unit 7 {
                    description to-E;
                    encapsulation frame-relay;
                    dlci 7;
                    peer-unit 21;
                    family inet6 {
                        address 2001:db8:0:4::/64 {
                            eui-64;
                        }
                    }
                }
            }
            lo0 {
                unit 4 {
                    family inet6 {
                        address 2001:db8::4/128;
                    }
                }
            }
        }
        protocols {
            bgp {
                group external-peers {
                    type external;
                    peer-as 17;
                    neighbor 2001:db8:0:4:2a0:a502:0:15da;
                }
            }
            routing-options {
```

```
                router-id 172.16.4.4;
                autonomous-system 79;
            }
        }
        E {
            interfaces {
                lt-0/1/0 {
                    unit 5 {
                        description to-B;
                        encapsulation frame-relay;
                        dlci 6;
                        peer-unit 6;
                        family inet6 {
                            address 2001:db8:0:2::/64 {
                                eui-64;
                            }
                        }
                    }
                    unit 9 {
                        description to-C;
                        encapsulation frame-relay;
                        dlci 10;
                        peer-unit 10;
                        family inet6 {
                            address 2001:db8:0:3::/64 {
                                eui-64;
                            }
                        }
                    }
                    unit 21 {
                        description to-D;
                        encapsulation frame-relay;
                        dlci 7;
                        peer-unit 7;
                        family inet6 {
                            address 2001:db8:0:4::/64 {
                                eui-64;
                            }
                        }
                    }
                    unit 25 {
                        description to-A;
                        encapsulation frame-relay;
```

```
                        dlci 1;
                        peer-unit 1;
                        family inet6 {
                            address 2001:db8:0:1::/64 {
                                eui-64;
                            }
                        }
                    }
                }
                lo0 {
                    unit 5 {
                        family inet6 {
                            address 2001:db8::5/128;
                        }
                    }
                }
            }
            protocols {
                bgp {
                    group external-peers {
                        type external;
                        peer-as 22;
                        neighbor 2001:db8:0:1:2a0:a502:0:1da;
                        neighbor 2001:db8:0:2:2a0:a502:0:6da;
                        neighbor 2001:db8:0:3:2a0:a502:0:ada;
                        neighbor 2001:db8:0:4:2a0:a502:0:7da {
                            peer-as 79;
                        }
                    }
                }
            }
            routing-options {
                router-id 172.16.5.5;
                autonomous-system 17;
            }
        }
```

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

Verification

IN THIS SECTION

- Verifying BGP Neighbors | 230
- Verifying BGP Groups | 234
- Verifying BGP Summary Information | 234
- Checking the Routing Table | 235

Confirm that the configuration is working properly.

Verifying BGP Neighbors

Purpose

Verify that BGP is running on configured interfaces and that the BGP session is active for each neighbor address.

Action

From operational mode, run the `show bgp neighbor` command.

```
user@R1:E> show bgp neighbor
Peer: 2001:db8:0:1:2a0:a502:0:1da+54987 AS 22 Local: 2001:db8:0:1:2a0:a502:0:19da+179 AS 17
  Type: External    State: Established    Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: Open Message Error
  Options: <Preference PeerAS Refresh>
  Holdtime: 90 Preference: 170
  Number of flaps: 0
  Error: 'Open Message Error' Sent: 20 Recv: 0
  Peer ID: 172.16.1.1         Local ID: 172.16.5.5          Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 0
  BFD: disabled, down
  Local Interface: lt-0/1/0.25
  NLRI for restart configured on peer: inet6-unicast
  NLRI advertised by peer: inet6-unicast
  NLRI for this session: inet6-unicast
```

```
  Peer supports Refresh capability (2)
  Stale routes from peer are kept for: 300
  Peer does not support Restarter functionality
  NLRI that restart is negotiated for: inet6-unicast
  NLRI of received end-of-rib markers: inet6-unicast
  NLRI of all end-of-rib markers sent: inet6-unicast
  Peer supports 4 byte AS extension (peer-as 22)
  Peer does not support Addpath
  Table inet6.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            0
    Accepted prefixes:            0
    Suppressed due to damping:    0
    Advertised prefixes:          0
  Last traffic (seconds): Received 7    Sent 18   Checked 81
  Input messages:  Total 1611   Updates 1       Refreshes 0     Octets 30660
  Output messages: Total 1594   Updates 0       Refreshes 0     Octets 30356
  Output Queue[0]: 0

Peer: 2001:db8:0:2:2a0:a502:0:6da+179 AS 22 Local: 2001:db8:0:2:2a0:a502:0:5da+55502 AS 17
  Type: External    State: Established    Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: Open Message Error
  Options: <Preference PeerAS Refresh>
  Holdtime: 90 Preference: 170
  Number of flaps: 0
  Error: 'Open Message Error' Sent: 26 Recv: 0
  Peer ID: 172.16.2.2         Local ID: 172.16.5.5           Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 2
  BFD: disabled, down
  Local Interface: lt-0/1/0.5
  NLRI for restart configured on peer: inet6-unicast
  NLRI advertised by peer: inet6-unicast
  NLRI for this session: inet6-unicast
  Peer supports Refresh capability (2)
  Stale routes from peer are kept for: 300
  Peer does not support Restarter functionality
  NLRI that restart is negotiated for: inet6-unicast
  NLRI of received end-of-rib markers: inet6-unicast
  NLRI of all end-of-rib markers sent: inet6-unicast
  Peer supports 4 byte AS extension (peer-as 22)
```

```
  Peer does not support Addpath
  Table inet6.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            0
    Accepted prefixes:            0
    Suppressed due to damping:    0
    Advertised prefixes:          0
  Last traffic (seconds): Received 15   Sent 8    Checked 8
  Input messages:  Total 1610   Updates 1       Refreshes 0     Octets 30601
  Output messages: Total 1645   Updates 0       Refreshes 0     Octets 32417
  Output Queue[0]: 0

Peer: 2001:db8:0:3:2a0:a502:0:ada+55983 AS 22 Local: 2001:db8:0:3:2a0:a502:0:9da+179 AS 17
  Type: External    State: Established    Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference PeerAS Refresh>
  Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 172.16.3.3         Local ID: 172.16.5.5           Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 3
  BFD: disabled, down
  Local Interface: lt-0/1/0.9
  NLRI for restart configured on peer: inet6-unicast
  NLRI advertised by peer: inet6-unicast
  NLRI for this session: inet6-unicast
  Peer supports Refresh capability (2)
  Stale routes from peer are kept for: 300
  Peer does not support Restarter functionality
  NLRI that restart is negotiated for: inet6-unicast
  NLRI of received end-of-rib markers: inet6-unicast
  NLRI of all end-of-rib markers sent: inet6-unicast
  Peer supports 4 byte AS extension (peer-as 22)
  Peer does not support Addpath
  Table inet6.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            0
    Accepted prefixes:            0
    Suppressed due to damping:    0
```

```
    Advertised prefixes:          0
  Last traffic (seconds): Received 21   Sent 21   Checked 67
  Input messages:  Total 1610   Updates 1       Refreshes 0     Octets 30641
  Output messages: Total 1587   Updates 0       Refreshes 0     Octets 30223
  Output Queue[0]: 0

Peer: 2001:db8:0:4:2a0:a502:0:7da+49255 AS 79 Local: 2001:db8:0:4:2a0:a502:0:15da+179 AS 17
  Type: External    State: Established    Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference PeerAS Refresh>
  Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 172.16.4.4         Local ID: 172.16.5.5           Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 1
  BFD: disabled, down
  Local Interface: lt-0/1/0.21
  NLRI for restart configured on peer: inet6-unicast
  NLRI advertised by peer: inet6-unicast
  NLRI for this session: inet6-unicast
  Peer supports Refresh capability (2)
  Stale routes from peer are kept for: 300
  Peer does not support Restarter functionality
  NLRI that restart is negotiated for: inet6-unicast
  NLRI of received end-of-rib markers: inet6-unicast
  NLRI of all end-of-rib markers sent: inet6-unicast
  Peer supports 4 byte AS extension (peer-as 79)
  Peer does not support Addpath
  Table inet6.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            0
    Accepted prefixes:            0
    Suppressed due to damping:    0
    Advertised prefixes:          0
  Last traffic (seconds): Received 6    Sent 17   Checked 25
  Input messages:  Total 1615   Updates 1       Refreshes 0     Octets 30736
  Output messages: Total 1593   Updates 0       Refreshes 0     Octets 30337
  Output Queue[0]: 0
```

Meaning

IPv6 unicast network layer reachability information (NLRI) is being exchanged between the neighbors.

Verifying BGP Groups

Purpose

Verify that the BGP groups are configured correctly.

Action

From operational mode, run the `show bgp group` command.

```
user@R1:E> show bgp group
Group Type: External                              Local AS: 17
  Name: external-peers  Index: 0                  Flags: <>
  Holdtime: 0
  Total peers: 4        Established: 4
  2001:db8:0:1:2a0:a502:0:1da+54987
  2001:db8:0:2:2a0:a502:0:6da+179
  2001:db8:0:3:2a0:a502:0:ada+55983
  2001:db8:0:4:2a0:a502:0:7da+49255
  inet6.0: 0/0/0/0

Groups: 1  Peers: 4    External: 4    Internal: 0    Down peers: 0   Flaps: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet6.0                0          0          0          0          0          0
inet6.2                0          0          0          0          0          0
```

Meaning

The group type is external, and the group has four peers.

Verifying BGP Summary Information

Purpose

Verify that the BGP peer relationships are established.

Action

From operational mode, run the `show bgp summary` command.

```
user@R1:E> show bgp summary
Groups: 1 Peers: 4 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet6.0                0          0          0          0          0          0
inet6.2                0          0          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
2001:db8:0:1:2a0:a502:0:1da          22       1617       1600       0       0    12:07:00 Establ
  inet6.0: 0/0/0/0
2001:db8:0:2:2a0:a502:0:6da          22       1616       1651       0       0    12:06:56 Establ
  inet6.0: 0/0/0/0
2001:db8:0:3:2a0:a502:0:ada          22       1617       1594       0       0    12:04:32 Establ
  inet6.0: 0/0/0/0
2001:db8:0:4:2a0:a502:0:7da          79       1621       1599       0       0    12:07:00 Establ
  inet6.0: 0/0/0/0
```

Meaning

The Down peers: 0 output shows that the BGP peers are in the established state.

Checking the Routing Table

Purpose

Verify that the inet6.0 routing table is populated with local and direct routes.

Action

From operational mode, run the `show route` command.

```
user@R1:E> show route
inet6.0: 15 destinations, 18 routes (15 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2001:db8::5/128    *[Direct/0] 12:41:18
                    > via lo0.5
```

```
2001:db8:0:1::/64  *[Direct/0] 14:40:01
                    > via lt-0/1/0.25
2001:db8:0:1:2a0:a502:0:19da/128
                   *[Local/0] 14:40:01
                      Local via lt-0/1/0.25
2001:db8:0:2::/64  *[Direct/0] 14:40:02
                    > via lt-0/1/0.5
2001:db8:0:2:2a0:a502:0:5da/128
                   *[Local/0] 14:40:02
                      Local via lt-0/1/0.5
2001:db8:0:3::/64  *[Direct/0] 14:40:02
                    > via lt-0/1/0.9
2001:db8:0:3:2a0:a502:0:9da/128
                   *[Local/0] 14:40:02
                      Local via lt-0/1/0.9
2001:db8:0:4::/64  *[Direct/0] 14:40:01
                    > via lt-0/1/0.21
2001:db8:0:4:2a0:a502:0:15da/128
                   *[Local/0] 14:40:01
                      Local via lt-0/1/0.21
fe80::/64          *[Direct/0] 14:40:02
                    > via lt-0/1/0.5
                    [Direct/0] 14:40:02
                    > via lt-0/1/0.9
                    [Direct/0] 14:40:01
                    > via lt-0/1/0.21
                    [Direct/0] 14:40:01
                    > via lt-0/1/0.25
fe80::2a0:a502:0:5da/128
                   *[Local/0] 14:40:02
                      Local via lt-0/1/0.5
fe80::2a0:a502:0:9da/128
                   *[Local/0] 14:40:02
                      Local via lt-0/1/0.9
fe80::2a0:a502:0:15da/128
                   *[Local/0] 14:40:01
                      Local via lt-0/1/0.21
fe80::2a0:a502:0:19da/128
                   *[Local/0] 14:40:01
                      Local via lt-0/1/0.25
fe80::2a0:a50f:fc56:1da/128
```

```
                    *[Direct/0] 12:41:18
                     > via lo0.5
```

Meaning

The inet6.0 routing table contains local and direct routes. To populate the routing table with other types of routes, you must configure routing policies.

SEE ALSO

- Understanding External BGP Peering Sessions

Example: Configuring BFD on Internal BGP Peer Sessions

IN THIS SECTION

- Requirements | 237
- Overview | 237
- Configuration | 239
- Verification | 245

This example shows how to configure internal BGP (IBGP) peer sessions with the Bidirectional Forwarding Detection (BFD) protocol to detect failures in a network.

Requirements

No special configuration beyond device initialization is required before you configure this example.

Overview

The minimum configuration to enable BFD on IBGP sessions is to include the `bfd-liveness-detection minimum-interval` statement in the BGP configuration of all neighbors participating in the BFD session. The `minimum-interval` statement specifies the minimum transmit and receive intervals for failure detection. Specifically, this value represents the minimum interval after which the local routing device transmits hello packets as well as the minimum interval that the routing device expects to receive a reply from a

neighbor with which it has established a BFD session. You can configure a value from 1 through 255,000 milliseconds.

Optionally, you can specify the minimum transmit and receive intervals separately using the `transmit-interval minimum-interval` and `minimum-receive-interval` statements. For information about these and other optional BFD configuration statements, see `bfd-liveness-detection`.

NOTE: BFD is an intensive protocol that consumes system resources. Specifying a minimum interval for BFD less than 100 milliseconds for Routing Engine-based sessions and less than 10 milliseconds for distributed BFD sessions can cause undesired BFD flapping.

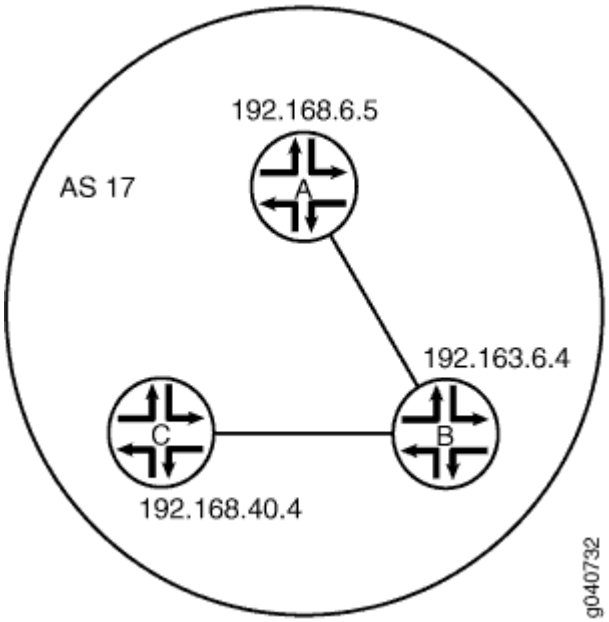
Depending on your network environment, these additional recommendations might apply:

- To prevent BFD flapping during the general Routing Engine switchover event, specify a minimum interval of 5000 milliseconds for Routing Engine-based sessions. This minimum value is required because, during the general Routing Engine switchover event, processes such as RPD, MIBD, and SNMPD utilize CPU resources for more than the specified threshold value. Hence, BFD processing and scheduling is affected because of this lack of CPU resources.
- For BFD sessions to remain up during the dual chassis cluster control link scenario, when the first control link fails, specify the minimum interval of 6000 milliseconds to prevent the LACP from flapping on the secondary node for Routing Engine-based sessions.
- For large-scale network deployments with a large number of BFD sessions, specify a minimum interval of 300 milliseconds for Routing Engine-based sessions and 100 milliseconds for distributed BFD sessions.
- For very large-scale network deployments with a large number of BFD sessions, contact Juniper Networks customer support for more information.
- For BFD sessions to remain up during a Routing Engine switchover event when nonstop active routing (NSR) is configured, specify a minimum interval of 2500 milliseconds for Routing Engine-based sessions. For distributed BFD sessions with NSR configured, the minimum interval recommendations are unchanged and depend only on your network deployment.

BFD is supported on the default routing instance (the main router), routing instances, and logical systems. This example shows BFD on logical systems.

Figure 17 on page 239 shows a typical network with internal peer sessions.

Figure 17: Typical Network with IBGP Sessions



Configuration

IN THIS SECTION

- CLI Quick Configuration | 239
- Configuring Device A | 241

CLI Quick Configuration

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

Device A

```
set logical-systems A interfaces lt-1/2/0 unit 1 description to-B
set logical-systems A interfaces lt-1/2/0 unit 1 encapsulation ethernet
```

```
set logical-systems A interfaces lt-1/2/0 unit 1 peer-unit 2
set logical-systems A interfaces lt-1/2/0 unit 1 family inet address 10.10.10.1/30
set logical-systems A interfaces lo0 unit 1 family inet address 192.168.6.5/32
set logical-systems A protocols bgp group internal-peers type internal
set logical-systems A protocols bgp group internal-peers traceoptions file bgp-bfd
set logical-systems A protocols bgp group internal-peers traceoptions flag bfd detail
set logical-systems A protocols bgp group internal-peers local-address 192.168.6.5
set logical-systems A protocols bgp group internal-peers export send-direct
set logical-systems A protocols bgp group internal-peers bfd-liveness-detection minimum-interval
1000
set logical-systems A protocols bgp group internal-peers neighbor 192.163.6.4
set logical-systems A protocols bgp group internal-peers neighbor 192.168.40.4
set logical-systems A protocols ospf area 0.0.0.0 interface lo0.1 passive
set logical-systems A protocols ospf area 0.0.0.0 interface lt-1/2/0.1
set logical-systems A policy-options policy-statement send-direct term 2 from protocol direct
set logical-systems A policy-options policy-statement send-direct term 2 then accept
set logical-systems A routing-options router-id 192.168.6.5
set logical-systems A routing-options autonomous-system 17
```

Device B

```
set logical-systems B interfaces lt-1/2/0 unit 2 description to-A
set logical-systems B interfaces lt-1/2/0 unit 2 encapsulation ethernet
set logical-systems B interfaces lt-1/2/0 unit 2 peer-unit 1
set logical-systems B interfaces lt-1/2/0 unit 2 family inet address 10.10.10.2/30
set logical-systems B interfaces lt-1/2/0 unit 5 description to-C
set logical-systems B interfaces lt-1/2/0 unit 5 encapsulation ethernet
set logical-systems B interfaces lt-1/2/0 unit 5 peer-unit 6
set logical-systems B interfaces lt-1/2/0 unit 5 family inet address 10.10.10.5/30
set logical-systems B interfaces lo0 unit 2 family inet address 192.163.6.4/32
set logical-systems B protocols bgp group internal-peers type internal
set logical-systems B protocols bgp group internal-peers local-address 192.163.6.4
set logical-systems B protocols bgp group internal-peers export send-direct
set logical-systems B protocols bgp group internal-peers bfd-liveness-detection minimum-interval
1000
set logical-systems B protocols bgp group internal-peers neighbor 192.168.40.4
set logical-systems B protocols bgp group internal-peers neighbor 192.168.6.5
set logical-systems B protocols ospf area 0.0.0.0 interface lo0.2 passive
set logical-systems B protocols ospf area 0.0.0.0 interface lt-1/2/0.2
set logical-systems B protocols ospf area 0.0.0.0 interface lt-1/2/0.5
set logical-systems B policy-options policy-statement send-direct term 2 from protocol direct
set logical-systems B policy-options policy-statement send-direct term 2 then accept
```

```
set logical-systems B routing-options router-id 192.163.6.4
set logical-systems B routing-options autonomous-system 17
```

Device C

```
set logical-systems C interfaces lt-1/2/0 unit 6 description to-B
set logical-systems C interfaces lt-1/2/0 unit 6 encapsulation ethernet
set logical-systems C interfaces lt-1/2/0 unit 6 peer-unit 5
set logical-systems C interfaces lt-1/2/0 unit 6 family inet address 10.10.10.6/30
set logical-systems C interfaces lo0 unit 3 family inet address 192.168.40.4/32
set logical-systems C protocols bgp group internal-peers type internal
set logical-systems C protocols bgp group internal-peers local-address 192.168.40.4
set logical-systems C protocols bgp group internal-peers export send-direct
set logical-systems C protocols bgp group internal-peers bfd-liveness-detection minimum-interval
1000
set logical-systems C protocols bgp group internal-peers neighbor 192.163.6.4
set logical-systems C protocols bgp group internal-peers neighbor 192.168.6.5
set logical-systems C protocols ospf area 0.0.0.0 interface lo0.3 passive
set logical-systems C protocols ospf area 0.0.0.0 interface lt-1/2/0.6
set logical-systems C policy-options policy-statement send-direct term 2 from protocol direct
set logical-systems C policy-options policy-statement send-direct term 2 then accept
set logical-systems C routing-options router-id 192.168.40.4
set logical-systems C routing-options autonomous-system 17
```

Configuring Device A

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure Device A:

1. Set the CLI to Logical System A.

```
user@host> set cli logical-system A
```

2. Configure the interfaces.

```
[edit interfaces lt-1/2/0 unit 1]
user@host:A# set description to-B
user@host:A# set encapsulation ethernet
user@host:A# set peer-unit 2
user@host:A# set family inet address 10.10.10.1/30
[edit interfaces lo0 unit 1]
user@host:A# set family inet address 192.168.6.5/32
```

3. Configure BGP.

   The `neighbor` statements are included for both Device B and Device C, even though Device A is not directly connected to Device C.

```
[edit protocols bgp group internal-peers]
user@host:A# set type internal
user@host:A# set local-address 192.168.6.5
user@host:A# set export send-direct
user@host:A# set neighbor 192.163.6.4
user@host:A# set neighbor 192.168.40.4
```

4. Configure BFD.

```
[edit protocols bgp group internal-peers]
user@host:A# set bfd-liveness-detection minimum-interval 1000
```

   You must configure the same minimum interval on the connecting peer.

5. (Optional) Configure BFD tracing.

```
[edit protocols bgp group internal-peers]
user@host:A# set traceoptions file bgp-bfd
user@host:A# set traceoptions flag bfd detail
```

6. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@host:A# set interface lo0.1 passive
user@host:A# set interface lt-1/2/0.1
```

7. Configure a policy that accepts direct routes.

   Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-direct term 2]
user@host:A# set from protocol direct
user@host:A# set then accept
```

8. Configure the router ID and the autonomous system (AS) number.

```
[edit routing-options]
user@host:A# set router-id 192.168.6.5
user@host:A# set autonomous-system 17
```

9. If you are done configuring the device, enter `commit` from configuration mode.
   Repeat these steps to configure Device B and Device C.

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show policy-options`, `show protocols`, and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host:A# show interfaces
lt-1/2/0 {
    unit 1 {
        description to-B;
        encapsulation ethernet;
        peer-unit 2;
        family inet {
            address 10.10.10.1/30;
        }
```

```
    }
}
lo0 {
    unit 1 {
        family inet {
            address 192.168.6.5/32;
        }
    }
}
```

```
user@host:A# show policy-options
policy-statement send-direct {
    term 2 {
        from protocol direct;
        then accept;
    }
}
```

```
user@host:A# show protocols
bgp {
    group internal-peers {
        type internal;
        traceoptions {
            file bgp-bfd;
            flag bfd detail;
        }
        local-address 192.168.6.5;
        export send-direct;
        bfd-liveness-detection {
            minimum-interval 1000;
        }
        neighbor 192.163.6.4;
        neighbor 192.168.40.4;
    }
}
ospf {
    area 0.0.0.0 {
        interface lo0.1 {
            passive;
        }
```

```
        interface lt-1/2/0.1;
    }
}
```

```
user@host:A# show routing-options
router-id 192.168.6.5;
autonomous-system 17;
```

Verification

IN THIS SECTION

- Verifying That BFD Is Enabled | 245
- Verifying That BFD Sessions Are Up | 246
- Viewing Detailed BFD Events | 247
- Viewing Detailed BFD Events After Deactivating and Reactivating a Loopback Interface | 248

Confirm that the configuration is working properly.

Verifying That BFD Is Enabled

Purpose

Verify that BFD is enabled between the IBGP peers.

Action

From operational mode, enter the `show bgp neighbor` command. You can use the `| match bfd` filter to narrow the output.

```
user@host:A> show bgp neighbor | match bfd
Options: <BfdEnabled>
  BFD: enabled, up
  Trace file: /var/log/A/bgp-bfd size 131072 files 10
  Options: <BfdEnabled>
```

```
  BFD: enabled, up
  Trace file: /var/log/A/bgp-bfd size 131072 files 10
```

Meaning

The output shows that Logical System A has two neighbors with BFD enabled. When BFD is not enabled, the output displays `BFD: disabled, down`, and the `<BfdEnabled>` option is absent. If BFD is enabled and the session is down, the output displays `BFD: enabled, down`. The output also shows that BFD-related events are being written to a log file because trace operations are configured.

Verifying That BFD Sessions Are Up

Purpose

Verify that the BFD sessions are up, and view details about the BFD sessions.

Action

From operational mode, enter the `show bfd session extensive` command.

```
user@host:A> show bfd session extensive
                                                Detect   Transmit
Address                 State     Interface     Time     Interval  Multiplier
192.163.6.4             Up                      3.000     1.000        3
 Client BGP, TX interval 1.000, RX interval 1.000
 Session up time 00:54:40
 Local diagnostic None, remote diagnostic None
 Remote state Up, version 1
 Logical system 12, routing table index 25
 Min async interval 1.000, min slow interval 1.000
 Adaptive async TX interval 1.000, RX interval 1.000
 Local min TX interval 1.000, minimum RX interval 1.000, multiplier 3
 Remote min TX interval 1.000, min RX interval 1.000, multiplier 3
 Local discriminator 10, remote discriminator 9
 Echo mode disabled/inactive
 Multi-hop route table 25, local-address 192.168.6.5

                                                Detect   Transmit
Address                 State     Interface     Time     Interval  Multiplier
192.168.40.4            Up                      3.000     1.000        3
 Client BGP, TX interval 1.000, RX interval 1.000
```

```
  Session up time 00:48:03
  Local diagnostic None, remote diagnostic None
  Remote state Up, version 1
  Logical system 12, routing table index 25
  Min async interval 1.000, min slow interval 1.000
  Adaptive async TX interval 1.000, RX interval 1.000
  Local min TX interval 1.000, minimum RX interval 1.000, multiplier 3
  Remote min TX interval 1.000, min RX interval 1.000, multiplier 3
  Local discriminator 14, remote discriminator 13
  Echo mode disabled/inactive
  Multi-hop route table 25, local-address 192.168.6.5

2 sessions, 2 clients
Cumulative transmit rate 2.0 pps, cumulative receive rate 2.0 pps
```

Meaning

The `TX interval 1.000, RX interval 1.000` output represents the setting configured with the `minimum-interval` statement. All of the other output represents the default settings for BFD. To modify the default settings, include the optional statements under the `bfd-liveness-detection` statement.

Viewing Detailed BFD Events

Purpose

View the contents of the BFD trace file to assist in troubleshooting, if needed.

Action

From operational mode, enter the `file show /var/log/A/bgp-bfd` command.

```
user@host:A> file show /var/log/A/bgp-bfd
Aug 15 17:07:25 trace_on: Tracing to "/var/log/A/bgp-bfd" started
Aug 15 17:07:26.492190 bgp_peer_init: BGP peer 192.163.6.4 (Internal AS 17) local address
192.168.6.5 not found.  Leaving peer idled
Aug 15 17:07:26.493176 bgp_peer_init: BGP peer 192.168.40.4 (Internal AS 17) local address
192.168.6.5 not found.  Leaving peer idled
Aug 15 17:07:32.597979 task_connect: task BGP_17.192.163.6.4+179 addr 192.163.6.4+179: No route
to host
Aug 15 17:07:32.599623 bgp_connect_start: connect 192.163.6.4 (Internal AS 17): No route to host
Aug 15 17:07:36.869394 task_connect: task BGP_17.192.168.40.4+179 addr 192.168.40.4+179: No
```

```
route to host
Aug 15 17:07:36.870624 bgp_connect_start: connect 192.168.40.4 (Internal AS 17): No route to host
Aug 15 17:08:04.599220 task_connect: task BGP_17.192.163.6.4+179 addr 192.163.6.4+179: No route
to host
Aug 15 17:08:04.601135 bgp_connect_start: connect 192.163.6.4 (Internal AS 17): No route to host
Aug 15 17:08:08.869717 task_connect: task BGP_17.192.168.40.4+179 addr 192.168.40.4+179: No
route to host
Aug 15 17:08:08.869934 bgp_connect_start: connect 192.168.40.4 (Internal AS 17): No route to host
Aug 15 17:08:36.603544 advertising receiving-speaker only capabilty to neighbor 192.163.6.4
(Internal AS 17)
Aug 15 17:08:36.606726 bgp_read_message: 192.163.6.4 (Internal AS 17): 0 bytes buffered
Aug 15 17:08:36.609119 Initiated BFD session to peer 192.163.6.4 (Internal AS 17):
address=192.163.6.4 ifindex=0 ifname=(none) txivl=1000 rxivl=1000 mult=3 ver=255
Aug 15 17:08:36.734033 advertising receiving-speaker only capabilty to neighbor 192.168.40.4
(Internal AS 17)
Aug 15 17:08:36.738436 Initiated BFD session to peer 192.168.40.4 (Internal AS 17):
address=192.168.40.4 ifindex=0 ifname=(none) txivl=1000 rxivl=1000 mult=3 ver=255
Aug 15 17:08:40.537552 BFD session to peer 192.163.6.4 (Internal AS 17) up
Aug 15 17:08:40.694410 BFD session to peer 192.168.40.4 (Internal AS 17) up
```

Meaning

Before the routes are established, the `No route to host` message appears in the output. After the routes are established, the last two lines show that both BFD sessions come up.

Viewing Detailed BFD Events After Deactivating and Reactivating a Loopback Interface

Purpose

Check to see what happens after bringing down a router or switch and then bringing it back up. To simulate bringing down a router or switch, deactivate the loopback interface on Logical System B.

Action

1. From configuration mode, enter the `deactivate logical-systems B interfaces lo0 unit 2 family inet` command.

```
user@host:A# deactivate logical-systems B interfaces lo0 unit 2 family inet
user@host:A# commit
```

2. From operational mode, enter the `file show /var/log/A/bgp-bfd` command.

```
user@host:A> file show /var/log/A/bgp-bfd
...
Aug 15 17:20:55.995648 bgp_read_v4_message:9747: NOTIFICATION received from 192.163.6.4
(Internal AS 17): code 6 (Cease) subcode 6 (Other Configuration Change)
Aug 15 17:20:56.004508 Terminated BFD session to peer 192.163.6.4 (Internal AS 17)
Aug 15 17:21:28.007755 task_connect: task BGP_17.192.163.6.4+179 addr 192.163.6.4+179: No
route to host
Aug 15 17:21:28.008597 bgp_connect_start: connect 192.163.6.4 (Internal AS 17): No route to
host
```

3. From configuration mode, enter the `activate logical-systems B interfaces lo0 unit 2 family inet` command.

```
user@host:A# activate logical-systems B interfaces lo0 unit 2 family inet
user@host:A# commit
```

4. From operational mode, enter the `file show /var/log/A/bgp-bfd` command.

```
user@host:A> file show /var/log/A/bgp-bfd
...
Aug 15 17:25:53.623743 advertising receiving-speaker only capabilty to neighbor 192.163.6.4
(Internal AS 17)
Aug 15 17:25:53.631314 Initiated BFD session to peer 192.163.6.4 (Internal AS 17):
address=192.163.6.4 ifindex=0 ifname=(none) txivl=1000 rxivl=1000 mult=3 ver=255
Aug 15 17:25:57.570932 BFD session to peer 192.163.6.4 (Internal AS 17) up
```

SEE ALSO

Example: Configuring BFD Authentication for BGP

Example: Configuring EBGP Multihop Sessions on Logical Systems

IN THIS SECTION

- Requirements | 250
- Overview | 250
- Configuration | 251
- Verification | 262

This example shows how to configure an external BGP (EBGP) peer that is more than one hop away from the local router. This type of session is called a *multihop* EBGP session.

Requirements

In this example, no special configuration beyond device initialization is required.

Overview

When EBGP peers are not directly connected to each other, they must cross one or more non-BGP routing devices to reach each other. Configuring multihop EBGP enables the peers to pass through the other routing devices to form peer relationships and exchange update messages. This type of configuration is typically used when a Juniper Networks routing device needs to run EBGP with a third-party routing device that does not allow direct connection of the two EBGP peers. EBGP multihop enables a neighbor connection between two EBGP peers that do not have a direct connection.

The configuration to enable multihop EBGP sessions requires connectivity between the two EBGP peers. This example uses static routes to provide connectivity between the devices.

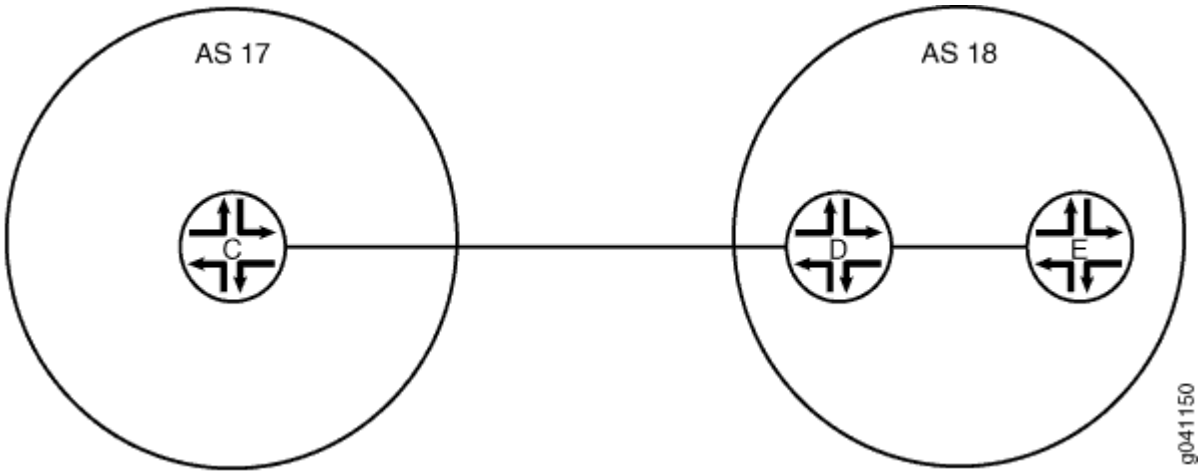
For directly connected EBGP sessions, physical addresses are typically used in the `neighbor` statements. For multihop EBGP, you must use loopback interface addresses, and specify the loopback interface address of the indirectly connected peer. In the use of loopback interfaces addresses, EBGP multihop is similar to internal BGP (IBGP).

Finally, you must add the `multihop` statement. Optionally, you can set a maximum time-to-live (TTL) value with the `ttl` statement. The TTL is carried in the IP header of BGP packets. If you do not specify a TTL value, the system's default maximum TTL value is used. The default TTL value is 64 for multihop EBGP sessions. Another option is to retain the BGP next-hop value for route advertisements by including the `no-nexthop-change` statement.

Figure 18 on page 251 shows a typical EBGP multihop network.

Device C and Device E have an established EBGP session. Device D is not a BGP-enabled device. All of the devices have connectivity via static routes.

Figure 18: Typical Network with EBGP Multihop Sessions



Configuration

IN THIS SECTION

- CLI Quick Configuration | **251**
- Device C | **253**
- Device D | **256**
- Device E | **259**

CLI Quick Configuration

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

Device C

```
set logical-systems C interfaces lt-1/2/0 unit 9 description to-D
set logical-systems C interfaces lt-1/2/0 unit 9 encapsulation ethernet
set logical-systems C interfaces lt-1/2/0 unit 9 peer-unit 10
```

```
set logical-systems C interfaces lt-1/2/0 unit 9 family inet address 10.10.10.9/30
set logical-systems C interfaces lo0 unit 3 family inet address 192.168.40.4/32
set logical-systems C protocols bgp group external-peers type external
set logical-systems C protocols bgp group external-peers multihop ttl 2
set logical-systems C protocols bgp group external-peers local-address 192.168.40.4
set logical-systems C protocols bgp group external-peers export send-static
set logical-systems C protocols bgp group external-peers peer-as 18
set logical-systems C protocols bgp group external-peers neighbor 192.168.6.7
set logical-systems C policy-options policy-statement send-static term 1 from protocol static
set logical-systems C policy-options policy-statement send-static term 1 then accept
set logical-systems C routing-options static route 10.10.10.14/32 next-hop 10.10.10.10
set logical-systems C routing-options static route 192.168.6.7/32 next-hop 10.10.10.10
set logical-systems C routing-options router-id 192.168.40.4
set logical-systems C routing-options autonomous-system 17
```

Device D

```
set logical-systems D interfaces lt-1/2/0 unit 10 description to-C
set logical-systems D interfaces lt-1/2/0 unit 10 encapsulation ethernet
set logical-systems D interfaces lt-1/2/0 unit 10 peer-unit 9
set logical-systems D interfaces lt-1/2/0 unit 10 family inet address 10.10.10.10/30
set logical-systems D interfaces lt-1/2/0 unit 13 description to-E
set logical-systems D interfaces lt-1/2/0 unit 13 encapsulation ethernet
set logical-systems D interfaces lt-1/2/0 unit 13 peer-unit 14
set logical-systems D interfaces lt-1/2/0 unit 13 family inet address 10.10.10.13/30
set logical-systems D interfaces lo0 unit 4 family inet address 192.168.6.6/32
set logical-systems D routing-options static route 192.168.40.4/32 next-hop 10.10.10.9
set logical-systems D routing-options static route 192.168.6.7/32 next-hop 10.10.10.14
set logical-systems D routing-options router-id 192.168.6.6
```

Device E

```
set logical-systems E interfaces lt-1/2/0 unit 14 description to-D
set logical-systems E interfaces lt-1/2/0 unit 14 encapsulation ethernet
set logical-systems E interfaces lt-1/2/0 unit 14 peer-unit 13
set logical-systems E interfaces lt-1/2/0 unit 14 family inet address 10.10.10.14/30
set logical-systems E interfaces lo0 unit 5 family inet address 192.168.6.7/32
set logical-systems E protocols bgp group external-peers multihop ttl 2
set logical-systems E protocols bgp group external-peers local-address 192.168.6.7
set logical-systems E protocols bgp group external-peers export send-static
set logical-systems E protocols bgp group external-peers peer-as 17
```

```
set logical-systems E protocols bgp group external-peers neighbor 192.168.40.4
set logical-systems E policy-options policy-statement send-static term 1 from protocol static
set logical-systems E policy-options policy-statement send-static term 1 then accept
set logical-systems E routing-options static route 10.10.10.8/30 next-hop 10.10.10.13
set logical-systems E routing-options static route 192.168.40.4/32 next-hop 10.10.10.13
set logical-systems E routing-options router-id 192.168.6.7
set logical-systems E routing-options autonomous-system 18
```

Device C

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure Device C:

1. Set the CLI to Logical System C.

```
user@host> set cli logical-system C
```

2. Configure the interface to the directly-connected device (to-D), and configure the loopback interface.

```
[edit interfaces lt-1/2/0 unit 9]
user@host:C# set description to-D
user@host:C# set encapsulation ethernet
user@host:C# set peer-unit 10
user@host:C# set family inet address 10.10.10.9/30
[edit interfaces lo0 unit 3]
user@host:C# set family inet address 192.168.40.4/32
```

3. Configure an EBGP session with Logical System E.

   The `neighbor` statement points to the loopback interface on Logical System E.

```
[edit protocols bgp group external-peers]
user@host:C# set type external
user@host:C# set local-address 192.168.40.4
user@host:C# set export send-static
```

```
user@host:C# set peer-as 18
user@host:C# set neighbor 192.168.6.7
```

4. Configure the multihop statement to enable Logical System C and Logical System E to become EBGP peers.

   Because the peers are two hops away from each other, the example uses the `ttl 2` statement.

```
[edit protocols bgp group external-peers]
user@host:C# set multihop ttl 2
```

5. Configure connectivity to Logical System E, using static routes.

   You must configure a route to both the loopback interface address and to the address on the physical interface.

```
[edit  logical-systems C routing-options]
user@host:C# set static route 10.10.10.14/32 next-hop 10.10.10.10
user@host:C# set static route 192.168.6.7/32 next-hop 10.10.10.10
```

6. Configure the local router ID and the autonomous system (AS) number.

```
[edit routing-options]
user@host:C# set router-id 192.168.40.4
user@host:C# set autonomous-system 17
```

7. Configure a policy that accepts direct routes.

   Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-static term 1]
user@host:C# set from protocol static
user@host:C# set then accept
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols`, `show policy-options`, and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host:C# show interfaces
lt-1/2/0 {
    unit 9 {
        description to-D;
        encapsulation ethernet;
        peer-unit 10;
        family inet {
            address 10.10.10.9/30;
        }
    }
}
lo0 {
    unit 3 {
        family inet {
            address 192.168.40.4/32;
        }
    }
}
```

```
user@host:C# show protocols
bgp {
    group external-peers {
        type external;
        multihop {
            ttl 2;
        }
        local-address 192.168.40.4;
        export send-static;
        peer-as 18;
        neighbor 192.168.6.7;
```

```
    }
}
```

```
user@host:C# show policy-options
policy-statement send-static {
    term 1 {
        from protocol static;
        then accept;
    }
}
```

```
user@host:C# show routing-options
static {
    route 10.10.10.14/32 next-hop 10.10.10.10;
    route 192.168.6.7/32 next-hop 10.10.10.10;
}
router-id 192.168.40.4;
autonomous-system 17;
```

If you are done configuring the device, enter **commit** from configuration mode. Repeat these steps for all BFD sessions in the topology.

Device D

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure Device D:

1. Set the CLI to Logical System D.

```
user@host> set cli logical-system D
```

2. Configure the interfaces to the directly-connected devices, and configure a loopback interface.

```
[edit interfaces lt-1/2/0 unit 10]
user@host:D# set description to-C
user@host:D# set encapsulation ethernet
user@host:D# set peer-unit 9
user@host:D# set family inet address 10.10.10.10/30
[edit interfaces lt-1/2/0 unit 13]
user@host:D# set description to-E
user@host:D# set encapsulation ethernet
user@host:D# set peer-unit 14
user@host:D# set family inet address 10.10.10.13/30
[edit interfaces lo0 unit 4]
user@host:D# set family inet address 192.168.6.6/32
```

3. Configure connectivity to the other devices using a static routes to the loopback interface addresses.

   On Logical System D, you do not need static routes to the physical addresses because Logical System D is directly connected to Logical System C and Logical System E.

```
[edit routing-options]
user@host:D# set static route 192.168.40.4/32 next-hop 10.10.10.9
user@host:D# set static route 192.168.6.7/32 next-hop 10.10.10.14
```

4. Configure the local router ID and the autonomous system (AS) number.

```
[edit routing-options]
user@host:D# set router-id 192.168.6.6
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host:D# show interfaces
lt-1/2/0 {
    unit 10 {
        description to-C;
```

```
        encapsulation ethernet;
        peer-unit 9;
        family inet {
            address 10.10.10.10/30;
        }
    }
    unit 13 {
        description to-E;
        encapsulation ethernet;
        peer-unit 14;
        family inet {
            address 10.10.10.13/30;
        }
    }
}
lo0 {
    unit 4 {
        family inet {
            address 192.168.6.6/32;
        }
    }
}
```

```
user@host:D# show protocols
```

```
user@host:D# show routing-options
static {
    route 192.168.40.4/32 next-hop 10.10.10.9;
    route 192.168.6.7/32 next-hop 10.10.10.14;
}
router-id 192.168.6.6;
```

If you are done configuring the device, enter **commit** from configuration mode. Repeat these steps for all BFD sessions in the topology.

Device E

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure Device E:

1. Set the CLI to Logical System E.

```
user@host> set cli logical-system E
```

2. Configure the interface to the directly-connected device (to-D), and configure the loopback interface.

```
[edit interfaces lt-1/2/0 unit 14]
user@host:E# set description to-D
user@host:E# set encapsulation ethernet
user@host:E# set peer-unit 13
user@host:E# set family inet address 10.10.10.14/30
[edit interfaces lo0 unit 5]
user@host:E# set family inet address 192.168.6.7/32
```

3. Configure an EBGP session with Logical System E.

   The `neighbor` statement points to the loopback interface on Logical System C.

```
[edit protocols bgp group external-peers]
user@host:E# set local-address 192.168.6.7
user@host:E# set export send-static
user@host:E# set peer-as 17
user@host:E# set neighbor 192.168.40.4
```

4. Configure the `multihop` statement to enable Logical System C and Logical System E to become EBGP peers.

Because the peers are two hops away from each other, the example uses the `ttl 2` statement.

```
[edit protocols bgp group external-peers]
user@host:E# set multihop ttl 2
```

5. Configure connectivity to Logical System E, using static routes.

   You must configure a route to both the loopback interface address and to the address on the physical interface.

```
[edit routing-options]
user@host:E# set static route 10.10.10.8/30 next-hop 10.10.10.13
user@host:E# set static route 192.168.40.4/32 next-hop 10.10.10.13
```

6. Configure the local router ID and the autonomous system (AS) number.

```
[edit routing-options]
user@host:E# set router-id 192.168.6.7
user@host:E# set autonomous-system 18
```

7. Configure a policy that accepts direct routes.

   Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-static term 1]
user@host:E# set from protocol static
user@host:E# set send-static then accept
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols`, `show policy-options`, and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host:E# show interfaces
lt-1/2/0 {
    unit 14 {
```

```
        description to-D;
        encapsulation ethernet;
        peer-unit 13;
        family inet {
            address 10.10.10.14/30;
        }
    }
}
lo0 {
    unit 5 {
        family inet {
            address 192.168.6.7/32;
        }
    }
}
```

```
user@host:E# show protocols
bgp {
    group external-peers {
        multihop {
            ttl 2;
        }
        local-address 192.168.6.7;
        export send-static;
        peer-as 17;
        neighbor 192.168.40.4;
    }
}
```

```
user@host:E# show policy-options
policy-statement send-static {
    term 1 {
        from protocol static;
        then accept;
    }
}
```

```
user@host:E# show routing-options
static {
```

```
    route 10.10.10.8/30 next-hop 10.10.10.13;
    route 192.168.40.4/32 next-hop 10.10.10.13;
}
router-id 192.168.6.7;
autonomous-system 18;
```

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

Verification

IN THIS SECTION

- Verifying Connectivity | **262**
- Verifying the BGP Sessions Are Established | **263**
- Viewing Advertised Routes | **264**

Confirm that the configuration is working properly.

Verifying Connectivity

Purpose

Make sure that Device C can ping Device E, specifying the loopback interface address as the source of the ping request.

The loopback interface address is the source address that BGP will be using.

Action

From operational mode, enter the `ping 10.10.10.14 source 192.168.40.4` command from Logical System C, and enter the `ping 10.10.10.9 source 192.168.6.7` command from Logical System E.

```
user@host:C> ping 10.10.10.14 source 192.168.40.4PING 10.10.10.14 (10.10.10.14): 56 data bytes
64 bytes from 10.10.10.14: icmp_seq=0 ttl=63 time=1.262 ms
64 bytes from 10.10.10.14: icmp_seq=1 ttl=63 time=1.202 ms
^C
--- 10.10.10.14 ping statistics ---
```

```
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.202/1.232/1.262/0.030 ms
```

```
user@host:E> ping 10.10.10.9 source 192.168.6.7PING 10.10.10.9 (10.10.10.9): 56 data bytes
64 bytes from 10.10.10.9: icmp_seq=0 ttl=63 time=1.255 ms
64 bytes from 10.10.10.9: icmp_seq=1 ttl=63 time=1.158 ms
^C
--- 10.10.10.9 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.158/1.206/1.255/0.049 ms
```

Meaning

The static routes are working if the pings work.

Verifying the BGP Sessions Are Established

Purpose

Verify that the BGP sessions are up.

Action

From operational mode, enter the `show bgp summary` command.

```
user@host:C> show bgp summaryGroups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 2          0          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
192.168.6.7              18        147        147       0       1     1:04:27
0/2/2/0              0/0/0/0
```

```
user@host:E> show bgp summary

Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 2          0          0          0          0          0
```

```
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
192.168.40.4             17        202        202       0       1     1:02:18
0/2/2/0              0/0/0/0
```

Meaning

The output shows that both devices have one peer each. No peers are down.

Viewing Advertised Routes

Purpose

Checking to make sure that routes are being advertised by BGP.

Action

From operational mode, enter the `show route advertising-protocol bgp` *neighbor* command.

```
user@host:C> show route advertising-protocol bgp 192.168.6.7inet.0: 5 destinations, 7 routes (5
active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 10.10.10.14/32          Self                                    I
* 192.168.6.7/32          Self                                    I
```

```
user@host:E> show route advertising-protocol bgp 192.168.40.4

inet.0: 5 destinations, 7 routes (5 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 10.10.10.8/30           Self                                    I
* 192.168.40.4/32         Self                                    I
```

Meaning

The **send-static** routing policy is exporting the static routes from the routing table into BGP. BGP is advertising these routes between the peers because the BGP peer session is established.

SEE ALSO

- Understanding BFD for BGP
- Routing Policies, Firewall Filters, and Traffic Policers User Guide
- Understanding External BGP Peering Sessions
- BGP Configuration Overview

RELATED DOCUMENTATION

- User Access on Logical Systems | 14
- BGP User Guide

IS-IS on Logical Systems

IN THIS SECTION

- Example: Configuring IS-IS on Logical Systems Within the Same Router | 265
- Example: Configuring an IS-IS Default Route Policy on Logical Systems | 280

Logical Systems enable you to configure an IS-IS network, running on a single physical router. For more information, see the following topics:

Example: Configuring IS-IS on Logical Systems Within the Same Router

IN THIS SECTION

- Requirements | 266
- Overview | 266
- Configuration | 267

- Verification | 275

This example shows how to configure an IS-IS network by using multiple logical systems that are running on a single physical router. The logical systems are connected by logical tunnel interfaces.

Requirements

You must connect the logical systems by using logical tunnel (lt) interfaces. See "Example: Connecting Logical Systems Within the Same Device Using Logical Tunnel Interfaces on MX Series Routers and EX Series Switches" on page 27.

Overview

IN THIS SECTION

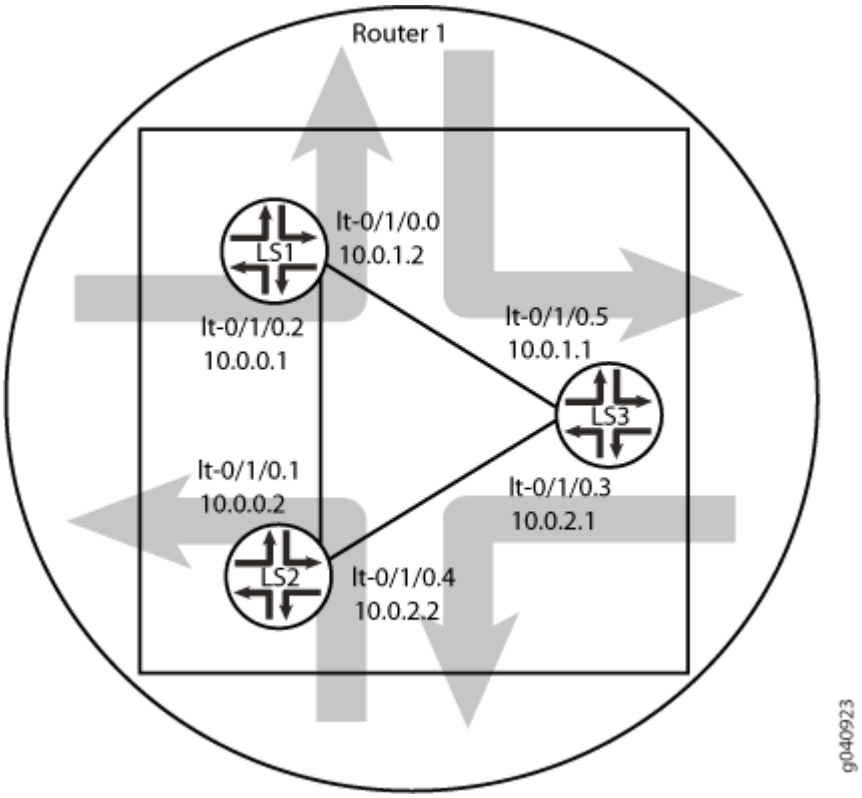
- Topology | 267

This example shows an IS-IS configuration with three logical systems running on one physical router. Each logical system has its own routing table. The configuration enables the protocol on all logical tunnel interfaces that participate in the IS-IS domain.

Topology

Figure 19 on page 267 shows the sample network.

Figure 19: IS-IS on Logical Systems



Configuration

IN THIS SECTION

- CLI Quick Configuration | **268**
- Procedure | **269**
- Results | **272**

CLI Quick Configuration

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

```
set logical-systems LS1 interfaces lt-0/1/0 unit 2 description LS1->LS2
set logical-systems LS1 interfaces lt-0/1/0 unit 2 encapsulation ethernet
set logical-systems LS1 interfaces lt-0/1/0 unit 2 peer-unit 1
set logical-systems LS1 interfaces lt-0/1/0 unit 2 family inet address 10.0.0.1/30
set logical-systems LS1 interfaces lt-0/1/0 unit 2 family iso
set logical-systems LS1 interfaces lt-0/1/0 unit 0 description LS1->LS3
set logical-systems LS1 interfaces lt-0/1/0 unit 0 encapsulation ethernet
set logical-systems LS1 interfaces lt-0/1/0 unit 0 peer-unit 5
set logical-systems LS1 interfaces lt-0/1/0 unit 0 family inet address 10.0.1.2/30
set logical-systems LS1 interfaces lt-0/1/0 unit 0 family iso
set logical-systems LS1 interfaces lo0 unit 1 family iso address 49.0001.1720.1600.1001.00
set logical-systems LS1 protocols isis interface lt-0/1/0.0
set logical-systems LS1 protocols isis interface lt-0/1/0.2
set logical-systems LS1 protocols isis interface lo0.1 passive
set logical-systems LS2 interfaces lt-0/1/0 unit 1 description LS2->LS1
set logical-systems LS2 interfaces lt-0/1/0 unit 1 encapsulation ethernet
set logical-systems LS2 interfaces lt-0/1/0 unit 1 peer-unit 2
set logical-systems LS2 interfaces lt-0/1/0 unit 1 family inet address 10.0.0.2/30
set logical-systems LS2 interfaces lt-0/1/0 unit 1 family iso
set logical-systems LS2 interfaces lt-0/1/0 unit 4 description LS2->LS3
set logical-systems LS2 interfaces lt-0/1/0 unit 4 encapsulation ethernet
set logical-systems LS2 interfaces lt-0/1/0 unit 4 peer-unit 3
set logical-systems LS2 interfaces lt-0/1/0 unit 4 family inet address 10.0.2.2/30
set logical-systems LS2 interfaces lt-0/1/0 unit 4 family iso
set logical-systems LS2 interfaces lo0 unit 2 family iso address 49.0001.1720.1600.2002.00
set logical-systems LS2 protocols isis interface lt-0/1/0.1
set logical-systems LS2 protocols isis interface lt-0/1/0.4
set logical-systems LS2 protocols isis interface lo0.2 passive
set logical-systems LS3 interfaces lt-0/1/0 unit 3 description LS3->LS2
set logical-systems LS3 interfaces lt-0/1/0 unit 3 encapsulation ethernet
set logical-systems LS3 interfaces lt-0/1/0 unit 3 peer-unit 4
set logical-systems LS3 interfaces lt-0/1/0 unit 3 family inet address 10.0.2.1/30
set logical-systems LS3 interfaces lt-0/1/0 unit 3 family iso
set logical-systems LS3 interfaces lt-0/1/0 unit 5 description LS3->LS1
set logical-systems LS3 interfaces lt-0/1/0 unit 5 encapsulation ethernet
set logical-systems LS3 interfaces lt-0/1/0 unit 5 peer-unit 0
```

```
set logical-systems LS3 interfaces lt-0/1/0 unit 5 family inet address 10.0.1.1/30
set logical-systems LS3 interfaces lt-0/1/0 unit 5 family iso
set logical-systems LS3 interfaces lo0 unit 3 family iso address 49.0001.1234.1600.2231.00
set logical-systems LS3 protocols isis interface lt-0/1/0.5
set logical-systems LS3 protocols isis interface lt-0/1/0.3
set logical-systems LS3 protocols isis interface lo0.3 passive
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure IS-IS on logical systems:

1. Configure the logical tunnel interface on Logical System LS1 connecting to Logical System LS2.

```
[edit logical-systems LS1]
user@host# set interfaces lt-0/1/0 unit 2 description LS1->LS2
user@host# set interfaces lt-0/1/0 unit 2 encapsulation ethernet
user@host# set interfaces lt-0/1/0 unit 2 peer-unit 1
user@host# set interfaces lt-0/1/0 unit 2 family inet address 10.0.0.1/30
user@host# set interfaces lt-0/1/0 unit 2 family iso
```

2. Configure the logical tunnel interface on Logical System LS1 connecting to Logical System LS3.

```
[edit logical-systems LS1]
user@host# set interfaces lt-0/1/0 unit 0 description LS1->LS3
user@host# set interfaces lt-0/1/0 unit 0 encapsulation ethernet
user@host# set interfaces lt-0/1/0 unit 0 peer-unit 5
user@host# set interfaces lt-0/1/0 unit 0 family inet address 10.0.1.2/30
user@host# set interfaces lt-0/1/0 unit 0 family iso
```

3. Configure the logical tunnel interface on Logical System LS2 connecting to Logical System LS1.

```
[edit logical-systems LS2]
user@host# set interfaces lt-0/1/0 unit 1 description LS2->LS1
user@host# set interfaces lt-0/1/0 unit 1 encapsulation ethernet
```

```
user@host# set interfaces lt-0/1/0 unit 1 peer-unit 2
user@host# set interfaces lt-0/1/0 unit 1 family inet address 10.0.0.2/30
user@host# set interfaces lt-0/1/0 unit 1 family iso
```

4. Configure the logical tunnel interface on Logical System LS2 connecting to Logical System LS3.

```
[edit logical-systems LS2]
user@host# set interfaces lt-0/1/0 unit 4 description LS2->LS3
user@host# set interfaces lt-0/1/0 unit 4 encapsulation ethernet
user@host# set interfaces lt-0/1/0 unit 4 peer-unit 3
user@host# set interfaces lt-0/1/0 unit 4 family inet address 10.0.2.2/30
user@host# set interfaces lt-0/1/0 unit 4 family iso
```

5. Configure the logical tunnel interface on Logical System LS3 connecting to Logical System LS2.

```
[edit logical-systems LS3]
user@host# set interfaces lt-0/1/0 unit 3 description LS3->LS2
user@host# set interfaces lt-0/1/0 unit 3 encapsulation ethernet
user@host# set interfaces lt-0/1/0 unit 3 peer-unit 4
user@host# set interfaces lt-0/1/0 unit 3 family inet address 10.0.2.1/30
user@host# set interfaces lt-0/1/0 unit 3 family iso
```

6. Configure the logical tunnel interface on Logical System LS3 connecting to Logical System LS1.

```
[edit logical-systems LS3]
user@host# set interfaces lt-0/1/0 unit 5 description LS3->LS1
user@host# set interfaces lt-0/1/0 unit 5 encapsulation ethernet
user@host# set interfaces lt-0/1/0 unit 5 peer-unit 0
user@host# set interfaces lt-0/1/0 unit 5 family inet address 10.0.1.1/30
user@host# set interfaces lt-0/1/0 unit 5 family iso
```

7. Configure the ISO address on the loopback interface for the three logical systems.

```
[edit logical-systems LS1]
user@host# set interfaces lo0 unit 1 family iso address 49.0001.1720.1600.1001.00
user@host# set protocols isis interface lo0.1 passive
```

```
[edit logical-systems LS2]
user@host# set interfaces lo0 unit 2 family iso address 49.0001.1720.1600.2002.00
user@host# set protocols isis interface lo0.2 passive
```

```
[edit logical-systems LS3]
user@host# set interfaces lo0 unit 3 family iso address 49.0001.1234.1600.2231.00
user@host# set protocols isis interface lo0.3 passive
```

8. Configure IS-IS on all the interfaces.

```
[edit logical-systems LS1 protocols isis]
user@host# set interface lt-0/1/0.0
user@host# set interface lt-0/1/0.2
```

```
[edit logical-systems LS2 protocols isis]
user@host# set interface lt-0/1/0.1
user@host# set interface lt-0/1/0.4
```

```
[edit logical-systems LS3 protocols isis]
user@host# set interface lt-0/1/0.5
user@host# set interface lt-0/1/0.3
```

9. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Results

From configuration mode, confirm your configuration by issuing the `show logical-systems` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show logical-systems
LS1 {
    interfaces {
        lt-0/1/0 {
            unit 0 {
                description LS1->LS3;
                encapsulation ethernet;
                peer-unit 5;
                family inet {
                    address 10.0.1.2/30;
                }
                family iso;
            }
            unit 2 {
                description LS1->LS2;
                encapsulation ethernet;
                peer-unit 1;
                family inet {
                    address 10.0.0.1/30;
                }
                family iso;
            }
        }
        lo0 {
            unit 1 {
                family iso {
                    address 49.0001.1720.1600.1001.00;
                }
            }
        }
    }
    protocols {
        isis {
            interface lt-0/1/0.0;
            interface lt-0/1/0.2;
            interface lo0.1 {
```

```
                passive;
            }
        }
    }
}
LS2 {
    interfaces {
        lt-0/1/0 {
            unit 1 {
                description LS2->LS1;
                encapsulation ethernet;
                peer-unit 2;
                family inet {
                    address 10.0.0.2/30;
                }
                family iso;
            }
            unit 4 {
                description LS2->LS3;
                encapsulation ethernet;
                peer-unit 3;
                family inet {
                    address 10.0.2.2/30;
                }
                family iso;
            }
        }
        lo0 {
            unit 2 {
                family iso {
                    address 49.0001.1720.1600.2002.00;
                }
            }
        }
    }
    protocols {
        isis {
            interface lt-0/1/0.1;
            interface lt-0/1/0.4;
            interface lo0.2 {
                passive;
            }
```

```
        }
    }
}
LS3 {
    interfaces {
        lt-0/1/0 {
            unit 3 {
                description LS3->LS2;
                encapsulation ethernet;
                peer-unit 4;
                family inet {
                    address 10.0.2.1/30;
                }
                family iso;
            }
            unit 5 {
                description LS3->LS1;
                encapsulation ethernet;
                peer-unit 0;
                family inet {
                    address 10.0.1.1/30;
                }
                family iso;
            }
        }
        lo0 {
            unit 3 {
                family iso {
                    address 49.0001.1234.1600.2231.00;
                }
            }
        }
    }
    protocols {
        isis {
            interface lt-0/1/0.3;
            interface lt-0/1/0.5;
            interface lo0.3 {
                passive;
            }
        }
```

```
    }
}
```

Verification

IN THIS SECTION

- Verifying That the Logical Systems Are Up | 275
- Verifying Connectivity Between the Logical Systems | 276

Confirm that the configuration is working properly.

Verifying That the Logical Systems Are Up

Purpose

Make sure that the interfaces are properly configured.

Action

```
user@host> show interfaces terse
Interface               Admin Link Proto    Local                 Remote
...
lt-0/1/0                up    up
lt-0/1/0.0              up    up   inet     10.0.1.2/30
                                  iso
lt-0/1/0.1              up    up   inet     10.0.0.2/30
                                  iso
lt-0/1/0.2              up    up   inet     10.0.0.1/30
                                  iso
lt-0/1/0.3              up    up   inet     10.0.2.1/30
                                  iso
lt-0/1/0.4              up    up   inet     10.0.2.2/30
                                  iso
lt-0/1/0.5              up    up   inet     10.0.1.1/30
                                  iso
...
```

Verifying Connectivity Between the Logical Systems

Purpose

Make sure that the IS-IS adjacencies are established by checking the logical system routing entries and by pinging the logical systems.

Action

```
user@host> show route logical-system LS1
inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.0/30        *[Direct/0] 3w0d 01:37:52
                    > via lt-0/1/0.2
10.0.0.1/32        *[Local/0] 3w0d 01:37:52
                      Local via lt-0/1/0.2
10.0.1.0/30        *[Direct/0] 3w0d 01:37:52
                    > via lt-0/1/0.0
10.0.1.2/32        *[Local/0] 3w0d 01:37:52
                      Local via lt-0/1/0.0
10.0.2.0/30        *[IS-IS/15] 3w0d 01:37:13, metric 20
                    > to 10.0.1.1 via lt-0/1/0.0
                      to 10.0.0.2 via lt-0/1/0.2

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

49.0001.1720.1600.1001/72
                   *[Direct/0] 3w0d 01:37:52
                    > via lo0.1
```

```
user@host> show route logical-system LS2
inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.0/30        *[Direct/0] 3w0d 01:38:01
                    > via lt-0/1/0.1
10.0.0.2/32        *[Local/0] 3w0d 01:38:01
```

```
                      Local via lt-0/1/0.1
10.0.1.0/30        *[IS-IS/15] 3w0d 01:37:01, metric 20
                      to 10.0.0.1 via lt-0/1/0.1
                    > to 10.0.2.1 via lt-0/1/0.4
10.0.2.0/30        *[Direct/0] 3w0d 01:38:01
                    > via lt-0/1/0.4
10.0.2.2/32        *[Local/0] 3w0d 01:38:01
                      Local via lt-0/1/0.4

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

49.0001.1720.1600.2002/72
                   *[Direct/0] 3w0d 01:38:01
                    > via lo0.2
```

```
user@host> show route logical-system LS3
inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.0/30        *[IS-IS/15] 3w0d 01:37:10, metric 20
                      to 10.0.2.2 via lt-0/1/0.3
                    > to 10.0.1.2 via lt-0/1/0.5
10.0.1.0/30        *[Direct/0] 3w0d 01:38:10
                    > via lt-0/1/0.5
10.0.1.1/32        *[Local/0] 3w0d 01:38:11
                      Local via lt-0/1/0.5
10.0.2.0/30        *[Direct/0] 3w0d 01:38:11
                    > via lt-0/1/0.3
10.0.2.1/32        *[Local/0] 3w0d 01:38:11
                      Local via lt-0/1/0.3

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

49.0001.1234.1600.2231/72
                   *[Direct/0] 3w0d 01:38:11
                    > via lo0.3
```

From LS1, Ping LS3

```
user@host> set cli logical-system LS1
```

```
user@host:LS1> ping 10.0.2.1
PING 10.0.2.1 (10.0.2.1): 56 data bytes
64 bytes from 10.0.2.1: icmp_seq=0 ttl=63 time=1.264 ms
64 bytes from 10.0.2.1: icmp_seq=1 ttl=63 time=1.189 ms
64 bytes from 10.0.2.1: icmp_seq=2 ttl=63 time=1.165 ms
^C
--- 10.0.2.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.165/1.206/1.264/0.042 ms
```

From LS3, Ping LS1

```
user@host> set cli logical-system LS3
```

```
user@host:LS3> ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1): 56 data bytes
64 bytes from 10.0.0.1: icmp_seq=0 ttl=63 time=1.254 ms
64 bytes from 10.0.0.1: icmp_seq=1 ttl=63 time=1.210 ms
^C
--- 10.0.0.1 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.210/1.232/1.254/0.022 ms
```

From LS1, Ping LS2

```
user@host> set cli logical-system LS1
```

```
user@host:LS1> ping 10.0.2.2
PING 10.0.2.2 (10.0.2.2): 56 data bytes
64 bytes from 10.0.2.2: icmp_seq=0 ttl=64 time=1.240 ms
64 bytes from 10.0.2.2: icmp_seq=1 ttl=64 time=1.204 ms
64 bytes from 10.0.2.2: icmp_seq=2 ttl=64 time=1.217 ms
^C
```

```
--- 10.0.2.2 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.204/1.220/1.240/0.015 ms
```

From LS2, Ping LS1

```
user@host> set cli logical-system LS2
```

```
user@host:LS2> ping 10.0.1.2
PING 10.0.1.2 (10.0.1.2): 56 data bytes
64 bytes from 10.0.1.2: icmp_seq=0 ttl=64 time=1.308 ms
64 bytes from 10.0.1.2: icmp_seq=1 ttl=64 time=1.235 ms
^C
--- 10.0.1.2 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.235/1.272/1.308/0.037 ms
```

From LS2, Ping LS3

```
user@host> set cli logical-system LS2
```

```
user@host:LS2> ping 10.0.1.1
PING 10.0.1.1 (10.0.1.1): 56 data bytes
64 bytes from 10.0.1.1: icmp_seq=0 ttl=64 time=1.253 ms
64 bytes from 10.0.1.1: icmp_seq=1 ttl=64 time=1.194 ms
64 bytes from 10.0.1.1: icmp_seq=2 ttl=64 time=1.212 ms
64 bytes from 10.0.1.1: icmp_seq=3 ttl=64 time=1.221 ms
64 bytes from 10.0.1.1: icmp_seq=4 ttl=64 time=1.195 ms
^C
--- 10.0.1.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.194/1.215/1.253/0.022 ms
```

From LS3, Ping LS2

```
user@host> set cli logical-system LS3
```

```
user@host:LS3> ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2): 56 data bytes
64 bytes from 10.0.0.2: icmp_seq=0 ttl=64 time=1.240 ms
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=1.217 ms
^C
--- 10.0.0.2 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.217/1.228/1.240/0.012 ms
```

SEE ALSO

Example: Creating an Interface on a Logical System | 19

Example: Connecting Logical Systems Within the Same Device Using Logical Tunnel Interfaces on MX Series Routers and EX Series Switches | 27

Example: Configuring an IS-IS Default Route Policy on Logical Systems

IN THIS SECTION

- Requirements | 281
- Overview | 281
- Configuration | 282
- Verification | 286

This example shows logical systems configured on a single physical router and explains how to configure a default route on one logical system.

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

IN THIS SECTION

- Topology | 282

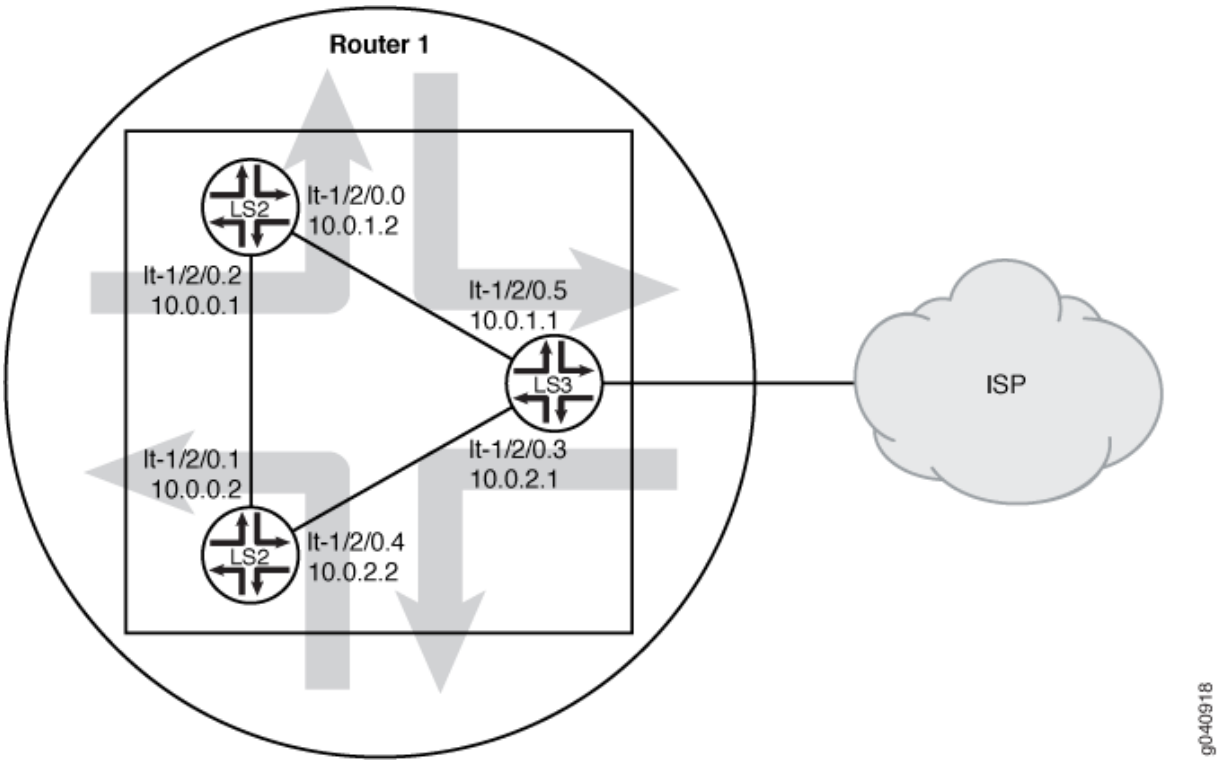
This example shows a logical system redistributing a default route to other logical systems. All logical systems are running IS-IS. A common reason for a default route is to provide a path for sending traffic destined outside the IS-IS domain.

In this example, the default route is not used for forwarding traffic. The `no-install` statement prevents the route from being installed in the forwarding table of Logical System LS3. If you configure a route so it is not installed in the forwarding table, the route is still eligible to be exported from the routing table to other protocols. The `discard` statement silently drops packets without notice.

Topology

Figure 20 on page 282 shows the sample network.

Figure 20: IS-IS with a Default Route to an ISP



Configuration

IN THIS SECTION

- CLI Quick Configuration | 283
- Procedure | 283
- Results | 285

CLI Quick Configuration

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

```
set logical-systems LS3 interfaces lt-1/2/0 unit 3 description LS3->LS2
set logical-systems LS3 interfaces lt-1/2/0 unit 3 encapsulation ethernet
set logical-systems LS3 interfaces lt-1/2/0 unit 3 peer-unit 4
set logical-systems LS3 interfaces lt-1/2/0 unit 3 family inet address 10.0.2.1/30
set logical-systems LS3 interfaces lt-1/2/0 unit 3 family iso
set logical-systems LS3 interfaces lt-1/2/0 unit 5 description LS3->LS1
set logical-systems LS3 interfaces lt-1/2/0 unit 5 encapsulation ethernet
set logical-systems LS3 interfaces lt-1/2/0 unit 5 peer-unit 0
set logical-systems LS3 interfaces lt-1/2/0 unit 5 family inet address 10.0.1.1/30
set logical-systems LS3 interfaces lt-1/2/0 unit 5 family iso
set logical-systems LS3 interfaces lo0 unit 3 family iso address 49.0001.1234.1600.2231.00
set logical-systems LS3 protocols isis export isis-default
set logical-systems LS3 protocols isis interface lt-1/2/0.3
set logical-systems LS3 protocols isis interface lt-1/2/0.5
set logical-systems LS3 protocols isis interface lo0.3 passive
set logical-systems LS3 routing-options static route 0.0.0.0/0 discard
set logical-systems LS3 routing-options static route 0.0.0.0/0 no-install
set logical-systems LS3 policy-options policy-statement isis-default from protocol static
set logical-systems LS3 policy-options policy-statement isis-default from route-filter 0.0.0.0/0
exact
set logical-systems LS3 policy-options policy-statement isis-default then accept
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure an IS-IS default route policy on logical systems:

1. Configure the logical tunnel interfaces.

```
[edit logical-systems LS3 interfaces lt-1/2/0]
user@R1# set unit 3 description LS3->LS2
user@R1# set unit 3 encapsulation ethernet
user@R1# set unit 3 peer-unit 4
user@R1# set unit 3 family inet address 10.0.2.1/30
user@R1# set unit 3 family iso
user@R1# set unit 5 description LS3->LS1
user@R1# set unit 5 encapsulation ethernet
user@R1# set unit 5 peer-unit 0
user@R1# set unit 5 family inet address 10.0.1.1/30
user@R1# set unit 5 family iso
[edit logical-systems LS3 interfaces lo0 unit 3]
user@R1# set family iso address 49.0001.1234.1600.2231.00
```

2. Enable IS-IS on the interfaces.

```
[edit logical-systems LS3 protocols isis]
user@R1# set interface lt-1/2/0.3
user@R1# set interface lt-1/2/0.5
user@R1# set interface lo0.3 passive
```

3. Configure the default route on Logical System LS3.

```
[edit logical-systems LS3 routing-options]
user@R1# set static route 0.0.0.0/0 discard
user@R1# set static route 0.0.0.0/0 no-install
```

4. Configure the default route policy on Logical System LS3.

```
[edit logical-systems LS3 policy-options]
user@R1# set policy-statement isis-default from protocol static
user@R1# set policy-statement isis-default from route-filter 0.0.0.0/0 exact
user@R1# set policy-statement isis-default then accept
```

5. Apply the export policy to IS-IS on Logical System LS3.

```
[edit logical-systems LS3 protocols isis]
user@R1# set export isis-default
```

6. If you are done configuring the device, commit the configuration.

```
[edit]
user@R1# commit
```

Results

From configuration mode, confirm your configuration by issuing the `show logical-systems LS3` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R1# show logical-systems LS3
interfaces {
    lt-1/2/0 {
        unit 3 {
            description LS3->LS2;
            encapsulation ethernet;
            peer-unit 4;
            family inet {
                address 10.0.2.1/30;
            }
            family iso;
        }
        unit 5 {
            description LS3->LS1;
            encapsulation ethernet;
            peer-unit 0;
            family inet {
                address 10.0.1.1/30;
            }
            family iso;
        }
    }
    lo0 {
        unit 3 {
```

```
            family iso {
                address 49.0001.1234.1600.2231.00;
            }
        }
    }
    }
    protocols {
        isis {
            export isis-default;
            interface lt-1/2/0.3;
            interface lt-1/2/0.5;
            interface lo0.3 {
                passive;
            }
        }
    }
    policy-options {
        policy-statement isis-default {
            from {
                protocol static;
                route-filter 0.0.0.0/0 exact;
            }
            then accept;
        }
    }
    routing-options {
        static {
            route 0.0.0.0/0 {
                discard;
                no-install;
            }
        }
    }
}
```

Verification

IN THIS SECTION

- Verifying That the Static Route Is Redistributed | 287

Confirm that the configuration is working properly.

Verifying That the Static Route Is Redistributed

Purpose

Make sure that the IS-IS policy is working by checking the routing tables.

Action

```
user@R1> show route logical-system LS3
inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[Static/5] 00:00:45
                      Discard
10.0.0.0/30        *[IS-IS/15] 1w0d 10:14:14, metric 20
                      to 10.0.2.2 via lt-1/2/0.3
                    > to 10.0.1.2 via lt-1/2/0.5
10.0.1.0/30        *[Direct/0] 1w0d 10:15:18
                    > via lt-1/2/0.5
10.0.1.1/32        *[Local/0] 1w0d 10:15:18
                      Local via lt-1/2/0.5
10.0.2.0/30        *[Direct/0] 1w0d 10:15:18
                    > via lt-1/2/0.3
10.0.2.1/32        *[Local/0] 1w0d 10:15:18
                      Local via lt-1/2/0.3

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

49.0001.1234.1600.2231/72
                   *[Direct/0] 1w0d 10:17:19
                    > via lo0.3
```

```
user@R1> show route logical-system LS2
inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[IS-IS/160] 00:01:38, metric 10
```

```
                    > to 10.0.2.1 via lt-1/2/0.4
10.0.0.0/30         *[Direct/0] 1w0d 10:16:11
                    > via lt-1/2/0.1
10.0.0.2/32         *[Local/0] 1w0d 10:16:11
                      Local via lt-1/2/0.1
10.0.1.0/30         *[IS-IS/15] 1w0d 10:15:07, metric 20
                    > to 10.0.0.1 via lt-1/2/0.1
                      to 10.0.2.1 via lt-1/2/0.4
10.0.2.0/30         *[Direct/0] 1w0d 10:16:11
                    > via lt-1/2/0.4
10.0.2.2/32         *[Local/0] 1w0d 10:16:11
                      Local via lt-1/2/0.4

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

49.0001.1720.1600.2002/72
                    *[Direct/0] 1w0d 10:18:12
                    > via lo0.2
```

```
user@R1> show route logical-system LS1
inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0           *[IS-IS/160] 00:02:01, metric 10
                    > to 10.0.1.1 via lt-1/2/0.0
10.0.0.0/30         *[Direct/0] 1w0d 10:16:34
                    > via lt-1/2/0.2
10.0.0.1/32         *[Local/0] 1w0d 10:16:34
                      Local via lt-1/2/0.2
10.0.1.0/30         *[Direct/0] 1w0d 10:16:34
                    > via lt-1/2/0.0
10.0.1.2/32         *[Local/0] 1w0d 10:16:34
                      Local via lt-1/2/0.0
10.0.2.0/30         *[IS-IS/15] 1w0d 10:15:55, metric 20
                      to 10.0.1.1 via lt-1/2/0.0
                    > to 10.0.0.2 via lt-1/2/0.2

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
49.0001.1720.1600.1001/72
                    *[Direct/0] 1w0d 10:18:35
                     > via lo0.1
```

Meaning

The routing table on Logical System LS3 contains the default 0.0.0.0/0 route from protocol `Static`. The routing tables on Logical System LS1 and Logical System LS2 contain the default 0.0.0.0/0 route from protocol `IS-IS`. If Logical System LS1 and Logical System LS2 receive packets destined for networks not specified in their routing tables, those packets will be sent to Logical System LS3 for further processing. This configuration assumes that Logical System LS3 has a connection to an ISP or another external network.

SEE ALSO

Example: Creating an Interface on a Logical System | 19

RELATED DOCUMENTATION

IS-IS User Guide

Understanding Logical Systems for Routers and Switches | 2

OSPF on Logical Systems

IN THIS SECTION

- Example: Configuring OSPF on Logical Systems Within the Same Router | 290
- Understanding OSPF Routing Policy | 300
- Example: Configuring an OSPF Default Route Policy on Logical Systems | 303
- Example: Configuring a Conditional OSPF Default Route Policy on Logical Systems | 310
- Example: Configuring an OSPF Import Policy on Logical Systems | 321

Logical Systems enable you to configure an OSPF network, running on a single physical router. For more information, see the following topics:

Example: Configuring OSPF on Logical Systems Within the Same Router

IN THIS SECTION

- Requirements | 290
- Overview | 290
- Configuration | 291
- Verification | 297

This example shows how to configure an OSPF network using multiple logical systems that are running on a single physical router. The logical systems are connected by logical tunnel interfaces.

Requirements

You must connect the logical systems by using logical tunnel (lt) interfaces. See Example: Connecting Logical Systems Within the Same Device Using Logical Tunnel Interfaces on MX Series Routers and EX Series Switches.

Overview

IN THIS SECTION

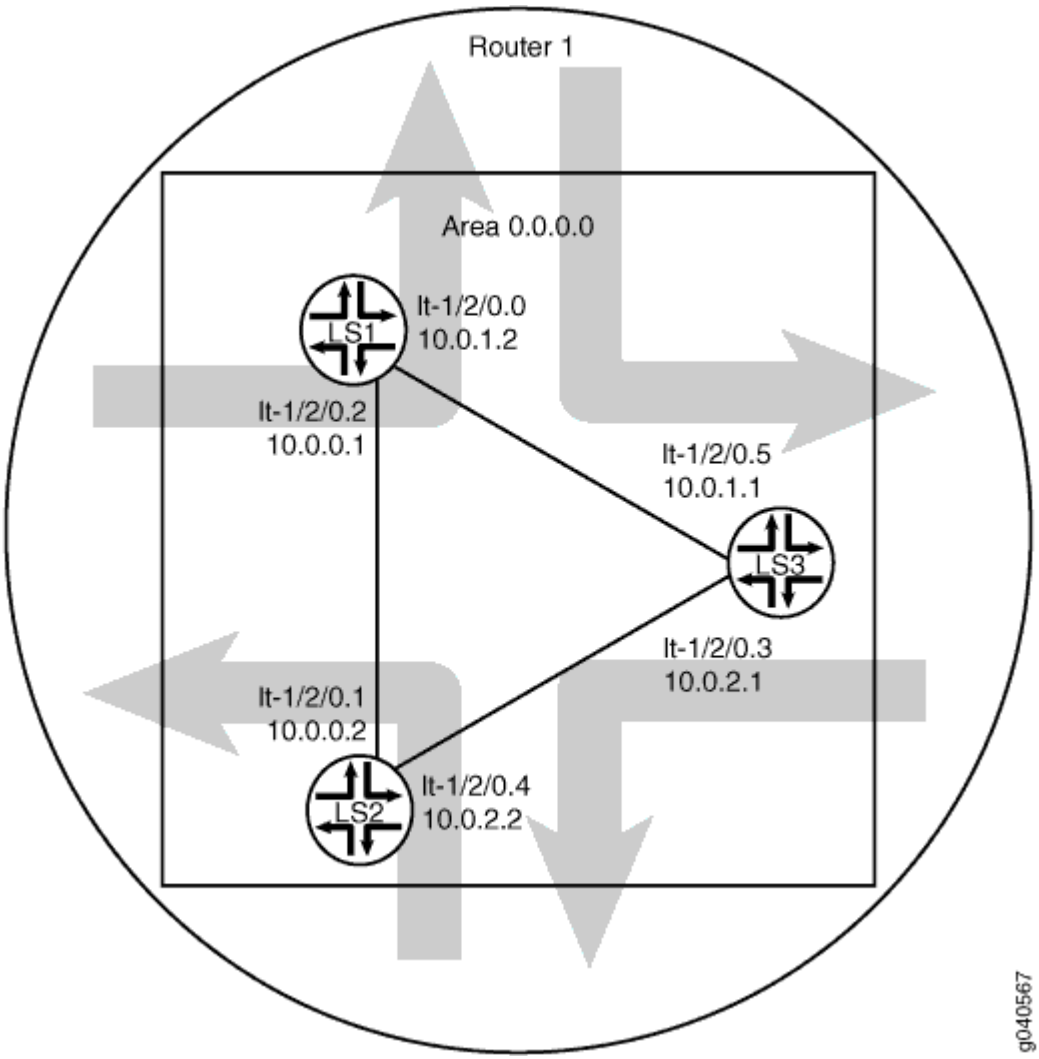
- Topology | 291

This example shows the configuration of a single OSPF area with three logical systems running on one physical router. Each logical system has its own routing table. The configuration enables the protocol on all logical system interfaces that participate in the OSPF domain and specifies the area that the interfaces are in.

Topology

Figure 21 on page 291 shows the sample network.

Figure 21: OSPF on Logical Systems



Configuration

IN THIS SECTION

- CLI Quick Configuration | 292
- Procedure | 293

- Results | 295

CLI Quick Configuration

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

```
set logical-systems LS1 interfaces lt-1/2/0 unit 0 description LS1->LS3
set logical-systems LS1 interfaces lt-1/2/0 unit 0 encapsulation ethernet
set logical-systems LS1 interfaces lt-1/2/0 unit 0 peer-unit 5
set logical-systems LS1 interfaces lt-1/2/0 unit 0 family inet address 10.0.1.2/30
set logical-systems LS1 interfaces lt-1/2/0 unit 2 description LS1->LS2
set logical-systems LS1 interfaces lt-1/2/0 unit 2 encapsulation ethernet
set logical-systems LS1 interfaces lt-1/2/0 unit 2 peer-unit 1
set logical-systems LS1 interfaces lt-1/2/0 unit 2 family inet address 10.0.0.1/30
set logical-systems LS1 protocols ospf area 0.0.0.0 interface lt-1/2/0.0
set logical-systems LS1 protocols ospf area 0.0.0.0 interface lt-1/2/0.2
set logical-systems LS2 interfaces lt-1/2/0 unit 1 description LS2->LS1
set logical-systems LS2 interfaces lt-1/2/0 unit 1 encapsulation ethernet
set logical-systems LS2 interfaces lt-1/2/0 unit 1 peer-unit 2
set logical-systems LS2 interfaces lt-1/2/0 unit 1 family inet address 10.0.0.2/30
set logical-systems LS2 interfaces lt-1/2/0 unit 4 description LS2->LS3
set logical-systems LS2 interfaces lt-1/2/0 unit 4 encapsulation ethernet
set logical-systems LS2 interfaces lt-1/2/0 unit 4 peer-unit 3
set logical-systems LS2 interfaces lt-1/2/0 unit 4 family inet address 10.0.2.2/30
set logical-systems LS2 protocols ospf area 0.0.0.0 interface lt-1/2/0.1
set logical-systems LS2 protocols ospf area 0.0.0.0 interface lt-1/2/0.4
set logical-systems LS3 interfaces lt-1/2/0 unit 3 description LS3->LS2
set logical-systems LS3 interfaces lt-1/2/0 unit 3 encapsulation ethernet
set logical-systems LS3 interfaces lt-1/2/0 unit 3 peer-unit 4
set logical-systems LS3 interfaces lt-1/2/0 unit 3 family inet address 10.0.2.1/30
set logical-systems LS3 interfaces lt-1/2/0 unit 5 description LS3->LS1
set logical-systems LS3 interfaces lt-1/2/0 unit 5 encapsulation ethernet
set logical-systems LS3 interfaces lt-1/2/0 unit 5 peer-unit 0
set logical-systems LS3 interfaces lt-1/2/0 unit 5 family inet address 10.0.1.1/30
set logical-systems LS3 protocols ospf area 0.0.0.0 interface lt-1/2/0.5
set logical-systems LS3 protocols ospf area 0.0.0.0 interface lt-1/2/0.3
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure OSPF on logical systems:

1. Configure the logical tunnel interface on Logical System LS1 connecting to Logical System LS2.

```
[edit]
user@host# set logical-systems LS1 interfaces lt-1/2/0 unit 2 description LS1->LS2
user@host# set logical-systems LS1 interfaces lt-1/2/0 unit 2 encapsulation ethernet
user@host# set logical-systems LS1 interfaces lt-1/2/0 unit 2 peer-unit 1
user@host# set logical-systems LS1 interfaces lt-1/2/0 unit 2 family inet address 10.0.0.1/30
```

2. Configure the logical tunnel interface on Logical System LS1 connecting to Logical System LS3.

```
[edit]
user@host# set logical-systems LS1 interfaces lt-1/2/0 unit 0 description LS1->LS3
user@host# set logical-systems LS1 interfaces lt-1/2/0 unit 0 encapsulation ethernet
user@host# set logical-systems LS1 interfaces lt-1/2/0 unit 0 peer-unit 5
user@host# set logical-systems LS1 interfaces lt-1/2/0 unit 0 family inet address 10.0.1.2/30
```

3. Configure the logical tunnel interface on Logical System LS2 connecting to Logical System LS1.

```
[edit]
user@host# set logical-systems LS2 interfaces lt-1/2/0 unit 1 description LS2->LS1
user@host# set logical-systems LS2 interfaces lt-1/2/0 unit 1 encapsulation ethernet
user@host# set logical-systems LS2 interfaces lt-1/2/0 unit 1 peer-unit 2
user@host# set logical-systems LS2 interfaces lt-1/2/0 unit 1 family inet address 10.0.0.2/30
```

4. Configure the logical tunnel interface on Logical System LS2 connecting to Logical System LS3.

```
[edit]
user@host# set logical-systems LS2 interfaces lt-1/2/0 unit 4 description LS2->LS3
user@host# set logical-systems LS2 interfaces lt-1/2/0 unit 4 encapsulation ethernet
```

```
user@host# set logical-systems LS2 interfaces lt-1/2/0 unit 4 peer-unit 3
user@host# set logical-systems LS2 interfaces lt-1/2/0 unit 4 family inet address 10.0.2.2/30
```

5. Configure the logical tunnel interface on Logical System LS3 connecting to Logical System LS2.

```
[edit]
user@host# set logical-systems LS3 interfaces lt-1/2/0 unit 3 description LS3->LS2
user@host# set logical-systems LS3 interfaces lt-1/2/0 unit 3 encapsulation ethernet
user@host# set logical-systems LS3 interfaces lt-1/2/0 unit 3 peer-unit 4
user@host# set logical-systems LS3 interfaces lt-1/2/0 unit 3 family inet address 10.0.2.1/30
```

6. Configure the logical tunnel interface on Logical System LS3 connecting to Logical System LS1.

```
[edit]
user@host# set logical-systems LS3 interfaces lt-1/2/0 unit 5 description LS3->LS1
user@host# set logical-systems LS3 interfaces lt-1/2/0 unit 5 encapsulation ethernet
user@host# set logical-systems LS3 interfaces lt-1/2/0 unit 5 peer-unit 0
user@host# set logical-systems LS3 interfaces lt-1/2/0 unit 5 family inet address 10.0.1.1/30
```

7. Configure OSPF on all the interfaces.

```
[edit]
user@host# set logical-systems LS1 protocols ospf area 0.0.0.0 interface lt-1/2/0.0
user@host# set logical-systems LS1 protocols ospf area 0.0.0.0 interface lt-1/2/0.2
user@host# set logical-systems LS2 protocols ospf area 0.0.0.0 interface lt-1/2/0.1
user@host# set logical-systems LS2 protocols ospf area 0.0.0.0 interface lt-1/2/0.4
user@host# set logical-systems LS3 protocols ospf area 0.0.0.0 interface lt-1/2/0.5
user@host# set logical-systems LS3 protocols ospf area 0.0.0.0 interface lt-1/2/0.3
```

8. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Results

Confirm your configuration by issuing the `show logical-systems` command.

```
show logical-systems
LS1 {
    interfaces {
        lt-1/2/0 {
            unit 0 {
                description LS1->LS3;
                encapsulation ethernet;
                peer-unit 5;
                family inet {
                    address 10.0.1.2/30;
                }
            }
            unit 2 {
                description LS1->LS2;
                encapsulation ethernet;
                peer-unit 1;
                family inet {
                    address 10.0.0.1/30;
                }
            }
        }
    }
    protocols {
        ospf {
            area 0.0.0.0 {
                interface lt-1/2/0.0;
                interface lt-1/2/0.2;
            }
        }
    }
}
LS2 {
    interfaces {
        lt-1/2/0 {
            unit 1 {
                description LS2->LS1;
                encapsulation ethernet;
                peer-unit 2;
```

```
                family inet {
                    address 10.0.0.2/30;
                }
            }
            unit 4 {
                description LS2->LS3;
                encapsulation ethernet;
                peer-unit 3;
                family inet {
                    address 10.0.2.2/30;
                }
            }
        }
    }
    protocols {
        ospf {
            area 0.0.0.0 {
                interface lt-1/2/0.1;
                interface lt-1/2/0.4;
            }
        }
    }
}
LS3 {
    interfaces {
        lt-1/2/0 {
            unit 3 {
                description LS3->LS2;
                encapsulation ethernet;
                peer-unit 4;
                family inet {
                    address 10.0.2.1/30;
                }
            }
            unit 5 {
                description LS3->LS1;
                encapsulation ethernet;
                peer-unit 0;
                family inet {
                    address 10.0.1.1/30;
                }
            }
        }
```

```
    }
    protocols {
        ospf {
            area 0.0.0.0 {
                interface lt-1/2/0.5;
                interface lt-1/2/0.3;
            }
        }
    }
}
```

Verification

IN THIS SECTION

- Verifying That the Logical Systems Are Up | **297**
- Verifying Connectivity Between the Logical Systems | **298**

Confirm that the configuration is working properly.

Verifying That the Logical Systems Are Up

Purpose

Make sure that the interfaces are properly configured.

Action

```
user@host> show interfaces terse
Interface               Admin Link Proto    Local                 Remote
...
lt-1/2/0                up    up
lt-1/2/0.0              up    up   inet     10.0.1.2/30
lt-1/2/0.1              up    up   inet     10.0.0.2/30
lt-1/2/0.2              up    up   inet     10.0.0.1/30
lt-1/2/0.3              up    up   inet     10.0.2.1/30
lt-1/2/0.4              up    up   inet     10.0.2.2/30
```

```
lt-1/2/0.5             up    up   inet     10.0.1.1/30
...
```

Verifying Connectivity Between the Logical Systems

Purpose

Make sure that the OSPF adjacencies are established by checking the OSPF neighbor tables, checking the routing tables, and pinging the logical systems.

Action

```
user@host> show ospf neighbor logical-system LS1
Address          Interface              State     ID               Pri  Dead
10.0.1.1         lt-1/2/0.0             Full      10.0.1.1         128    37
10.0.0.2         lt-1/2/0.2             Full      10.0.0.2         128    33
```

```
user@host> show ospf neighbor logical-system LS2
Address          Interface              State     ID               Pri  Dead
10.0.0.1         lt-1/2/0.1             Full      10.0.0.1         128    32
10.0.2.1         lt-1/2/0.4             Full      10.0.1.1         128    36
```

```
user@host> show ospf neighbor logical-system LS3
Address          Interface              State     ID               Pri  Dead
10.0.2.2         lt-1/2/0.3             Full      10.0.0.2         128    36
10.0.1.2         lt-1/2/0.5             Full      10.0.0.1         128    37
```

```
user@host> show route logical-system LS1
inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.0/30        *[Direct/0] 00:28:00
                    > via lt-1/2/0.2
10.0.0.1/32        *[Local/0] 00:28:00
                      Local via lt-1/2/0.2
10.0.1.0/30        *[Direct/0] 00:28:00
                    > via lt-1/2/0.0
```

```
10.0.1.2/32        *[Local/0] 00:28:00
                      Local via lt-1/2/0.0
10.0.2.0/30        *[OSPF/10] 00:27:05, metric 2
                    > to 10.0.1.1 via lt-1/2/0.0
                      to 10.0.0.2 via lt-1/2/0.2
224.0.0.5/32       *[OSPF/10] 00:28:03, metric 1
                      MultiRecv
```

```
user@host> show route logical-system LS2
inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.0/30        *[Direct/0] 00:28:31
                    > via lt-1/2/0.1
10.0.0.2/32        *[Local/0] 00:28:32
                      Local via lt-1/2/0.1
10.0.1.0/30        *[OSPF/10] 00:27:38, metric 2
                    > to 10.0.0.1 via lt-1/2/0.1
                      to 10.0.2.1 via lt-1/2/0.4
10.0.2.0/30        *[Direct/0] 00:28:32
                    > via lt-1/2/0.4
10.0.2.2/32        *[Local/0] 00:28:32
                      Local via lt-1/2/0.4
224.0.0.5/32       *[OSPF/10] 00:28:34, metric 1
                      MultiRecv
```

```
user@host> show route logical-system LS3
inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.0/30        *[OSPF/10] 00:28:23, metric 2
                    > to 10.0.2.2 via lt-1/2/0.3
                      to 10.0.1.2 via lt-1/2/0.5
10.0.1.0/30        *[Direct/0] 00:29:13
                    > via lt-1/2/0.5
10.0.1.1/32        *[Local/0] 00:29:15
                      Local via lt-1/2/0.5
10.0.2.0/30        *[Direct/0] 00:29:14
                    > via lt-1/2/0.3
10.0.2.1/32        *[Local/0] 00:29:15
```

```
                    Local via lt-1/2/0.3
224.0.0.5/32       *[OSPF/10] 00:29:16, metric 1
                    MultiRecv
```

From LS1, Ping LS3

```
user@host> set cli logical-system LS1
```

```
user@host:LS1> ping 10.0.2.1
PING 10.0.2.1 (10.0.2.1): 56 data bytes
64 bytes from 10.0.2.1: icmp_seq=0 ttl=64 time=1.215 ms
64 bytes from 10.0.2.1: icmp_seq=1 ttl=64 time=1.150 ms
64 bytes from 10.0.2.1: icmp_seq=2 ttl=64 time=1.134 ms
```

From LS3, Ping LS1

```
user@host> set cli logical-system LS3
```

```
user@host:LS3> ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1): 56 data bytes
64 bytes from 10.0.0.1: icmp_seq=0 ttl=64 time=1.193 ms
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=1.114 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=1.190 ms
```

Understanding OSPF Routing Policy

IN THIS SECTION

- Routing Policy Terms | 301
- Routing Policy Match Conditions | 301
- Routing Policy Actions | 302

Each routing policy is identified by a policy name. The name can contain letters, numbers, and hyphens (-) and can be up to 255 characters long. To include spaces in the name, enclose the entire name in double quotation marks. Each routing policy name must be unique within a configuration. Once a policy is created and named, it must be applied before it is active.

In the `import` statement, you list the name of the routing policy used to filter OSPF external routes from being installed into the routing tables of OSPF neighbors. You can filter the routes, but not link-state address (LSA) flooding. An external route is a route that is outside the OSPF Autonomous System (AS). The import policy does not impact the OSPF database. This means that the import policy has no impact on the link-state advertisements.

In the `export` statement, you list the name of the routing policy to be evaluated when routes are being exported from the routing table into OSPF.

By default, if a routing device has multiple OSPF areas, learned routes from other areas are automatically installed into area 0 of the routing table.

To specify more than one policy and create a policy chain, you list the policies using a space as a separator. If multiple policies are specified, the policies are evaluated in the order in which they are specified. As soon as an accept or reject action is executed, the policy chain evaluation ends.

This topic describes the following information:

Routing Policy Terms

Routing policies are made up of one or more terms. A term is a named structure in which match conditions and actions are defined. You can define one or more terms. The name can contain letters, numbers, and hyphens (-) and can be up to 255 characters long. To include spaces in the name, enclose the entire name in double quotation marks.

Each term contains a set of match conditions and a set of actions:

- Match conditions are criteria that a route must match before the actions can be applied. If a route matches all criteria, one or more actions are applied to the route.
- Actions specify whether to accept or reject the route, control how a series of policies are evaluated, and manipulate the characteristics associated with a route.

Routing Policy Match Conditions

A match condition defines the criteria that a route must match for an action to take place. You can define one or more match conditions for each term. If a route matches all of the match conditions for a particular term, the actions defined for that term are processed.

Each term can include two statements, `from` and `to`, that define the match conditions:

- In the `from` statement, you define the criteria that an incoming route must match. You can specify one or more match conditions. If you specify more than one, they all must match the route for a match to occur.

  The `from` statement is optional. If you omit the `from` and the `to` statements, all routes are considered to match.

  > **NOTE**: In export policies, omitting the `from` statement from a routing policy term might lead to unexpected results.

- In the `to` statement, you define the criteria that an outgoing route must match. You can specify one or more match conditions. If you specify more than one, they all must match the route for a match to occur.

The order of the match conditions in a term is not important because a route must match all match conditions in a term for an action to be taken.

For a complete list of match conditions, see Configuring Match Conditions in Routing Policy Terms.

Routing Policy Actions

An action defines what the routing device does with the route when the route matches all the match conditions in the `from` and `to` statements for a particular term. If a term does not have `from` and `to` statements, all routes are considered to match and the actions apply to all routes.

Each term can have one or more of the following types of actions. The actions are configured under the `then` statement.

- Flow control actions, which affect whether to accept or reject the route and whether to evaluate the next term or routing policy.
- Actions that manipulate route characteristics.
- Trace action, which logs route matches.

The `then` statement is optional. If you omit it, one of the following occurs:

- The next term in the routing policy, if one exists, is evaluated.
- If the routing policy has no more terms, the next routing policy, if one exists, is evaluated.
- If there are no more terms or routing policies, the `accept` or `reject` action specified by the default policy is executed.

For a complete list of routing policy actions, see Configuring Actions in Routing Policy Terms.

Example: Configuring an OSPF Default Route Policy on Logical Systems

IN THIS SECTION

- Requirements | 303
- Overview | 303
- Configuration | 304
- Verification | 307

This example shows how to configure a default route on one logical system and inject the default route into OSPF area 0. In this example, OSPF area 0 contains three logical systems that are configured on a single physical router.

Requirements

Before you begin:

- Connect the logical systems by using logical tunnel (**lt**) interfaces. See "Example: Connecting Logical Systems Within the Same Device Using Logical Tunnel Interfaces on MX Series Routers and EX Series Switches" on page 27.
- Enable OSPF on the interfaces. See *Example: Configuring OSPF on Logical Systems Within the Same Router*.

Overview

IN THIS SECTION

- Topology | 304

This example shows a logical system redistributing a default route to other logical systems. All logical systems are running OSPF. A common reason for a default route is to provide a path for sending traffic destined outside the OSPF domain.

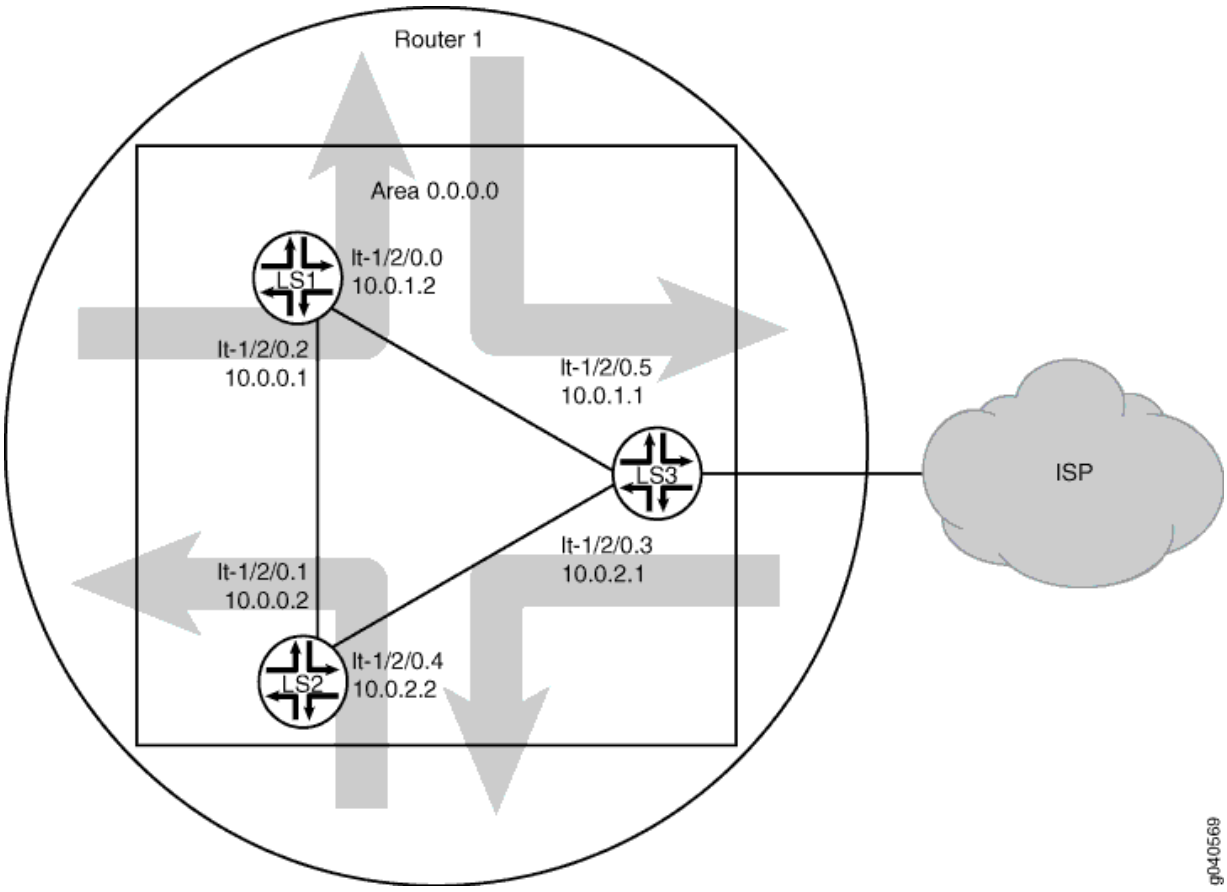
In this example, the default route is not used for forwarding traffic. The `no-install` statement prevents the route from being installed in the forwarding table of Logical System LS3. If you configure a route so it is

not installed in the forwarding table, the route is still eligible to be exported from the routing table to other protocols. The `discard` statement silently drops packets without notice.

Topology

Figure 22 on page 304 shows the sample network.

Figure 22: OSPF with a Default Route to an ISP



Configuration

IN THIS SECTION

- CLI Quick Configuration | 305
- Procedure | 305
- Results | 306

CLI Quick Configuration

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

```
set logical-systems LS3 routing-options static route 0.0.0.0/0 discard
set logical-systems LS3 routing-options static route 0.0.0.0/0 no-install
set logical-systems LS3 policy-options policy-statement ospf-default from protocol static
set logical-systems LS3 policy-options policy-statement ospf-default from route-filter 0.0.0.0/0
exact
set logical-systems LS3 policy-options policy-statement ospf-default then accept
set logical-systems LS3 protocols ospf export ospf-default
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure an OSPF default route policy on logical systems:

1. Change the context to Logical System LS3.

```
[edit]
user@host> set cli logical-system LS3
```

2. Configure the default route on Logical System LS3.

```
[edit]
user@host:LS3# set routing-options static route 0.0.0.0/0 discard
user@host:LS3# set routing-options static route 0.0.0.0/0 no-install
```

3. Configure the policy on Logical System LS3.

```
[edit]
user@host:LS3# set policy-options policy-statement ospf-default from protocol static
```

```
user@host:LS3# set policy-options policy-statement ospf-default from route-filter 0.0.0.0/0
exact
user@host:LS3# set policy-options policy-statement ospf-default then accept
```

4. Apply the export policy to OSPF on Logical System LS3.

```
[edit]
user@host:LS3# set protocols ospf export ospf-default
```

5. If you are done configuring the device, commit the configuration.

```
[edit]
user@host:LS3# commit
```

Results

Confirm your configuration by issuing the `show logical-systems LS3` command.

```
show logical-systems LS3
interfaces {
    lt-1/2/0 {
        unit 3 {
            description LS3->LS2;
            encapsulation ethernet;
            peer-unit 4;
            family inet {
                address 10.0.2.1/30;
            }
        }
        unit 5 {
            description LS3->LS1;
            encapsulation ethernet;
            peer-unit 0;
            family inet {
                address 10.0.1.1/30;
            }
        }
    }
}
```

```
protocols {
    ospf {
        export ospf-default;
        area 0.0.0.0 {
            interface lt-1/2/0.5;
            interface lt-1/2/0.3;
        }
    }
}
policy-options {
    policy-statement ospf-default {
        from {
            protocol static;
            route-filter 0.0.0.0/0 exact;
        }
        then accept;
    }
}
routing-options {
    static {
        route 0.0.0.0/0 {
            discard;
            no-install;
        }
    }
}
```

Verification

IN THIS SECTION

- Verifying That the Static Route Is Redistributed | **308**

Confirm that the configuration is working properly.

Verifying That the Static Route Is Redistributed

Purpose

Make sure that the OSPF policy is working by checking the routing tables.

Action

```
user@host> show route logical-system LS3
inet.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[Static/5] 01:04:38
                      Discard
10.0.0.0/30        *[OSPF/10] 11:53:55, metric 2
                      to 10.0.2.2 via lt-1/2/0.3
                    > to 10.0.1.2 via lt-1/2/0.5
10.0.1.0/30        *[Direct/0] 11:54:50
                    > via lt-1/2/0.5
10.0.1.1/32        *[Local/0] 11:54:54
                      Local via lt-1/2/0.5
10.0.2.0/30        *[Direct/0] 11:54:50
                    > via lt-1/2/0.3
10.0.2.1/32        *[Local/0] 11:54:54
                      Local via lt-1/2/0.3
224.0.0.5/32       *[OSPF/10] 11:56:55, metric 1
                      MultiRecv
```

```
user@host> show route logical-system LS1
inet.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[OSPF/150] 01:02:34, metric 0, tag 0
                    > to 10.0.1.1 via lt-1/2/0.0
10.0.0.0/30        *[Direct/0] 11:52:46
                    > via lt-1/2/0.2
10.0.0.1/32        *[Local/0] 11:52:50
                      Local via lt-1/2/0.2
10.0.1.0/30        *[Direct/0] 11:52:46
                    > via lt-1/2/0.0
```

```
10.0.1.2/32        *[Local/0] 11:52:50
                      Local via lt-1/2/0.0
10.0.2.0/30        *[OSPF/10] 11:51:56, metric 2
                    > to 10.0.1.1 via lt-1/2/0.0
                      to 10.0.0.2 via lt-1/2/0.2
224.0.0.5/32       *[OSPF/10] 11:54:50, metric 1
                      MultiRecv
```

```
user@host> show route logical-system LS2
inet.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[OSPF/150] 01:05:20, metric 0, tag 0
                    > to 10.0.2.1 via lt-1/2/0.4
10.0.0.0/30        *[Direct/0] 11:55:32
                    > via lt-1/2/0.1
10.0.0.2/32        *[Local/0] 11:55:36
                      Local via lt-1/2/0.1
10.0.1.0/30        *[OSPF/10] 11:54:37, metric 2
                    > to 10.0.0.1 via lt-1/2/0.1
                      to 10.0.2.1 via lt-1/2/0.4
10.0.2.0/30        *[Direct/0] 11:55:32
                    > via lt-1/2/0.4
10.0.2.2/32        *[Local/0] 11:55:36
                      Local via lt-1/2/0.4
224.0.0.5/32       *[OSPF/10] 11:57:36, metric 1
                      MultiRecv
```

Meaning

The routing table on Logical System LS3 contains the default 0.0.0.0/0 route from protocol **Static**. The routing tables on Logical System LS1 and Logical System LS2 contain the default 0.0.0.0/0 route from protocol **OSPF**. If Logical System LS1 and Logical System LS2 receive packets destined for networks not specified in their routing tables, those packets will be sent to Logical System LS3 for further processing. This configuration assumes that Logical System LS3 has a connection to an ISP or another external network.

SEE ALSO

Example: Creating an Interface on a Logical System | 19

Example: Configuring a Conditional OSPF Default Route Policy on Logical Systems

IN THIS SECTION

- Requirements | 310
- Overview | 310
- Configuration | 311
- Verification | 317

This example shows how to configure a conditional default route on one logical system and inject the default route into OSPF area 0.

Requirements

Before you begin:

- Connect the logical systems by using logical tunnel (**lt**) interfaces. See "Example: Connecting Logical Systems Within the Same Device Using Logical Tunnel Interfaces on MX Series Routers and EX Series Switches" on page 27.
- Enable OSPF on the interfaces. See *Example: Configuring OSPF on Logical Systems Within the Same Router*.

Overview

IN THIS SECTION

- Topology | 311

In this example, OSPF area 0 contains three logical systems that are configured on a single physical router. Logical System LS3 has a BGP session with an external peer, for example, an ISP.

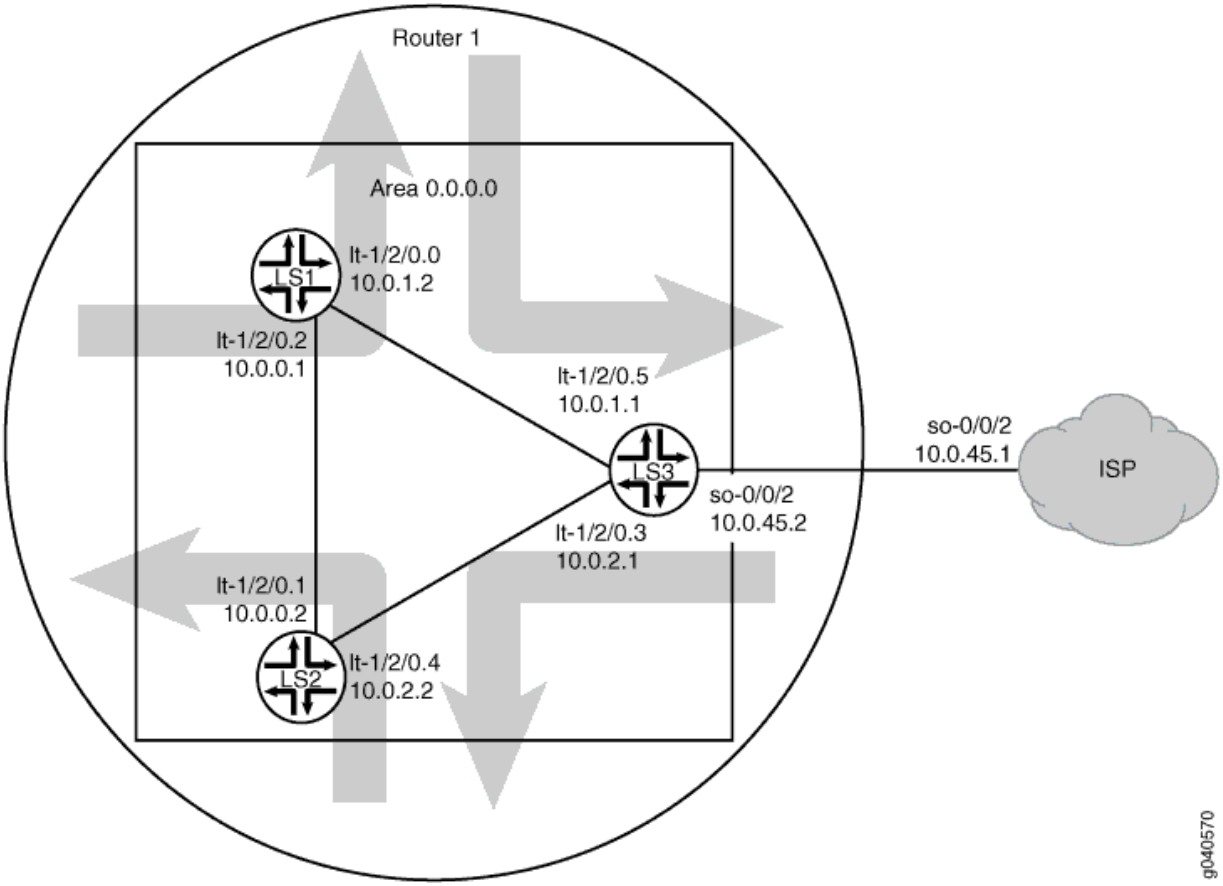
The ISP injects a default static route into BGP, which provides the customer network with a default static route to reach external networks. Logical System LS3 exports the default route into OSPF. The route policy on Logical System LS3 is conditional such that if the connection to the external peer goes

down, the default route is no longer active in the routing tables of the logical systems in area 0. This policy prevents null-route filtering of traffic. Null-route Filtering occurs when packets are dropped without notification.

Topology

Figure 23 on page 311 shows the sample network.

Figure 23: OSPF with a Conditional Default Route to an ISP



Configuration

IN THIS SECTION

- CLI Quick Configuration | 312
- Procedure | 314

- Results | 316

CLI Quick Configuration

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

Device LS1

```
set logical-systems LS1 interfaces lt-1/2/0 unit 0 description LS1->LS3
set logical-systems LS1 interfaces lt-1/2/0 unit 0 encapsulation ethernet
set logical-systems LS1 interfaces lt-1/2/0 unit 0 peer-unit 5
set logical-systems LS1 interfaces lt-1/2/0 unit 0 family inet address 10.0.1.2/30
set logical-systems LS1 interfaces lt-1/2/0 unit 2 description LS1->LS2
set logical-systems LS1 interfaces lt-1/2/0 unit 2 encapsulation ethernet
set logical-systems LS1 interfaces lt-1/2/0 unit 2 peer-unit 1
set logical-systems LS1 interfaces lt-1/2/0 unit 2 family inet address 10.0.0.1/30
set logical-systems LS1 protocols ospf area 0.0.0.0 interface lt-1/2/0.0
set logical-systems LS1 protocols ospf area 0.0.0.0 interface lt-1/2/0.2
```

Device LS2

```
set logical-systems LS2 interfaces lt-1/2/0 unit 1 description LS2->LS1
set logical-systems LS2 interfaces lt-1/2/0 unit 1 encapsulation ethernet
set logical-systems LS2 interfaces lt-1/2/0 unit 1 peer-unit 2
set logical-systems LS2 interfaces lt-1/2/0 unit 1 family inet address 10.0.0.2/30
set logical-systems LS2 interfaces lt-1/2/0 unit 4 description LS2->LS3
set logical-systems LS2 interfaces lt-1/2/0 unit 4 encapsulation ethernet
set logical-systems LS2 interfaces lt-1/2/0 unit 4 peer-unit 3
set logical-systems LS2 interfaces lt-1/2/0 unit 4 family inet address 10.0.2.2/30
set logical-systems LS2 protocols ospf area 0.0.0.0 interface lt-1/2/0.1
set logical-systems LS2 protocols ospf area 0.0.0.0 interface lt-1/2/0.4
```

Device LS3

```
set logical-systems LS3 interfaces lt-1/2/0 unit 3 description LS3->LS2
set logical-systems LS3 interfaces lt-1/2/0 unit 3 encapsulation ethernet
set logical-systems LS3 interfaces lt-1/2/0 unit 3 peer-unit 4
set logical-systems LS3 interfaces lt-1/2/0 unit 3 family inet address 10.0.2.1/30
set logical-systems LS3 interfaces lt-1/2/0 unit 5 description LS3->LS1
set logical-systems LS3 interfaces lt-1/2/0 unit 5 encapsulation ethernet
set logical-systems LS3 interfaces lt-1/2/0 unit 5 peer-unit 0
set logical-systems LS3 interfaces lt-1/2/0 unit 5 family inet address 10.0.1.1/30
set logical-systems LS3 interfaces so-0/0/2 unit 0 description LS3->ISP
set logical-systems LS3 interfaces so-0/0/2 unit 0 family inet address 10.0.45.2/30
set logical-systems LS3 protocols bgp group ext type external
set logical-systems LS3 protocols bgp group ext peer-as 64500
set logical-systems LS3 protocols bgp group ext neighbor 10.0.45.1
set logical-systems LS3 protocols ospf export gendefault
set logical-systems LS3 protocols ospf area 0.0.0.0 interface lt-1/2/0.5
set logical-systems LS3 protocols ospf area 0.0.0.0 interface lt-1/2/0.3
set logical-systems LS3 policy-options policy-statement gendefault term upstreamroutes from
protocol bgp
set logical-systems LS3 policy-options policy-statement gendefault term upstreamroutes from as-
path upstream
set logical-systems LS3 policy-options policy-statement gendefault term upstreamroutes from
route-filter 0.0.0.0/0 upto /16
set logical-systems LS3 policy-options policy-statement gendefault term upstreamroutes then next-
hop 10.0.45.1
set logical-systems LS3 policy-options policy-statement gendefault term upstreamroutes then
accept
set logical-systems LS3 policy-options policy-statement gendefault term end then reject
set logical-systems LS3 policy-options as-path upstream "^64500 "
set logical-systems LS3 routing-options generate route 0.0.0.0/0 policy gendefault
set logical-systems LS3 routing-options autonomous-system 64501
```

Device ISP

```
set interfaces so-0/0/2 unit 0 family inet address 10.0.45.1/30
set protocols bgp group ext type external
set protocols bgp group ext export advertise-default
set protocols bgp group ext peer-as 64501
set protocols bgp group ext neighbor 10.0.45.2
set policy-options policy-statement advertise-default term 1 from route-filter 0.0.0.0/0 exact
```

```
set policy-options policy-statement advertise-default term 1 then accept
set routing-options static route 0.0.0.0/0 discard
set routing-options autonomous-system 64500
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure a conditional default route:

1. Configure the interfaces.

```
[edit logical-systems LS3 interfaces]
user@R3# set lt-1/2/0 unit 3 description LS3->LS2
user@R3# set lt-1/2/0 unit 3 encapsulation ethernet
user@R3# set lt-1/2/0 unit 3 peer-unit 4
user@R3# set lt-1/2/0 unit 3 family inet address 10.0.2.1/30
user@R3# set lt-1/2/0 unit 5 description LS3->LS1
user@R3# set lt-1/2/0 unit 5 encapsulation ethernet
user@R3# set lt-1/2/0 unit 5 peer-unit 0
user@R3# set lt-1/2/0 unit 5 family inet address 10.0.1.1/30
user@R3# set so-0/0/2 unit 0 description LS3->ISP
user@R3# set so-0/0/2 unit 0 encapsulation ethernet
user@R3# set so-0/0/2 unit 0 peer-unit 7
user@R3# set so-0/0/2 unit 0 family inet address 10.0.45.2/30
```

2. Configure the autonomous system (AS) number.

```
[edit logical-systems LS3 routing-options]
user@R3# set autonomous-system 64501
```

3. Configure the BGP session with the ISP device.

```
[edit logical-systems LS3 protocols bgp group ext]
user@R3# set type external
```

```
user@R3# set peer-as 64500
user@R3# set neighbor 10.0.45.1
```

4. Configure OSPF.

```
[edit logical-systems LS3 protocols ospf area 0.0.0.0]
user@R3# set interface lt-1/2/0.5
user@R3# set interface lt-1/2/0.3
```

5. Configure the routing policy.

```
[edit logical-systems LS3 policy-options policy-statement gendefault]
user@R3# set term upstreamroutes from protocol bgp
user@R3# set term upstreamroutes from as-path upstream
user@R3# set term upstreamroutes from route-filter 0.0.0.0/0 upto /16
user@R3# set term upstreamroutes then next-hop 10.0.45.1
user@R3# set term upstreamroutes then accept
user@R3# set term end then reject
[edit logical-systems LS3 policy-options]
user@R3# set as-path upstream "^64500 "
```

6. Configure the generated route.

```
[edit logical-systems LS3 routing-options]
user@R3# set generate route 0.0.0.0/0 policy gendefault
```

7. Apply the export policy to OSPF.

```
[edit logical-systems LS3 protocols ospf]
user@R3# set export gendefault
```

8. If you are done configuring the device, commit the configuration.

```
[edit]
user@R3# commit
```

Results

Confirm your configuration by issuing the `show logical-systems LS3` command.

```
show logical-systems LS3
interfaces {
    lt-1/2/0 {
        unit 3 {
            description LS3->LS2;
            encapsulation ethernet;
            peer-unit 4;
            family inet {
                address 10.0.2.1/30;
            }
        }
        unit 5 {
            description LS3->LS1;
            encapsulation ethernet;
            peer-unit 0;
            family inet {
                address 10.0.1.1/30;
            }
        }
        unit 6 {
            description LS3->ISP;
            encapsulation ethernet;
            peer-unit 7;
            family inet {
                address 10.0.45.2/30;
            }
        }
    }
}
protocols {
    bgp {
        group ext {
            type external;
            peer-as 64500;
            neighbor 10.0.45.1;
        }
    }
    ospf {
```

```
        export gendefault;
        area 0.0.0.0 {
            interface lt-1/2/0.5;
            interface lt-1/2/0.3;
        }
    }
}
policy-options {
    policy-statement gendefault {
        term upstreamroutes {
            from {
                protocol bgp;
                as-path upstream;
                route-filter 0.0.0.0/0 upto /16;
            }
            then {
                next-hop 10.0.45.1;
                accept;
            }
        }
        term end {
            then reject;
        }
    }
    as-path upstream "^64500 ";
}
routing-options {
    generate {
        route 0.0.0.0/0 policy gendefault;
    }
    autonomous-system 64501;
}
```

Verification

IN THIS SECTION

- Verifying that the Route to the ISP Is Working | **318**
- Verifying That the Static Route Is Redistributed | **318**
- Testing the Policy Condition | **320**

Confirm that the configuration is working properly.

Verifying that the Route to the ISP Is Working

Purpose

Make sure connectivity is established between Logical System LS3 and the ISP's router.

Action

```
user@host>set cli logical-system LS3
Logical system: LS3
user@host:LS3>ping 10.0.45.1
PING 10.0.45.1 (10.0.45.1): 56 data bytes
64 bytes from 10.0.45.1: icmp_seq=0 ttl=64 time=1.185 ms
64 bytes from 10.0.45.1: icmp_seq=1 ttl=64 time=1.199 ms
64 bytes from 10.0.45.1: icmp_seq=2 ttl=64 time=1.186 ms
```

Meaning

The `ping` command confirms reachability.

Verifying That the Static Route Is Redistributed

Purpose

Make sure that the BGP policy is redistributing the static route into Logical System LS3's routing table. Also make sure that the OSPF policy is redistributing the static route into the routing tables of Logical System LS1 and Logical System LS2.

Action

```
user@host> show route logical-system LS3 protocol bgp

inet.0: 9 destinations, 10 routes (9 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[BGP/170] 00:00:25, localpref 100
```

```
                    AS path: 64500 I
                  > to 10.0.45.1 via so-0/0/2.0
```

```
user@host> show route logical-system LS1 protocol ospf

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

0.0.0.0/0          *[OSPF/150] 00:03:58, metric 0, tag 0
                    > to 10.0.1.1 via lt-1/2/0.0
10.0.2.0/30        *[OSPF/10] 03:37:45, metric 2
                      to 10.0.1.1 via lt-1/2/0.0
                    > to 10.0.0.2 via lt-1/2/0.2
224.0.0.5/32       *[OSPF/10] 03:38:41, metric 1
                      MultiRecv
```

```
user@host> show route logical-system LS2 protocol ospf
inet.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[OSPF/150] 00:04:04, metric 0, tag 0
                    > to 10.0.2.1 via lt-1/2/0.4
10.0.1.0/30        *[OSPF/10] 03:37:46, metric 2
                      to 10.0.0.1 via lt-1/2/0.1
                    > to 10.0.2.1 via lt-1/2/0.4
224.0.0.5/32       *[OSPF/10] 03:38:47, metric 1
                      MultiRecv
```

Meaning

The routing tables contain the default 0.0.0.0/0 route. If Logical System LS1 and Logical System LS2 receive packets destined for networks not specified in their routing tables, those packets will be sent to Logical System LS3 for further processing. If Logical System LS3 receives packets destined for networks not specified in its routing table, those packets will be sent to the ISP for further processing.

Testing the Policy Condition

Purpose

Deactivate the interface to make sure that the route is removed from the routing tables if the external network becomes unreachable.

Action

```
user@host> deactivate logical-systems LS3 interfaces so-0/0/2 unit 0 family inet address
10.0.45.2/30
user@host> commit
```

```
user@host> show route logical-system LS1 protocol ospf

inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.2.0/30        *[OSPF/10] 03:41:48, metric 2
                      to 10.0.1.1 via lt-1/2/0.0
                    > to 10.0.0.2 via lt-1/2/0.2
224.0.0.5/32       *[OSPF/10] 03:42:44, metric 1
                      MultiRecv

user@host> show route logical-system LS2 protocol ospf
inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.1.0/30        *[OSPF/10] 03:42:10, metric 2
                      to 10.0.0.1 via lt-1/2/0.1
                    > to 10.0.2.1 via lt-1/2/0.4
224.0.0.5/32       *[OSPF/10] 03:43:11, metric 1
                      MultiRecv
```

Meaning

The routing tables on Logical System LS1 and Logical System LS2 do not contain the default 0.0.0.0/0. This verifies that the default route is no longer present in the OSPF domain. To reactivate the

so-0/0/2.0 interface, issue the **activate logical-systems LS3 interfaces so-0/0/2 unit 0 family inet address 10.0.45.2/30** configuration-mode command.

SEE ALSO

Example: Creating an Interface on a Logical System | 19

Example: Configuring an OSPF Import Policy on Logical Systems

IN THIS SECTION

- Requirements | 321
- Overview | 321
- Configuration | 323
- Verification | 329

This example shows how to configure an OSPF import policy on logical systems. OSPF import policies apply to external routes only. An external route is a route that is outside the OSPF AS.

Requirements

This example shows logical systems that are configured within a single physical router. The logical systems connect to each other by using logical tunnel (**lt**) interfaces. See "Example: Connecting Logical Systems Within the Same Device Using Logical Tunnel Interfaces on MX Series Routers and EX Series Switches" on page 27. Alternatively, you can use multiple physical routers.

Overview

IN THIS SECTION

- Topology | 322

External routes are learned by Autonomous System Border Routers (ASBRs). External routes can be advertised throughout the OSPF domain if you configure the ASBR to redistribute the route into OSPF. An external route might be learned by the ASBR from a routing protocol other than OSPF, or the external route might be a static route that you configure on the ASBR.

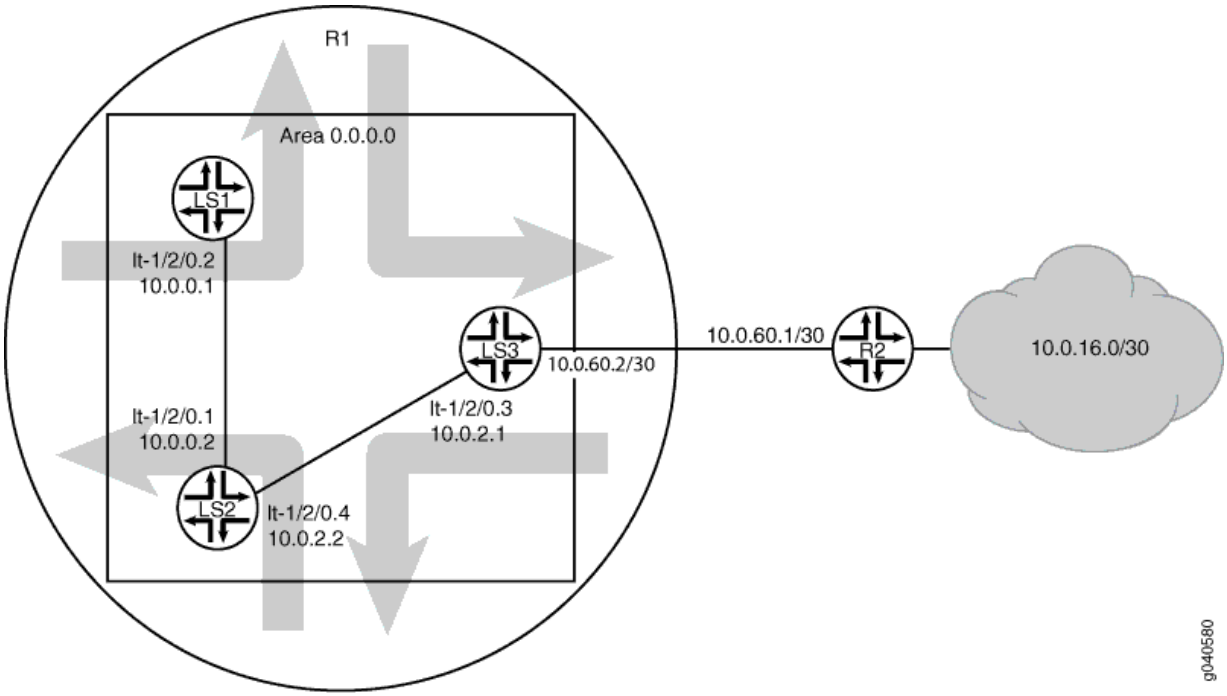
OSPF import policy allows you to prevent external routes from being added to the routing tables of OSPF neighbors. The import policy does not impact the OSPF database. This means that the import policy has no impact on the link-state advertisements.

OSPF import policies have practical applications. Suppose, for example, that you are using OSPF to advertise a static route to the devices in your datacenter because you want some of the devices in the datacenter to use the static route. However, you want other devices in the datacenter to ignore the static route. So, you apply the OSPF import policy on the devices that you want to ignore the static route. The filtering is done only on external routes in OSPF. The intra-area and inter-area routes are not considered for filtering. The default action is to accept the route when the route does not match the policy.

Topology

Figure 24 on page 322 shows the sample network.

Figure 24: OSPF Import Policy on Logical Systems



In this example, the logical systems operate as follows:

1. LS3—Logical System LS3 has a static route to the 10.0.16.0/30 network. The next hop for the static route is 10.0.60.1. LS3 has an OSPF export policy configured. The export policy redistributes static routes from LS3's routing table into LS3's OSPF database. Because the static route is in LS3's OSPF database, the route is advertised in a link state advertisement (LSA) to LS3's OSPF neighbor. LS3's OSPF neighbor is Logical System LS2.

2. LS2—Logical System LS2 receives the route advertisement from LS3. LS2 then installs the route into LS2's OSPF database. LS2 has an OSPF import policy configured that matches the static route to the 10.0.16.0/30 network and prevents the static route from being installed in LS2's routing table. However, because the route is in LS2's OSPF database, LS2 advertises the route to its OSPF neighbor, Logical System LS1.

3. LS1—Logical System LS1 receives the route advertisement from LS2. LS1 then installs the route into LS1's OSPF database. LS1 does not have an OSPF import policy configured that matches the static route to the 10.0.16.0/30 network . Therefore, the route gets installed in LS1's routing table.

Configuration

IN THIS SECTION

- CLI Quick Configuration | 323
- Procedure | 324
- Results | 326

CLI Quick Configuration

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

LS3

```
set logical-systems LS3 interfaces so-0/0/0 unit 0 family inet address 10.0.60.2/30
set logical-systems LS3 interfaces lt-1/2/0 unit 3 description LS3->LS2
set logical-systems LS3 interfaces lt-1/2/0 unit 3 encapsulation ethernet
set logical-systems LS3 interfaces lt-1/2/0 unit 3 peer-unit 4
set logical-systems LS3 interfaces lt-1/2/0 unit 3 family inet address 10.0.2.1/30
set logical-systems LS3 protocols ospf export export_static
set logical-systems LS3 protocols ospf area 0.0.0.0 interface lt-1/2/0.3
set logical-systems LS3 policy-options policy-statement export_static from protocol static
```

```
set logical-systems LS3 policy-options policy-statement export_static then accept
set logical-systems LS3 routing-options static route 10.0.16.0/30 next-hop 10.0.60.1
```

LS2

```
set logical-systems LS2 interfaces lt-1/2/0 unit 1 description LS2->LS1
set logical-systems LS2 interfaces lt-1/2/0 unit 1 encapsulation ethernet
set logical-systems LS2 interfaces lt-1/2/0 unit 1 peer-unit 2
set logical-systems LS2 interfaces lt-1/2/0 unit 1 family inet address 10.0.0.2/30
set logical-systems LS2 interfaces lt-1/2/0 unit 4 description LS2->LS3
set logical-systems LS2 interfaces lt-1/2/0 unit 4 encapsulation ethernet
set logical-systems LS2 interfaces lt-1/2/0 unit 4 peer-unit 3
set logical-systems LS2 interfaces lt-1/2/0 unit 4 family inet address 10.0.2.2/30
set logical-systems LS2 protocols ospf import filter_routes
set logical-systems LS2 protocols ospf area 0.0.0.0 interface lt-1/2/0.1
set logical-systems LS2 protocols ospf area 0.0.0.0 interface lt-1/2/0.4
set logical-systems LS2 policy-options policy-statement filter_routes from route-filter
10.0.16.0/30 exact
set logical-systems LS2 policy-options policy-statement filter_routes then reject
```

LS1

```
set logical-systems LS1 interfaces lt-1/2/0 unit 2 description LS1->LS2
set logical-systems LS1 interfaces lt-1/2/0 unit 2 encapsulation ethernet
set logical-systems LS1 interfaces lt-1/2/0 unit 2 peer-unit 1
set logical-systems LS1 interfaces lt-1/2/0 unit 2 family inet address 10.0.0.1/30
set logical-systems LS1 protocols ospf area 0.0.0.0 interface lt-1/2/0.2
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure an OSPF import policy on logical systems:

1. Configure the interfaces.

```
[edit]
user@R1# set logical-systems LS3 interfaces so-0/0/0 unit 0 family inet address 10.0.60.2/30
user@R1# set logical-systems LS3 interfaces lt-1/2/0 unit 3 description LS3->LS2
user@R1# set logical-systems LS3 interfaces lt-1/2/0 unit 3 encapsulation ethernet
user@R1# set logical-systems LS3 interfaces lt-1/2/0 unit 3 peer-unit 4
user@R1# set logical-systems LS3 interfaces lt-1/2/0 unit 3 family inet address 10.0.2.1/30
user@R1# set logical-systems LS2 interfaces lt-1/2/0 unit 1 description LS2->LS1
user@R1# set logical-systems LS2 interfaces lt-1/2/0 unit 1 encapsulation ethernet
user@R1# set logical-systems LS2 interfaces lt-1/2/0 unit 1 peer-unit 2
user@R1# set logical-systems LS2 interfaces lt-1/2/0 unit 1 family inet address 10.0.0.2/30
user@R1# set logical-systems LS2 interfaces lt-1/2/0 unit 4 description LS2->LS3
user@R1# set logical-systems LS2 interfaces lt-1/2/0 unit 4 encapsulation ethernet
user@R1# set logical-systems LS2 interfaces lt-1/2/0 unit 4 peer-unit 3
user@R1# set logical-systems LS2 interfaces lt-1/2/0 unit 4 family inet address 10.0.2.2/30
user@R1# set logical-systems LS1 interfaces lt-1/2/0 unit 2 description LS1->LS2
user@R1# set logical-systems LS1 interfaces lt-1/2/0 unit 2 encapsulation ethernet
user@R1# set logical-systems LS1 interfaces lt-1/2/0 unit 2 peer-unit 1
user@R1# set logical-systems LS1 interfaces lt-1/2/0 unit 2 family inet address 10.0.0.1/30
```

2. Enable OSPF on the interfaces.

```
[edit]
user@R1# set logical-systems LS3 protocols ospf area 0.0.0.0 interface lt-1/2/0.3
user@R1# set logical-systems LS2 protocols ospf area 0.0.0.0 interface lt-1/2/0.1
user@R1# set logical-systems LS2 protocols ospf area 0.0.0.0 interface lt-1/2/0.4
user@R1# set logical-systems LS1 protocols ospf area 0.0.0.0 interface lt-1/2/0.2
```

3. Configure the static route on Logical System LS3.

```
[edit]
user@R1# set logical-systems LS3 routing-options static route 10.0.16.0/30 next-hop 10.0.60.1
```

4. On Logical System LS3, redistribute the static route into OSPF.

```
[edit]
user@R1# set logical-systems LS3 protocols ospf export export_static
user@R1# set logical-systems LS3 policy-options policy-statement export_static from protocol
```

```
static
user@R1# set logical-systems LS3 policy-options policy-statement export_static then accept
```

5. On Logical System LS2, configure the OSPF import policy.

```
[edit]
user@R1# set logical-systems LS2 protocols ospf import filter_routes
user@R1# set logical-systems LS2 policy-options policy-statement filter_routes from route-
filter 10.0.16.0/30 exact
user@R1# set logical-systems LS2 policy-options policy-statement filter_routes then reject
```

6. If you are done configuring the device, commit the configuration.

```
[edit]
user@R1# commit
```

Results

Confirm your configuration by issuing the `show logical-systems` command.

```
user@R1# show logical-systems
LS1 {
    interfaces {
        lt-1/2/0 {
            unit 2 {
                description LS1->LS2;
                encapsulation ethernet;
                peer-unit 1;
                family inet {
                    address 10.0.0.1/30;
                }
            }
        }
    }
    protocols {
        ospf {
            area 0.0.0.0 {
                interface lt-1/2/0.2;
            }
```

```
        }
    }
}
LS2 {
    interfaces {
        lt-1/2/0 {
            unit 1 {
                description LS2->LS1;
                encapsulation ethernet;
                peer-unit 2;
                family inet {
                    address 10.0.0.2/30;
                }
            }
            unit 4 {
                description LS2->LS3;
                encapsulation ethernet;
                peer-unit 3;
                family inet {
                    address 10.0.2.2/30;
                }
            }
        }
    }
    protocols {
        ospf {
            import filter_routes;
            area 0.0.0.0 {
                interface lt-1/2/0.1;
                interface lt-1/2/0.4;
            }
        }
    }
    policy-options {
        policy-statement filter_routes {
            from {
                route-filter 10.0.16.0/30 exact;
            }
            then reject;
        }
    }
}
LS3 {
```

```
    interfaces {
        so-0/0/0 {
            unit 0 {
                family inet {
                    address 10.0.60.2/30;
                }
            }
        }
        lt-1/2/0 {
            unit 3 {
                description LS3->LS2;
                encapsulation ethernet;
                peer-unit 4;
                family inet {
                    address 10.0.2.1/30;
                }
            }
        }
    }
    protocols {
        ospf {
            export export_static;
            area 0.0.0.0 {
                interface lt-1/2/0.3;
            }
        }
    }
    policy-options {
        policy-statement export_static {
            from protocol static;
            then accept;
        }
    }
    routing-options {
        static {
            route 10.0.16.0/30 next-hop 10.0.60.1;
        }
    }
}
```

Verification

IN THIS SECTION

- Viewing the OSPF Databases of the Logical Systems | 329
- Viewing the Routing Tables of the Logical Systems | 330

Confirm that the configuration is working properly.

Viewing the OSPF Databases of the Logical Systems

Purpose

Verify that OSPF is advertising the static route.

Action

```
user@R1> show ospf database logical-system all
logical-system: LS2

    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router   10.0.0.1         10.0.0.1         0x8000001f   107  0x22 0x8f59  36
Router  *10.0.0.2         10.0.0.2         0x80000025   101  0x22 0x4074  48
Router   10.0.2.1         10.0.2.1         0x80000018   107  0x22 0xab3a  36
Network  10.0.0.1         10.0.0.1         0x80000001   107  0x22 0x7b94  32
Network  10.0.2.1         10.0.2.1         0x8000000c   190  0x22 0x53ab  32
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Extern   10.0.16.0        10.0.2.1         0x80000007  1785  0x22 0x4147  36
-----

logical-system: LS1

    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router  *10.0.0.1         10.0.0.1         0x8000001f   107  0x22 0x8f59  36
Router   10.0.0.2         10.0.0.2         0x80000025   103  0x22 0x4074  48
```

```
Router   10.0.2.1         10.0.2.1         0x80000018   109  0x22 0xab3a  36
Network *10.0.0.1         10.0.0.1         0x80000001   107  0x22 0x7b94  32
Network  10.0.2.1         10.0.2.1         0x8000000c   192  0x22 0x53ab  32

    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Extern   10.0.16.0        10.0.2.1         0x80000007  1787  0x22 0x4147  36
-----

logical-system: LS3

    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router   10.0.0.1         10.0.0.1         0x8000001f   109  0x22 0x8f59  36
Router   10.0.0.2         10.0.0.2         0x80000025   103  0x22 0x4074  48
Router  *10.0.2.1         10.0.2.1         0x80000018   107  0x22 0xab3a  36
Network  10.0.0.1         10.0.0.1         0x80000001   109  0x22 0x7b94  32
Network *10.0.2.1         10.0.2.1         0x8000000c   190  0x22 0x53ab  32

    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Extern  *10.0.16.0        10.0.2.1         0x80000007  1785  0x22 0x4147  36
...
```

Meaning

The **Extern *10.0.16.0** output shows that OSPF is advertising the external route.

Viewing the Routing Tables of the Logical Systems

Purpose

Make sure that Logical System LS3 and Logical System LS1 have the route to the 10.0.16.0/30 network installed in their respective routing tables. Make sure that Logical System LS2 does not have the route installed in its routing table.

Action

```
user@R1> show route logical-system all
logical-system: LS2
```

```
inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.0/30        *[Direct/0] 04:22:19
                    > via lt-1/2/0.1
10.0.0.2/32        *[Local/0] 04:22:19
                      Local via lt-1/2/0.1
10.0.2.0/30        *[Direct/0] 04:22:19
                    > via lt-1/2/0.4
10.0.2.2/32        *[Local/0] 04:22:19
                      Local via lt-1/2/0.4
224.0.0.5/32       *[OSPF/10] 04:22:23, metric 1
                      MultiRecv
-----

logical-system: LS1

inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.0/30        *[Direct/0] 04:22:19
                    > via lt-1/2/0.2
10.0.0.1/32        *[Local/0] 04:22:19
                      Local via lt-1/2/0.2
10.0.2.0/30        *[OSPF/10] 00:07:52, metric 2
                    > to 10.0.0.2 via lt-1/2/0.2
10.0.16.0/30       *[OSPF/150] 00:07:52, metric 0, tag 0
                    > to 10.0.0.2 via lt-1/2/0.2
224.0.0.5/32       *[OSPF/10] 04:22:23, metric 1
                      MultiRecv
-----

logical-system: LS3

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

10.0.0.0/30        *[OSPF/10] 00:07:57, metric 2
                    > to 10.0.2.2 via lt-1/2/0.3
10.0.2.0/30        *[Direct/0] 04:22:19
                    > via lt-1/2/0.3
10.0.2.1/32        *[Local/0] 04:22:19
                      Local via lt-1/2/0.3
```

```
10.0.16.0/30       *[Static/5] 03:51:18
                    > to 10.0.60.1 via so-0/0/0.0
10.0.60.0/30       *[Direct/0] 03:53:52
                    > via so-0/0/0.0
10.0.60.2/32       *[Local/0] 03:53:58
                      Local via so-0/0/0.0
224.0.0.5/32       *[OSPF/10] 04:22:23, metric 1
                      MultiRecv
```

Meaning

The route to 10.0.16.0/30 is not installed in Logical System LS2's routing table. The route to 10.0.16.0/30 is installed in Logical System LS1's routing table as a route learned from OSPF. Because it is an OSPF external route, it has a preference value of 150 (instead of 10). By default, routes resulting from OSPF external LSAs are installed with a preference value of 150. The route to 10.0.16.0/30 is installed in Logical System LS3's routing table as a static route.

SEE ALSO

Example: Connecting Logical Systems Within the Same Device Using Logical Tunnel Interfaces on MX Series Routers and EX Series Switches | **27**

Example: Creating an Interface on a Logical System | **19**

RELATED DOCUMENTATION

Understanding Logical Systems for Routers and Switches | **2**

OSPF User Guide

RSVP-Signaled Point-to-Multipoint LSPs on Logical Systems

IN THIS SECTION

- Point-to-Multipoint LSPs Overview | 333
- Example: Configuring an RSVP-Signaled Point-to-Multipoint LSP on Logical Systems | 335

Logical Systems enables a physical router to act as a collection of paths for an RSVP-signaled point-to-multipoint LSP. For more information, see the following topics:

Point-to-Multipoint LSPs Overview

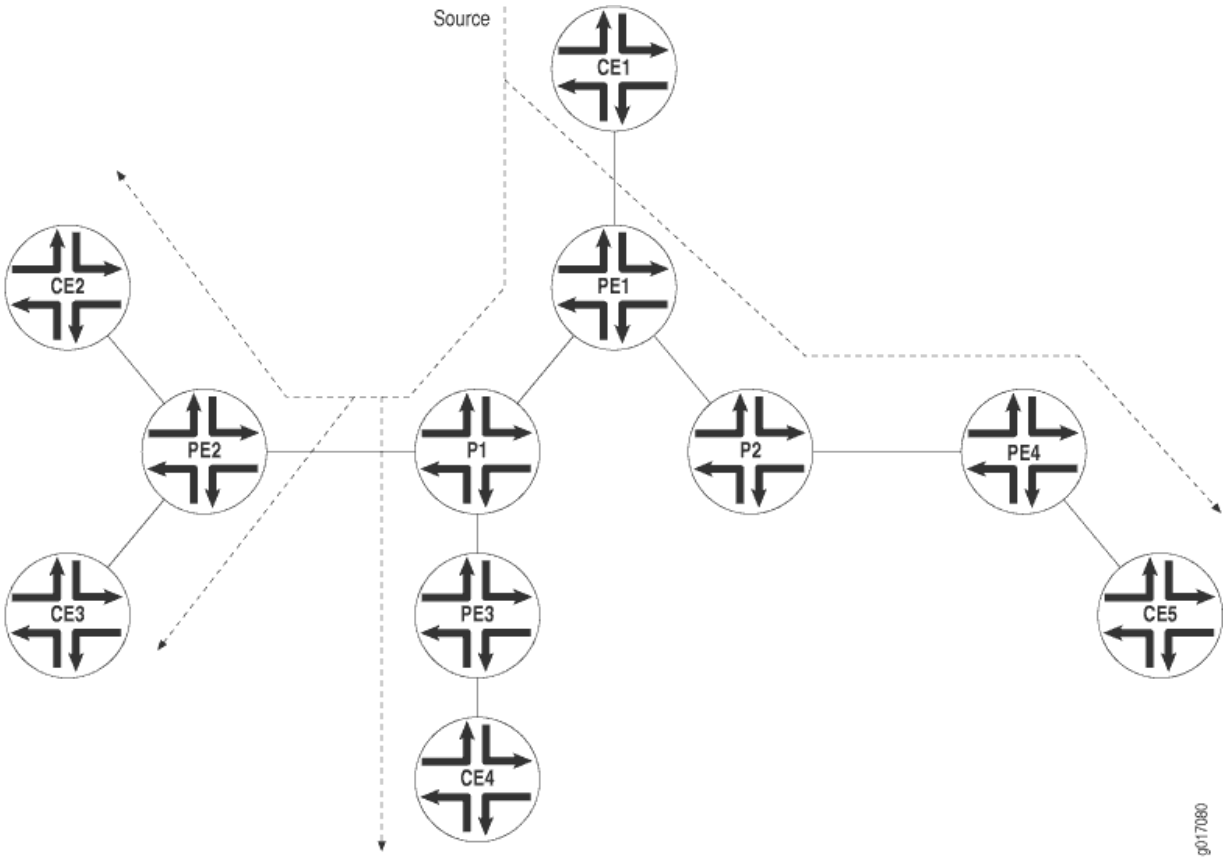
A point-to-multipoint MPLS LSP is an LSP with a single source and multiple destinations. By taking advantage of the MPLS packet replication capability of the network, point-to-multipoint LSPs avoid unnecessary packet replication at the ingress router. Packet replication takes place only when packets are forwarded to two or more different destinations requiring different network paths.

This process is illustrated in Figure 25 on page 334. Router PE1 is configured with a point-to-multipoint LSP to Routers PE2, PE3, and PE4. When Router PE1 sends a packet on the point-to-multipoint LSP to Routers P1 and P2, Router P1 replicates the packet and forwards it to Routers PE2 and PE3. Router P2 sends the packet to Router PE4.

This feature is described in detail in the Internet drafts draft-raggarwa-mpls-p2mp-te-02.txt (expired February 2004), *Establishing Point to Multipoint MPLS TE LSPs*, draft-ietf-mpls-rsvp-te-p2mp-02.txt, *Extensions to Resource Reservation Protocol-Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label-Switched Paths (LSPs)*, and RFC 6388, *Label Distribution Protocol Extensions for Point-to-*

Multipoint and Multipoint-to-Multipoint Label Switched Paths (only point-to-multipoint LSPs are supported).

Figure 25: Point-to-Multipoint LSPs



The following are some of the properties of point-to-multipoint LSPs:

- A point-to-multipoint LSP enables you to use MPLS for point-to-multipoint data distribution. This functionality is similar to that provided by IP multicast.
- You can add and remove branch LSPs from a main point-to-multipoint LSP without disrupting traffic. The unaffected parts of the point-to-multipoint LSP continue to function normally.
- You can configure a node to be both a transit and an egress router for different branch LSPs of the same point-to-multipoint LSP.
- You can enable link protection on a point-to-multipoint LSP. Link protection can provide a bypass LSP for each of the branch LSPs that make up the point-to-multipoint LSP. If any of the primary paths fail, traffic can be quickly switched to the bypass.

- You can configure branch LSPs either statically, dynamically, or as a combination of static and dynamic LSPs.
- You can enable *graceful Routing Engine switchover* (GRES) and graceful restart for point-to-multipoint LSPs at ingress and egress routers. The point-to-multipoint LSPs must be configured using either static routes or circuit cross-connect (CCC). GRES and graceful restart allow the traffic to be forwarded at the Packet Forwarding Engine based on the old state while the control plane recovers. Feature parity for GRES and graceful restart for MPLS point-to-multipoint LSPs on the Junos Trio chipset is supported in Junos OS Releases 11.1R2, 11.2R2, and 11.4.

Example: Configuring an RSVP-Signaled Point-to-Multipoint LSP on Logical Systems

IN THIS SECTION

- Requirements | 335
- Overview | 336
- Configuration | 337
- Verification | 363

In this example, multiple logical systems in a physical router act as a collection of paths for an RSVP-signaled point-to-multipoint LSP. The logical systems are chained together and connected internally over a series of logical tunnel (`lt`) interfaces.

Requirements

This example uses the following hardware and software components:

- One MX Series router running logical systems. You do not need to use an MX Series router for the logical systems. You can use any Juniper Networks router that supports logical systems.
- On the MX Series router, the logical systems are connected using logical tunnel (`lt`) interfaces. For more information, see "Example: Connecting Logical Systems Within the Same Router Using Logical Tunnel Interfaces" on page 33 and "Example: Connecting Logical Systems Within the Same Device Using Logical Tunnel Interfaces on MX Series Routers and EX Series Switches" on page 27. An alternative to using `lt` interfaces is to create external back-to-back interconnections between ports on the router.

- Four customer-edge (CE) devices running in separate physical devices. You do not need to use routers for the CE devices. For example, the CE devices can be EX Series Ethernet Switches.
- Junos OS Release 12.1 or later running on the MX Series router.

On M Series Multiservice Edge and T Series Core Routers, you can create an `lt` interface if you have a Tunnel Services PIC installed on an Enhanced FPC in your routing platform.

On M40e routers, you can create an `lt` interface if you have a Tunnel Services PIC. (An Enhanced FPC is not required.)

On an M7i router, `lt` interfaces can be created by using the integrated Adaptive Services Module.

On an MX Series router, as is shown in this example, the primary administrator can configure `lt` interfaces by including the `tunnel-services` statement at the [`edit chassis fpc` *`slot-number`* `pic` *`number`*] hierarchy level.

Overview

IN THIS SECTION

- Topology Diagram | 337

In this example, the logical systems serve as the transit, branch, and leaf nodes of a single point-to-multipoint LSP. Logical system LS1 is the ingress node. The branches go from LS1 to LS5, LS1 to LS7, and LS1 to LS4. Static unicast routes on the ingress node (LS1) point to the egress nodes.

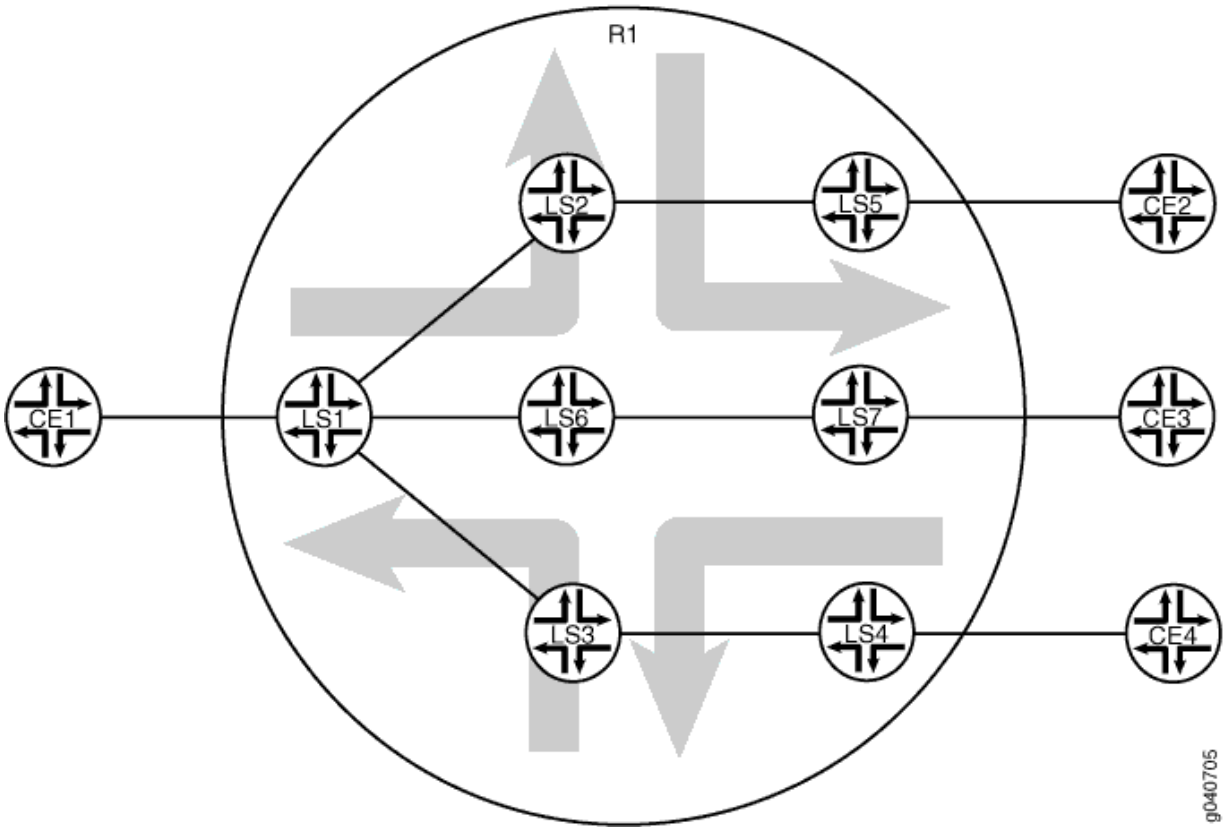
The following topologies are supported:

- A single logical system in a physical router. The logical system is one node in an RSVP-signaled point-to-multipoint LSP.
- Multiple logical systems in a physical router, with each logical system acting as a label-switched router (LSR). The multiple logical systems can be unconnected, connected to each other internally with `lt` interfaces, or connected to each other externally with back-to-back connections.
- One RSVP-signaled point-to-multipoint LSP, with some nodes being logical systems and other nodes being physical routers.

Topology Diagram

Figure 26 on page 337 shows the topology used in this example.

Figure 26: RSVP-Signaled Point-to-Multipoint LSP on Logical Systems



Configuration

IN THIS SECTION

- CLI Quick Configuration | **338**
- Configuring the MX Series Router to Support Logical Tunnel Interfaces | **340**
- Configuring the Ingress LSR (Logical System LS1) | **341**
- Configuring the Transit and Egress LSRs (Logical Systems LS2, LS3, LS4, LS5, LS6, and LS7) | **344**
- Configuring Device CE1 | **358**
- Configuring Device CE2 | **359**

Configuring Device CE3 | 360
Configuring Device CE4 | 362

CLI Quick Configuration

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

Router R1

```
set logical-systems LS1 interfaces ge-2/0/2 unit 0 description LS1-to-CE1
set logical-systems LS1 interfaces ge-2/0/2 unit 0 family inet address 10.0.244.10/30
set logical-systems LS1 interfaces lt-2/0/10 unit 1 description LS1-to-LS2
set logical-systems LS1 interfaces lt-2/0/10 unit 1 encapsulation ethernet
set logical-systems LS1 interfaces lt-2/0/10 unit 1 peer-unit 2
set logical-systems LS1 interfaces lt-2/0/10 unit 1 family inet address 2.2.2.1/24
set logical-systems LS1 interfaces lt-2/0/10 unit 1 family mpls
set logical-systems LS1 interfaces lt-2/0/10 unit 8 description LS1-to-LS6
set logical-systems LS1 interfaces lt-2/0/10 unit 8 encapsulation ethernet
set logical-systems LS1 interfaces lt-2/0/10 unit 8 peer-unit 6
set logical-systems LS1 interfaces lt-2/0/10 unit 8 family inet address 6.6.6.1/24
set logical-systems LS1 interfaces lt-2/0/10 unit 8 family mpls
set logical-systems LS1 interfaces lt-2/0/10 unit 9 description LS1-to-LS3
set logical-systems LS1 interfaces lt-2/0/10 unit 9 encapsulation ethernet
set logical-systems LS1 interfaces lt-2/0/10 unit 9 peer-unit 3
set logical-systems LS1 interfaces lt-2/0/10 unit 9 family inet address 3.3.3.1/24
set logical-systems LS1 interfaces lt-2/0/10 unit 9 family mpls
set logical-systems LS1 interfaces lo0 unit 1 family inet address 100.10.10.10/32
set logical-systems LS1 protocols rsvp interface lt-2/0/10.1
set logical-systems LS1 protocols rsvp interface lt-2/0/10.8
set logical-systems LS1 protocols rsvp interface lt-2/0/10.9
set logical-systems LS1 protocols rsvp interface lo0.1
set logical-systems LS1 protocols mpls traffic-engineering bgp-igp
set logical-systems LS1 protocols mpls label-switched-path LS1-LS5 to 100.50.50.50
set logical-systems LS1 protocols mpls label-switched-path LS1-LS5 p2mp p2mp1
set logical-systems LS1 protocols mpls label-switched-path LS1-LS7 to 100.70.70.70
set logical-systems LS1 protocols mpls label-switched-path LS1-LS7 p2mp p2mp1
set logical-systems LS1 protocols mpls label-switched-path LS1-LS4 to 100.40.40.40
```

```
set logical-systems LS1 protocols mpls label-switched-path LS1-LS4 p2mp p2mp1
set logical-systems LS1 protocols mpls interface lt-2/0/10.1
set logical-systems LS1 protocols mpls interface lt-2/0/10.8
set logical-systems LS1 protocols mpls interface lt-2/0/10.9
set logical-systems LS1 protocols mpls interface lo0.1
set logical-systems LS1 protocols ospf traffic-engineering
set logical-systems LS1 protocols ospf area 0.0.0.0 interface ge-2/0/2.0
set logical-systems LS1 protocols ospf area 0.0.0.0 interface lt-2/0/10.1
set logical-systems LS1 protocols ospf area 0.0.0.0 interface lt-2/0/10.8
set logical-systems LS1 protocols ospf area 0.0.0.0 interface lt-2/0/10.9
set logical-systems LS1 protocols ospf area 0.0.0.0 interface lo0.1
set logical-systems LS1 routing-options static route 5.5.5.0/24 p2mp-lsp-next-hop p2mp1
set logical-systems LS1 routing-options static route 7.7.7.0/24 p2mp-lsp-next-hop p2mp1
set logical-systems LS1 routing-options static route 4.4.4.0/24 p2mp-lsp-next-hop p2mp1
set logical-systems LS1 routing-options router-id 100.10.10.10
```

Device CE1

```
set interfaces ge-1/3/2 unit 0 family inet address 10.0.244.9/30
set interfaces ge-1/3/2 unit 0 description CE1-to-LS1
set routing-options static route 10.0.104.8/30 next-hop 10.0.244.10
set routing-options static route 10.0.134.8/30 next-hop 10.0.244.10
set routing-options static route 10.0.224.8/30 next-hop 10.0.244.10
```

Device CE2

```
set interfaces ge-1/3/3 unit 0 family inet address 10.0.224.9/30
set interfaces ge-1/3/3 unit 0 description CE2-to-LS5
set routing-options static route 10.0.244.8/30 next-hop 10.0.224.10
```

Device CE3

```
set interfaces ge-2/0/1 unit 0 family inet address 10.0.134.9/30
set interfaces ge-2/0/1 unit 0 description CE3-to-LS7
set routing-options static route 10.0.244.8/30 next-hop 10.0.134.10
```

Device CE4

```
set interfaces ge-3/1/3 unit 0 family inet address 10.0.104.10/30
set interfaces ge-3/1/3 unit 0 description CE4-to-LS4
set routing-options static route 10.0.244.8/30 next-hop 10.0.104.9
```

Configuring the MX Series Router to Support Logical Tunnel Interfaces

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

This procedure is required for MX Series routers only. If you have an M Series or T Series router, skip this procedure.

To enable `lt` interfaces on the MX Series router:

1. Run the `show chassis fpc` command to verify that the router has a DPC, MPC, or MIC installed and is in the online state.

   ```
   user@host> show chassis fpc
                       Temp  CPU Utilization (%)   Memory    Utilization (%)
   Slot State           (C)  Total  Interrupt      DRAM (MB) Heap     Buffer
     0  Empty
     1  Empty
     2  Online           31      3          0       1024       14     21
   ```

   This output shows that slot 0 and slot 1 are empty. Slot 2 is online.

2. Configure FPC slot 2 to support `lt` interfaces.

   ```
   [edit]
   user@host# set chassis fpc 2 pic 0 tunnel-services bandwidth 1g
   ```

   This command creates several tunnel interface types, including `gr`, `ip`, and `lt`. For this example, the important one is the `lt` interface.

3. Commit the configuration.

```
[edit]
user@host# commit
user@host# exit
```

4. Run the `show interfaces terse` command to verify that the router has an `lt` interface.

```
user@host> show interfaces terse
Interface               Admin Link Proto    Local                 Remote
...
gr-2/0/10               up    up
ip-2/0/10               up    up
lt-2/0/10               up    up
...
```

Configuring the Ingress LSR (Logical System LS1)

Step-by-Step Procedure

To configure Logical System LS1:

1. From the main router, configure the logical system.

```
[edit]
user@R1# set logical-systems LS1
```

2. Commit the configuration.

```
[edit]
user@R1# commit
user@R1# exit
```

3. Set the CLI to view the logical system.

```
user@R1> set cli logical-system LS1
Logical system: LS1
user@R1:LS1>
```

4. Configure the interfaces, interface encapsulation, and protocol families.

```
[edit]
user@R1:LS1# edit interfaces
[edit interfaces]
user@R1:LS1# set ge-2/0/2 unit 0 description R1-to-CE1
user@R1:LS1# set ge-2/0/2 unit 0 family inet address 10.0.244.10/30
user@R1:LS1# set lt-2/0/10 unit 1 description LS1-to-LS2
user@R1:LS1# set lt-2/0/10 unit 1 encapsulation ethernet
user@R1:LS1# set lt-2/0/10 unit 1 peer-unit 2
user@R1:LS1# set lt-2/0/10 unit 1 family inet address 2.2.2.1/24
user@R1:LS1# set lt-2/0/10 unit 1 family mpls
user@R1:LS1# set lt-2/0/10 unit 8 description LS1-to-LS6
user@R1:LS1# set lt-2/0/10 unit 8 encapsulation ethernet
user@R1:LS1# set lt-2/0/10 unit 8 peer-unit 6
user@R1:LS1# set lt-2/0/10 unit 8 family inet address 6.6.6.1/24
user@R1:LS1# set lt-2/0/10 unit 8 family mpls
user@R1:LS1# set lt-2/0/10 unit 9 description LS1-to-LS3
user@R1:LS1# set lt-2/0/10 unit 9 encapsulation ethernet
user@R1:LS1# set lt-2/0/10 unit 9 peer-unit 3
user@R1:LS1# set lt-2/0/10 unit 9 family inet address 3.3.3.1/24
user@R1:LS1# set lt-2/0/10 unit 9 family mpls
user@R1:LS1# set lo0 unit 1 family inet address 100.10.10.10/32
user@R1:LS1# exit
```

5. Enable RSVP, MPLS, and OSPF on the interfaces.

```
[edit]
user@R1:LS1# edit protocols
[edit protocols]
user@R1:LS1# set rsvp interface lt-2/0/10.1
user@R1:LS1# set rsvp interface lt-2/0/10.8
user@R1:LS1# set rsvp interface lt-2/0/10.9
user@R1:LS1# set rsvp interface lo0.1
```

```
user@R1:LS1# set mpls interface lt-2/0/10.1
user@R1:LS1# set mpls interface lt-2/0/10.8
user@R1:LS1# set mpls interface lt-2/0/10.9
user@R1:LS1# set mpls interface lo0.1
user@R1:LS1# set ospf area 0.0.0.0 interface ge-2/0/2.0
user@R1:LS1# set ospf area 0.0.0.0 interface lt-2/0/10.1
user@R1:LS1# set ospf area 0.0.0.0 interface lt-2/0/10.8
user@R1:LS1# set ospf area 0.0.0.0 interface lt-2/0/10.9
user@R1:LS1# set ospf area 0.0.0.0 interface lo0.1
```

6. Configure the MPLS point-to-multipoint LSPs.

```
[edit protocols]
user@R1:LS1# set mpls label-switched-path LS1-LS5 to 100.50.50.50
user@R1:LS1# set mpls label-switched-path LS1-LS5 p2mp p2mp1
user@R1:LS1# set mpls label-switched-path LS1-LS7 to 100.70.70.70
user@R1:LS1# set mpls label-switched-path LS1-LS7 p2mp p2mp1
user@R1:LS1# set mpls label-switched-path LS1-LS4 to 100.40.40.40
user@R1:LS1# set mpls label-switched-path LS1-LS4 p2mp p2mp1
```

7. Enable MPLS to perform traffic engineering for OSPF.

```
[edit protocols]
user@R1:LS1# set mpls traffic-engineering bgp-igp
user@R1:LS1# exit
```

This causes the ingress routes to be installed in the `inet.0` routing table. By default, MPLS performs traffic engineering for BGP only. You need to enable MPLS traffic engineering on the ingress LSR only.

8. Enable traffic engineering for OSPF.

```
[edit protocols]
user@R1:LS1# set ospf traffic-engineering
user@R1:LS1#  exit
```

This causes the shortest-path first (SPF) algorithm to take into account the LSPs configured under MPLS.

9. Configure the router ID.

```
[edit]
user@R1:LS1# edit routing-options
[edit routing-options]
user@R1:LS1# set router-id 100.10.10.10
```

10. Configure static IP unicast routes with the point-to-multipoint LSP name as the next hop for each route.

```
[edit routing-options]
user@R1:LS1# set static route 5.5.5.0/24p2mp-lsp-next-hop p2mp1
user@R1:LS1# set static route 7.7.7.0/24 p2mp-lsp-next-hop p2mp1
user@R1:LS1# set static route 4.4.4.0/24 p2mp-lsp-next-hop p2mp1
user@R1:LS1#  exit
```

11. If you are done configuring the device, commit the configuration.

```
[edit]
user@R1:LS1# commit
```

Configuring the Transit and Egress LSRs (Logical Systems LS2, LS3, LS4, LS5, LS6, and LS7)

Step-by-Step Procedure

To configure the transit and egress LSRs:

1. Configure the interfaces, interface encapsulation, and protocol families.

```
[edit]
user@R1# edit logical-systems
[edit logical-systems]
user@R1# set LS2 interfaces lt-2/0/10 unit 2 description LS2-to-LS1
user@R1# set LS2 interfaces lt-2/0/10 unit 2 encapsulation ethernet
user@R1# set LS2 interfaces lt-2/0/10 unit 2 peer-unit 1
user@R1# set LS2 interfaces lt-2/0/10 unit 2 family inet address 172.16.2.2/24
user@R1# set LS2 interfaces lt-2/0/10 unit 2 family mpls
user@R1# set LS2 interfaces lt-2/0/10 unit 10 description LS2-to-LS5
user@R1# set LS2 interfaces lt-2/0/10 unit 10 encapsulation ethernet
```

```
user@R1# set LS2 interfaces lt-2/0/10 unit 10 peer-unit 5
user@R1# set LS2 interfaces lt-2/0/10 unit 10 family inet address 5.5.5.1/24
user@R1# set LS2 interfaces lt-2/0/10 unit 10 family mpls
user@R1# set LS2 interfaces lo0 unit 2 family inet address 100.20.20.20/32
user@R1# set LS3 interfaces lt-2/0/10 unit 3 description LS3-to-LS1
user@R1# set LS3 interfaces lt-2/0/10 unit 3 encapsulation ethernet
user@R1# set LS3 interfaces lt-2/0/10 unit 3 peer-unit 9
user@R1# set LS3 interfaces lt-2/0/10 unit 3 family inet address 3.3.3.2/24
user@R1# set LS3 interfaces lt-2/0/10 unit 3 family mpls
user@R1# set LS3 interfaces lt-2/0/10 unit 12 description LS3-to-LS4
user@R1# set LS3 interfaces lt-2/0/10 unit 12 encapsulation ethernet
user@R1# set LS3 interfaces lt-2/0/10 unit 12 peer-unit 4
user@R1# set LS3 interfaces lt-2/0/10 unit 12 family inet address 4.4.4.1/24
user@R1# set LS3 interfaces lt-2/0/10 unit 12 family mpls
user@R1# set LS3 interfaces lo0 unit 3 family inet address 100.30.30.30/32
user@R1# set LS4 interfaces ge-2/0/0 unit 0 description R1-to-CE4
user@R1# set LS4 interfaces ge-2/0/0 unit 0 family inet address 10.0.104.9/30
user@R1# set LS4 interfaces lt-2/0/10 unit 4 description LS4-to-LS3
user@R1# set LS4 interfaces lt-2/0/10 unit 4 encapsulation ethernet
user@R1# set LS4 interfaces lt-2/0/10 unit 4 peer-unit 12
user@R1# set LS4 interfaces lt-2/0/10 unit 4 family inet address 4.4.4.2/24
user@R1# set LS4 interfaces lt-2/0/10 unit 4 family mpls
user@R1# set LS4 interfaces lo0 unit 4 family inet address 100.40.40.40/32
user@R1# set LS5 interfaces ge-2/0/3 unit 0 description LS1-to-CE1
user@R1# set LS5 interfaces ge-2/0/3 unit 0 family inet address 10.0.224.10/30
user@R1# set LS5 interfaces lt-2/0/10 unit 5 description LS5-to-LS2
user@R1# set LS5 interfaces lt-2/0/10 unit 5 encapsulation ethernet
user@R1# set LS5 interfaces lt-2/0/10 unit 5 peer-unit 10
user@R1# set LS5 interfaces lt-2/0/10 unit 5 family inet address 5.5.5.2/24
user@R1# set LS5 interfaces lt-2/0/10 unit 5 family mpls
user@R1# set LS5 interfaces lo0 unit 5 family inet address 100.50.50.50/32
user@R1# set LS6 interfaces lt-2/0/10 unit 6 description LS6-to-LS1
user@R1# set LS6 interfaces lt-2/0/10 unit 6 encapsulation ethernet
user@R1# set LS6 interfaces lt-2/0/10 unit 6 peer-unit 8
user@R1# set LS6 interfaces lt-2/0/10 unit 6 family inet address 6.6.6.2/24
user@R1# set LS6 interfaces lt-2/0/10 unit 6 family mpls
user@R1# set LS6 interfaces lt-2/0/10 unit 11 description LS6-to-LS7
user@R1# set LS6 interfaces lt-2/0/10 unit 11 encapsulation ethernet
user@R1# set LS6 interfaces lt-2/0/10 unit 11 peer-unit 7
user@R1# set LS6 interfaces lt-2/0/10 unit 11 family inet address 7.7.7.1/24
user@R1# set LS6 interfaces lt-2/0/10 unit 11 family mpls
user@R1# set LS6 interfaces lo0 unit 6 family inet address 100.60.60.60/32
user@R1# set LS7 interfaces ge-2/0/1 unit 0 description R1-to-CE3
```

```
user@R1# set LS7 interfaces ge-2/0/1 unit 0 family inet address 10.0.134.10/30
user@R1# set LS7 interfaces lt-2/0/10 unit 7 description LS7-to-LS6
user@R1# set LS7 interfaces lt-2/0/10 unit 7 encapsulation ethernet
user@R1# set LS7 interfaces lt-2/0/10 unit 7 peer-unit 11
user@R1# set LS7 interfaces lt-2/0/10 unit 7 family inet address 7.7.7.2/24
user@R1# set LS7 interfaces lt-2/0/10 unit 7 family mpls
user@R1# set LS7 interfaces lo0 unit 7 family inet address 100.70.70.70/32
```

2. Enable RSVP, MPLS, and OSPF on the interfaces.

```
[edit]
user@R1# edit logical-systems
[edit logical-systems]
user@R1# set LS2 protocols rsvp interface lt-2/0/10.2
user@R1# set LS2 protocols rsvp interface lt-2/0/10.10
user@R1# set LS2 protocols rsvp interface lo0.2
user@R1# set LS2 protocols mpls interface lt-2/0/10.2
user@R1# set LS2 protocols mpls interface lt-2/0/10.10
user@R1# set LS2 protocols mpls interface lo0.2
user@R1# set LS2 protocols ospf area 0.0.0.0 interface lt-2/0/10.2
user@R1# set LS2 protocols ospf area 0.0.0.0 interface lt-2/0/10.10
user@R1# set LS2 protocols ospf area 0.0.0.0 interface lo0.2
user@R1# set LS3 protocols rsvp interface lt-2/0/10.3
user@R1# set LS3 protocols rsvp interface lt-2/0/10.12
user@R1# set LS3 protocols rsvp interface lo0.3
user@R1# set LS3 protocols mpls interface lt-2/0/10.3
user@R1# set LS3 protocols mpls interface lt-2/0/10.12
user@R1# set LS3 protocols mpls interface lo0.3
user@R1# set LS3 protocols ospf area 0.0.0.0 interface lt-2/0/10.3
user@R1# set LS3 protocols ospf area 0.0.0.0 interface lt-2/0/10.12
user@R1# set LS3 protocols ospf area 0.0.0.0 interface lo0.3
user@R1# set LS4 protocols rsvp interface lt-2/0/10.4
user@R1# set LS4 protocols rsvp interface lo0.4
user@R1# set LS4 protocols mpls interface lt-2/0/10.4
user@R1# set LS4 protocols mpls interface lo0.4
user@R1# set LS4 protocols ospf area 0.0.0.0 interface ge-2/0/0.0
user@R1# set LS4 protocols ospf area 0.0.0.0 interface lt-2/0/10.4
user@R1# set LS4 protocols ospf area 0.0.0.0 interface lo0.4
user@R1# set LS5 protocols rsvp interface lt-2/0/10.5
user@R1# set LS5 protocols rsvp interface lo0.5
user@R1# set LS5 protocols mpls interface lt-2/0/10.5
user@R1# set LS5 protocols mpls interface lo0.5
```

```
user@R1# set LS5 protocols ospf area 0.0.0.0 interface ge-2/0/3.0
user@R1# set LS5 protocols ospf area 0.0.0.0 interface lt-2/0/10.5
user@R1# set LS5 protocols ospf area 0.0.0.0 interface lo0.5
user@R1# set LS6 protocols rsvp interface lt-2/0/10.6
user@R1# set LS6 protocols rsvp interface lt-2/0/10.11
user@R1# set LS6 protocols rsvp interface lo0.6
user@R1# set LS6 protocols mpls interface lt-2/0/10.6
user@R1# set LS6 protocols mpls interface lt-2/0/10.11
user@R1# set LS6 protocols mpls interface lo0.6
user@R1# set LS6 protocols ospf area 0.0.0.0 interface lt-2/0/10.6
user@R1# set LS6 protocols ospf area 0.0.0.0 interface lt-2/0/10.11
user@R1# set LS6 protocols ospf area 0.0.0.0 interface lo0.6
user@R1# set LS7 protocols rsvp interface lt-2/0/10.7
user@R1# set LS7 protocols rsvp interface lo0.7
user@R1# set LS7 protocols mpls interface lt-2/0/10.7
user@R1# set LS7 protocols mpls interface lo0.7
user@R1# set LS7 protocols ospf area 0.0.0.0 interface ge-2/0/1.0
user@R1# set LS7 protocols ospf area 0.0.0.0 interface lt-2/0/10.7
user@R1# set LS7 protocols ospf area 0.0.0.0 interface lo0.7
```

3. Enable traffic engineering for OSPF.

```
[edit logical-systems]
user@R1# set LS2 protocols ospf traffic-engineering
user@R1# set LS3 protocols ospf traffic-engineering
user@R1# set LS4 protocols ospf traffic-engineering
user@R1# set LS5 protocols ospf traffic-engineering
user@R1# set LS6 protocols ospf traffic-engineering
user@R1# set LS7 protocols ospf traffic-engineering
```

This causes the SPF algorithm to take into account the LSPs configured under MPLS.

4. Configure the router IDs.

```
[edit logical-systems]
user@R1# set LS2 routing-options router-id 100.20.20.20
user@R1# set LS3 routing-options router-id 100.30.30.30
user@R1# set LS4 routing-options router-id 100.40.40.40
user@R1# set LS5 routing-options router-id 100.50.50.50
user@R1# set LS6 routing-options router-id 100.60.60.60
user@R1# set LS7 routing-options router-id 100.70.70.70
```

5. If you are done configuring the device, commit the configuration.

```
[edit logical-systems]
user@R1# commit
```

Results

From configuration mode, confirm your configuration by entering the `show logical-systems` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R1# show logical-systems
LS1 {
    interfaces {
        ge-2/0/2 {
            unit 0 {
                description R1-to-CE1;
                family inet {
                    address 10.0.244.10/30;
                }
            }
        }
        lt-2/0/10 {
            unit 1 {
                description LS1-to-LS2;
                encapsulation ethernet;
                peer-unit 2;
                family inet {
                    address 2.2.2.1/24;
                }
                family mpls;
            }
            unit 8 {
                description LS1-to-LS6;
                encapsulation ethernet;
                peer-unit 6;
                family inet {
                    address 6.6.6.1/24;
                }
                family mpls;
            }
```

```
            unit 9 {
                description LS1-to-LS3;
                encapsulation ethernet;
                peer-unit 3;
                family inet {
                    address 3.3.3.1/24;
                }
                family mpls;
            }
        }
        lo0 {
            unit 1 {
                family inet {
                    address 100.10.10.10/32;
                }
            }
        }
    }
    protocols {
        rsvp {
            interface lt-2/0/10.1;
            interface lt-2/0/10.8;
            interface lt-2/0/10.9;
            interface lo0.1;
        }
        mpls {
            traffic-engineering bgp-igp;
            label-switched-path LS1-to-LS5 {
                to 100.50.50.50;
                p2mp p2mp1;
            }
            label-switched-path LS1-to-LS7 {
                to 100.70.70.70;
                p2mp p2mp1;
            }
            label-switched-path LS1-to-LS4 {
                to 100.40.40.40;
                p2mp p2mp1;
            }
            interface lt-2/0/10.1;
            interface lt-2/0/10.8;
            interface lt-2/0/10.9;
            interface lo0.1;
```

```
        }
        ospf {
            traffic-engineering;
            area 0.0.0.0 {
                interface ge-2/0/2.0;
                interface lt-2/0/10.1;
                interface lt-2/0/10.8;
                interface lt-2/0/10.9;
                interface lo0.1;
            }
        }
    }
    routing-options {
        static {
            route 5.5.5.0/24 {
                p2mp-lsp-next-hop p2mp1;
            }
            route 7.7.7.0/24 {
                p2mp-lsp-next-hop p2mp1;
            }
            route 4.4.4.0/24 {
                p2mp-lsp-next-hop p2mp1;
            }
        }
        router-id 100.10.10.10;
    }
}
LS2 {
    interfaces {
        lt-2/0/10 {
            unit 2 {
                description LS2-to-LS1;
                encapsulation ethernet;
                peer-unit 1;
                family inet {
                    address 172.16.2.2/24;
                }
                family mpls;
            }
            unit 10 {
                description LS2-to-LS5;
                encapsulation ethernet;
                peer-unit 5;
```

```
                family inet {
                    address 5.5.5.1/24;
                }
                family mpls;
            }
        }
        lo0 {
            unit 2 {
                family inet {
                    address 100.20.20.20/32;
                }
            }
        }
    }
    protocols {
        rsvp {
            interface lt-2/0/10.2;
            interface lt-2/0/10.10;
            interface lo0.2;
        }
        mpls {
            interface lt-2/0/10.2;
            interface lt-2/0/10.10;
            interface lo0.2;
        }
        ospf {
            traffic-engineering;
            area 0.0.0.0 {
                interface lt-2/0/10.2;
                interface lt-2/0/10.10;
                interface lo0.2;
            }
        }
    }
    routing-options {
        router-id 100.20.20.20;
    }
}
LS3 {
    interfaces {
        lt-2/0/10 {
            unit 3 {
                description LS3-to-LS1;
```

```
                encapsulation ethernet;
                peer-unit 9;
                family inet {
                    address 3.3.3.2/24;
                }
                family mpls;
            }
            unit 12 {
                description LS3-to-LS4;
                encapsulation ethernet;
                peer-unit 4;
                family inet {
                    address 4.4.4.1/24;
                }
                family mpls;
            }
        }
        lo0 {
            unit 3 {
                family inet {
                    address 100.30.30.30/32;
                }
            }
        }
    }
    protocols {
        rsvp {
            interface lt-2/0/10.3;
            interface lt-2/0/10.12;
            interface lo0.3;
        }
        mpls {
            interface lt-2/0/10.3;
            interface lt-2/0/10.12;
            interface lo0.3;
        }
        ospf {
            traffic-engineering;
            area 0.0.0.0 {
                interface lt-2/0/10.3;
                interface lt-2/0/10.12;
                interface lo0.3;
            }
```

```
        }
    }
    routing-options {
        router-id 100.30.30.30;
    }
}
LS4 {
    interfaces {
        ge-2/0/0 {
            unit 0 {
                description R1-to-CE4;
                family inet {
                    address 10.0.104.9/30;
                }
            }
        }
        lt-2/0/10 {
            unit 4 {
                description LS4-to-LS3;
                encapsulation ethernet;
                peer-unit 12;
                family inet {
                    address 4.4.4.2/24;
                }
                family mpls;
            }
        }
        lo0 {
            unit 4 {
                family inet {
                    address 100.40.40.40/32;
                }
            }
        }
    }
    protocols {
        rsvp {
            interface lt-2/0/10.4;
            interface lo0.4;
        }
        mpls {
            interface lt-2/0/10.4;
            interface lo0.4;
```

```
        }
        ospf {
            traffic-engineering;
            area 0.0.0.0 {
                interface ge-2/0/0.0;
                interface lt-2/0/10.4;
                interface lo0.4;
            }
        }
    }
    routing-options {
        router-id 100.40.40.40;
    }
}
LS5 {
    interfaces {
        ge-2/0/3 {
            unit 0 {
                description LS1-to-CE1;
                family inet {
                    address 10.0.224.10/30;
                }
            }
        }
        lt-2/0/10 {
            unit 5 {
                description LS5-to-LS2;
                encapsulation ethernet;
                peer-unit 10;
                family inet {
                    address 5.5.5.2/24;
                }
                family mpls;
            }
        }
        lo0 {
            unit 5 {
                family inet {
                    address 100.50.50.50/32;
                }
            }
        }
    }
```

```
    protocols {
        rsvp {
            interface lt-2/0/10.5;
            interface lo0.5;
        }
        mpls {
            interface lt-2/0/10.5;
            interface lo0.5;
        }
        ospf {
            traffic-engineering;
            area 0.0.0.0 {
                interface ge-2/0/3.0;
                interface lt-2/0/10.5;
                interface lo0.5;
            }
        }
    }
    routing-options {
        router-id 100.50.50.50;
    }
}
LS6 {
    interfaces {
        lt-2/0/10 {
            unit 6 {
                description LS6-to-LS1;
                encapsulation ethernet;
                peer-unit 8;
                family inet {
                    address 6.6.6.2/24;
                }
                family mpls;
            }
            unit 11 {
                description LS6-to-LS7;
                encapsulation ethernet;
                peer-unit 7;
                family inet {
                    address 7.7.7.1/24;
                }
                family mpls;
            }
```

```
        }
        lo0 {
            unit 6 {
                family inet {
                    address 100.60.60.60/32;
                }
            }
        }
    }
    protocols {
        rsvp {
            interface lt-2/0/10.6;
            interface lt-2/0/10.11;
            interface lo0.6;
        }
        mpls {
            interface lt-2/0/10.6;
            interface lt-2/0/10.11;
            interface lo0.6;
        }
        ospf {
            traffic-engineering;
            area 0.0.0.0 {
                interface lt-2/0/10.6;
                interface lt-2/0/10.11;
                interface lo0.6;
            }
        }
    }
    routing-options {
        router-id 100.60.60.60;
    }
}
LS7 {
    interfaces {
        ge-2/0/1 {
            unit 0 {
                description R1-to-CE3;
                family inet {
                    address 10.0.134.10/30;
                }
            }
        }
```

```
        lt-2/0/10 {
            unit 7 {
                description LS7-to-LS6;
                encapsulation ethernet;
                peer-unit 11;
                family inet {
                    address 7.7.7.2/24;
                }
                family mpls;
            }
        }
        lo0 {
            unit 7 {
                family inet {
                    address 100.70.70.70/32;
                }
            }
        }
    }
    protocols {
        rsvp {
            interface lt-2/0/10.7;
            interface lo0.7;
        }
        mpls {
            interface lt-2/0/10.7;
            interface lo0.7;
        }
        ospf {
            traffic-engineering;
            area 0.0.0.0 {
                interface ge-2/0/1.0;
                interface lt-2/0/10.7;
                interface lo0.7;
            }
        }
    }
    routing-options {
        router-id 100.70.70.70;
    }
}
```

Configuring Device CE1

Step-by-Step Procedure

To configure Device CE1:

1. Configure an interface to Logical System LS1.

```
[edit]
user@CE1# edit interfaces
[edit interfaces]
user@CE1# set ge-1/3/2 unit 0 family inet address 10.0.244.9/30
user@CE1# set ge-1/3/2 unit 0 description CE1-to-LS1
user@CE1# exit
```

2. Configure static routes from Device CE1 to the three other customer networks, with Logical System LS1 as the next hop.

```
[edit]
user@CE1# edit routing-options
[edit routing-options]
set static route 10.0.104.8/30 next-hop 10.0.244.10
set static route 10.0.134.8/30 next-hop 10.0.244.10
set static route 10.0.224.8/30 next-hop 10.0.244.10
user@CE1# exit
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@CE1# commit
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@CE1# show interfaces
interfaces {
```

```
    ge-1/3/2 {
        unit 0 {
            family inet {
                address 10.0.244.9/30;
                description CE1-to-LS1;
            }
        }
    }
}
```

```
user@CE1# show routing-options
routing-options {
    static {
        route 10.0.104.8/30 next-hop 10.0.244.10;
        route 10.0.134.8/30 next-hop 10.0.244.10;
        route 10.0.224.8/30 next-hop 10.0.244.10;
    }
}
```

Configuring Device CE2

Step-by-Step Procedure

To configure Device CE2:

1. Configure an interface to Logical System LS5.

```
[edit]
user@CE2# edit interfaces
[edit interfaces]
user@CE2# set ge-1/3/3 unit 0 family inet address 10.0.224.9/30
user@CE2# set ge-1/3/3 unit 0 description CE2-to-LS5
user@CE2# exit
```

2. Configure a static route from Device CE2 to CE1, with Logical System LS5 as the next hop.

```
[edit]
user@CE2# edit routing-options
[edit routing-options]
```

```
user@CE2# set static route 10.0.244.8/30 next-hop 10.0.224.10
user@CE2# exit
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@CE2# commit
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@CE2# show interfaces
interfaces {
    ge-1/3/3 {
        unit 0 {
            family inet {
                address 10.0.224.9/30;
                description CE2-to-LS5;
            }
        }
    }
}
```

```
user@CE2# show routing-options
routing-options {
    static {
        route 10.0.244.8/30 next-hop 10.0.224.10;
    }
}
```

Configuring Device CE3

Step-by-Step Procedure

To configure Device CE3:

1. Configure an interface to Logical System LS7.

```
[edit]
user@CE3# edit interfaces
[edit interfaces]
user@CE3# set ge-2/0/1 unit 0 family inet address 10.0.134.9/30
user@CE3# set ge-2/0/1 unit 0 description CE3-to-LS7
user@CE3# exit
```

2. Configure a static route from Device CE3 to CE1, with Logical System LS7 as the next hop.

```
[edit]
user@CE3# edit routing-options
[edit routing-options]
user@CE3# set static route 10.0.244.8/30 next-hop 10.0.134.10
user@CE3# exit
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@CE3# commit
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@CE3# show interfaces
interfaces {
    ge-2/0/1 {
        unit 0 {
            family inet {
                address 10.0.134.9/30;
                description CE3-to-LS7;
            }
        }
```

```
    }
}
```

```
user@CE3# show routing-options
routing-options {
    static {
        route 10.0.244.8/30 next-hop 10.0.134.10;
    }
}
```

Configuring Device CE4

Step-by-Step Procedure

To configure Device CE4:

1. Configure an interface to Logical System LS4.

```
[edit]
user@CE4# edit interfaces
[edit interfaces]
user@CE4# set ge-3/1/3 unit 0 family inet address 10.0.104.10/30
user@CE4# set ge-3/1/3 unit 0 description CE4-to-LS4
```

2. Configure a static route from Device CE4 to CE1, with Logical System LS4 as the next hop.

```
[edit]
user@CE4# edit routing-options
[edit routing-options]
user@CE4# set static route 10.0.244.8/30 next-hop 10.0.104.9
user@CE4# exit
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@CE4# commit
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@CE4# show interfaces
interfaces {
    ge-3/1/3 {
        unit 0 {
            family inet {
                address 10.0.104.10/30;
                description CE4-to-LS4;
            }
        }
    }
}
```

```
user@CE4# show routing-options
routing-options {
    static {
        route 10.0.244.8/30 next-hop 10.0.104.9;
    }
}
```

Verification

IN THIS SECTION

- Verifying Connectivity | 364
- Verifying the State of the Point-to-Multipoint LSP | 365
- Checking the Forwarding Table | 365

Confirm that the configuration is working properly.

Verifying Connectivity

Purpose

Make sure that the devices can ping each other.

Action

Run the `ping` command from CE1 to the interface on CE2 connecting to LS5.

```
user@CE1> ping 10.0.224.9
PING 10.0.224.9 (10.0.224.9): 56 data bytes
64 bytes from 10.0.224.9: icmp_seq=0 ttl=61 time=1.387 ms
64 bytes from 10.0.224.9: icmp_seq=1 ttl=61 time=1.394 ms
64 bytes from 10.0.224.9: icmp_seq=2 ttl=61 time=1.506 ms
^C
--- 10.0.224.9 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.387/1.429/1.506/0.055 ms
```

Run the `ping` command from CE1 to the interface on CE3 connecting to LS7.

```
user@CE1> ping 10.0.134.9
PING 10.0.134.9 (10.0.134.9): 56 data bytes
64 bytes from 10.0.134.9: icmp_seq=0 ttl=61 time=1.068 ms
64 bytes from 10.0.134.9: icmp_seq=1 ttl=61 time=1.062 ms
64 bytes from 10.0.134.9: icmp_seq=2 ttl=61 time=1.053 ms
^C
--- 10.0.134.9 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.053/1.061/1.068/0.006 ms
```

Run the `ping` command from CE1 to the interface on CE4 connecting to LS4.

```
user@CE1> ping 10.0.104.10
PING 10.0.104.10 (10.0.104.10): 56 data bytes
64 bytes from 10.0.104.10: icmp_seq=0 ttl=61 time=1.079 ms
64 bytes from 10.0.104.10: icmp_seq=1 ttl=61 time=1.048 ms
64 bytes from 10.0.104.10: icmp_seq=2 ttl=61 time=1.070 ms
^C
```

```
--- 10.0.104.10 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.048/1.066/1.079/0.013 ms
```

Verifying the State of the Point-to-Multipoint LSP

Purpose

Make sure that the ingress, transit, and egress LSRs are in the Up state.

> **NOTE**: For this example, the `show rsvp session` command displays the same output as the `show mpls lsp p2mp` command.

Action

Run the `show mpls lsp p2mp` command on all of the LSRs. Only the ingress LSR is shown here.

```
user@R1> set cli logical-system LS1
Logical system: LS1
user@R1:LS1> show mpls lsp p2mp
Ingress LSP: 1 sessions
P2MP name: p2mp1, P2MP branch count: 3
To              From            State Rt P     ActivePath       LSPname
100.40.40.40    100.10.10.10    Up     0 *                      LS1-LS4
100.70.70.70    100.10.10.10    Up     0 *                      LS1-LS7
100.50.50.50    100.10.10.10    Up     0 *                      LS1-LS5
Total 3 displayed, Up 3, Down 0
...
```

Checking the Forwarding Table

Purpose

Make sure that the routes are set up as expected by running the `show route forwarding-table` command. Only the routes to the remote customer networks are shown here.

Action

```
user@R1:LS1> show route forwarding-table
Routing table: default.inet
Internet:
Destination        Type RtRef Next hop           Type Index NhRef Netif
...
10.0.104.8/30      user     0 3.3.3.2            ucst  1006     6 lt-2/0/10.9
10.0.134.8/30      user     0 6.6.6.2            ucst  1010     6 lt-2/0/10.8
10.0.224.8/30      user     0 172.16.2.2            ucst  1008     6 lt-2/0/10.1
...
```

SEE ALSO

- Point-to-Multipoint LSPs Overview
- MPLS Applications User Guide

RELATED DOCUMENTATION

- Understanding Logical Systems for Routers and Switches | 2
- Setting Up Logical Systems | 17

4

CHAPTER

Monitoring Logical Systems

Monitoring Logical Systems for Routers and Switches | 368

Monitoring Logical Systems for Routers and Switches

IN THIS SECTION

- Example: Running Operational Mode Commands on Logical Systems | 368
- Logical System Logging Overview | 371
- Example: Viewing BGP Trace Files on Logical Systems | 371

Logical systems enable you to monitor the operational mode commands for router and switches. For more information, see the following topics:

Example: Running Operational Mode Commands on Logical Systems

IN THIS SECTION

- Requirements | 368
- Overview | 368
- Configuration | 369

This example shows how to set the CLI to a specified logical system view, run operational-mode commands for the logical system, and then return to the main router view.

Requirements

You must have the **view** privilege for the logical system.

Overview

For some operational-mode commands, you can include a **logical-system** option to narrow the output of the command or to limit the operation of the command to the specified logical system. For example, the

`show route` command has a **logical-system** option. To run this command on a logical system called LS3, you can use `show route logical-system LS3`. However, some commands, such as `show interfaces`, do not have a **logical-system** option. For commands like this, you need another approach.

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 the CLI is in logical system context mode and you enter an operational-mode command, the output of the command displays information related to the logical system only.

Configuration

IN THIS SECTION

- Procedure | 369

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To set the CLI to a specific logical system context:

1. From the main router, configure the logical system.

```
[edit]
user@host# set logical-systems LS3
```

2. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
user@host# exit
```

3. Set the CLI to view the logical system.

```
user@host> set cli logical-system LS3
Logical system: LS3
user@host:LS3>
```

4. Run an operational-mode command.

```
user@host:LS3> show interfaces terse
Interface               Admin Link Proto    Local                 Remote
lt-1/2/0
lt-1/2/0.3              up    up   inet     10.0.2.1/30
```

5. Enter configuration mode to edit the logical system configuration.

```
user@host:LS3> edit
Entering configuration mode
user@host:LS3#
```

6. Exit configuration mode to return to operational mode.

```
user@host:LS3# exit
Exiting configuration mode
```

7. Clear the logical system view to return to the main router view.

```
user@host:LS3> clear cli logical-system
Cleared default logical system
user@host>
```

8. To achieve the same effect when using a Junos XML protocol client application, include the <**set-logical-system**> tag.

```
        <rpc>
          <set-logical-system>
            <logical-system>LS1</logical-system>
```

```
        </set-logical-system>
      </rpc>
```

SEE ALSO

CLI Explorer

Logical System Logging Overview

The Junos OS generates system log messages (also called *syslog messages*) to record system events that occur on the device.

Each system log message identifies the Junos OS process that generated the message and briefly describes the operation or error that occurred. This reference provides more detailed information about each system log message, and when applicable, describes possible causes of the message and action you can take to correct error conditions.

When a logical system is created, the system log messages known as *syslog* is configured at the `[edit system]` hierarchy level. The syslog is written with the same files as the default logical system with the name of the logical system prepended to the syslog message.

Each logical system has a directory structure created in the **/var/logical-systems/*logical-system-name*** directory. If the traceoptions are configured under the logical system, then the system log messages are saved in the **/var/logical-systems/*logical-system-name*** directory.

NOTE: The syslog do not pass to **/var/logical-systems/*logical-system-name*** directory, but the traceoptions pass to the **/var/logical-systems/*logical-system-name*** directory.

Example: Viewing BGP Trace Files on Logical Systems

IN THIS SECTION

- Requirements | **372**
- Overview | **372**

Configuration | 373
Verification | 378

This example shows how to list and view files that are stored on a logical system.

Requirements

- You must have the **view** privilege for the logical system.
- Configure a network, such as the BGP network shown in *Example: Configuring Internal BGP Peering Sessions on Logical Systems*.

Overview

Logical systems have their individual directory structure created in the **/var/logical-systems/*logical-system-name*** directory. It contains the following subdirectories:

- **/config**—Contains the active configuration specific to the logical system.
- **/log**—Contains system log and tracing files specific to the logical system.

  To maintain backward compatibility for the log files with previous versions of Junos OS, a symbolic link (symlink) from the **/var/logs/*logical-system-name*** directory to the **/var/logical-systems/*logical-system-name*** directory is created when a logical system is configured.
- **/tmp**—Contains temporary files specific to the logical system.

The file system for each logical system enables logical system users to view trace logs and modify logical system files. Logical system administrators have full access to view and modify all files specific to the logical system.

Logical system users and administrators can save and load configuration files at the logical-system level using the **save** and **load** configuration mode commands. In addition, they can also issue the **show log**, **monitor**, and **file** operational mode commands at the logical-system level.

This example shows how to configure and view a BGP trace file on a logical system. The steps can be adapted to apply to trace operations for any Junos OS hierarchy level that supports trace operations.

TIP: To view a list of hierarchy levels that support tracing operations, enter the **help apropos traceoptions** command in configuration mode.

Configuration

IN THIS SECTION

- CLI Quick Configuration | 373
- Configuring Trace Operations | 373
- Viewing the Trace File | 374
- Deactivating and Reactivating Trace Logging | 377
- Results | 378

CLI Quick Configuration

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

```
set logical-systems A protocols bgp group internal-peers traceoptions file bgp-log
set logical-systems A protocols bgp group internal-peers traceoptions file size 10k
set logical-systems A protocols bgp group internal-peers traceoptions file files 2
set logical-systems A protocols bgp group internal-peers traceoptions flag update detail
```

Configuring Trace Operations

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the trace operations:

1. Configure trace operations on the logical system.

```
[edit logical-systems A protocols bgp group internal-peers]
user@host# set traceoptions file bgp-log
user@host# set traceoptions file size 10k
```

```
user@host# set traceoptions file files 2
user@host# set traceoptions flag update detail
```

2. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Viewing the Trace File

Step-by-Step Procedure

To view the trace file:

1. In operational mode on the main router, list the directories on the logical system.

```
user@host> file list /var/logical-systems/A
/var/logical-systems/A:
config/
log/
tmp/
```

2. In operational mode on the main router, list the log files on the logical system.

```
user@host> file list /var/logical-systems/A/log/
/var/logical-systems/A/log:
bgp-log
```

3. View the contents of the **bgp-log** file.

```
user@host> file show /var/logical-systems/A/log/bgp-log
Aug 10 17:12:01 trace_on: Tracing to "/var/log/A/bgp-log" started
Aug 10 17:14:22.826182 bgp_peer_mgmt_clear:5829: NOTIFICATION sent to 192.163.6.4 (Internal
AS 17): code 6 (Cease) subcode 4 (Administratively Reset), Reason: Management session
cleared BGP neighbor
Aug 10 17:14:22.826445 bgp_send: sending 21 bytes to 192.163.6.4 (Internal AS 17)
Aug 10 17:14:22.826499
Aug 10 17:14:22.826499 BGP SEND 192.168.6.5+64965 -> 192.163.6.4+179
```

```
Aug 10 17:14:22.826559 BGP SEND message type 3 (Notification) length 21
Aug 10 17:14:22.826598 BGP SEND Notification code 6 (Cease) subcode 4 (Administratively
Reset)
Aug 10 17:14:22.831756 bgp_peer_mgmt_clear:5829: NOTIFICATION sent to 192.168.40.4
(Internal AS 17): code 6 (Cease) subcode 4 (Administratively Reset), Reason: Management
session cleared BGP neighbor
Aug 10 17:14:22.831851 bgp_send: sending 21 bytes to 192.168.40.4 (Internal AS 17)
Aug 10 17:14:22.831901
Aug 10 17:14:22.831901 BGP SEND 192.168.6.5+53889 -> 192.168.40.4+179
Aug 10 17:14:22.831959 BGP SEND message type 3 (Notification) length 21
Aug 10 17:14:22.831999 BGP SEND Notification code 6 (Cease) subcode 4 (Administratively
Reset)
...
```

4. Filter the output of the log file.

```
user@host> file show /var/logical-systems/A/log/bgp-log | match "flags 0x40"
Aug 10 17:14:54.867460 BGP SEND flags 0x40 code Origin(1): IGP
Aug 10 17:14:54.867595 BGP SEND flags 0x40 code ASPath(2) length 0: <null>
Aug 10 17:14:54.867650 BGP SEND flags 0x40 code NextHop(3): 192.168.6.5
Aug 10 17:14:54.867692 BGP SEND flags 0x40 code LocalPref(5): 100
Aug 10 17:14:54.884529 BGP RECV flags 0x40 code Origin(1): IGP
Aug 10 17:14:54.884581 BGP RECV flags 0x40 code ASPath(2) length 0: <null>
Aug 10 17:14:54.884628 BGP RECV flags 0x40 code NextHop(3): 192.163.6.4
Aug 10 17:14:54.884667 BGP RECV flags 0x40 code LocalPref(5): 100
Aug 10 17:14:54.911377 BGP RECV flags 0x40 code Origin(1): IGP
Aug 10 17:14:54.911422 BGP RECV flags 0x40 code ASPath(2) length 0: <null>
Aug 10 17:14:54.911466 BGP RECV flags 0x40 code NextHop(3): 192.168.40.4
Aug 10 17:14:54.911507 BGP RECV flags 0x40 code LocalPref(5): 100
Aug 10 17:14:54.916008 BGP SEND flags 0x40 code Origin(1): IGP
Aug 10 17:14:54.916054 BGP SEND flags 0x40 code ASPath(2) length 0: <null>
Aug 10 17:14:54.916100 BGP SEND flags 0x40 code NextHop(3): 192.168.6.5
Aug 10 17:14:54.916143 BGP SEND flags 0x40 code LocalPref(5): 100
Aug 10 17:14:54.920304 BGP RECV flags 0x40 code Origin(1): IGP
Aug 10 17:14:54.920348 BGP RECV flags 0x40 code ASPath(2) length 0: <null>
Aug 10 17:14:54.920393 BGP RECV flags 0x40 code NextHop(3): 10.0.0.10
Aug 10 17:14:54.920434 BGP RECV flags 0x40 code LocalPref(5): 100
```

5. View the tracing operations in real time.

```
user@host> clear bgp neighbor logical-system A
Cleared 2 connections
```

> **CAUTION**: Clearing the BGP neighbor table is disruptive in a production environment.

6. Run the **monitor start** command with an optional **match** condition.

```
user@host> monitor start A/bgp-log | match 0.0.0.0/0
Aug 10 19:21:40.773467 BGP RECV         0.0.0.0/0
Aug 10 19:21:40.773685 bgp_rcv_nlri: 0.0.0.0/0
Aug 10 19:21:40.773778 bgp_rcv_nlri: 0.0.0.0/0 belongs to meshgroup
Aug 10 19:21:40.773832 bgp_rcv_nlri: 0.0.0.0/0 qualified bnp->ribact 0x0 l2afcb 0x0
```

7. Pause the **monitor** command by pressing Esc-Q.
   To unpause the output, press Esc-Q again.

8. Halt the **monitor** command by pressing Enter and typing **monitor stop**.

```
[Enter]
user@host> monitor stop
```

9. When you are finished troubleshooting, consider deactivating trace logging to avoid any unnecessary impact to system resources.

```
[edit protocols bgp group internal-peers]
user@host:A# deactivate traceoptions
user@host:A# commit
```

When configuration is deactivated, it appears in the configuration with the **inactive** tag.To reactivate trace operations, use the **activate** configuration-mode statement.

```
[edit protocols bgp group internal-peers]
user@host:A# show
```

```
type internal;
inactive: traceoptions {
    file bgp-log size 10k files 2;
    flag update detail;
    flag all;
}
local-address 192.168.6.5;
export send-direct;
neighbor 192.163.6.4;
neighbor 192.168.40.4;
```

10. To reactivate trace operations, use the **activate** configuration-mode statement.

```
[edit protocols bgp group internal-peers]
user@host:A# activate traceoptions
user@host:A# commit
```

Deactivating and Reactivating Trace Logging

Step-by-Step Procedure

To deactivate and reactivate the trace file:

1. When you are finished troubleshooting, consider deactivating trace logging to avoid an unnecessary impact to system resources.

```
[edit protocols bgp group internal-peers]
user@host:A# deactivate traceoptions
user@host:A# commit
```

When configuration is deactivated, the statement appears in the configuration with the **inactive** tag.

```
[edit protocols bgp group internal-peers]
user@host:A# show

type internal;
inactive: traceoptions {
    file bgp-log size 10k files 2;
    flag update detail;
    flag all;
```

```
}
local-address 192.168.6.5;
export send-direct;
neighbor 192.163.6.4;
neighbor 192.168.40.4;
```

2. To reactivate logging, use the **activate** configuration-mode statement.

```
[edit protocols bgp group internal-peers]
user@host:A# activate traceoptions
user@host:A# commit
```

Results

From configuration mode, confirm your configuration by entering the **show logical-systems A protocols bgp group internal-peers** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show logical-systems A protocols bgp group internal-peers
traceoptions {
    file bgp-log size 10k files 2;
    flag update detail;
}
```

Verification

IN THIS SECTION

- Verifying That the Trace Log File Is Operating | 379

Confirm that the configuration is working properly.

Verifying That the Trace Log File Is Operating

Purpose

Make sure that events are being written to the log file.

Action

```
user@host:A> show log bgp-log
Aug 12 11:20:57 trace_on: Tracing to "/var/log/A/bgp-log" started
```

RELATED DOCUMENTATION

Understanding Logical Systems for Routers and Switches | 2

User Access on Logical Systems | 14

5
CHAPTER

Configuration Statements

logical-system-mux | 381

logical-system-mux

IN THIS SECTION

- Syntax | 381
- Hierarchy Level | 382
- Description | 382
- Options | 382
- Required Privilege Level | 383

Syntax

```
logical-system-mux {
    traceoptions {
        file {
            <file name>;
            files;
            no-world-readable;
            size;
            world-readable;
            }

        flag {
            all;
            debug;
            general;
            normal;
            parse;
            policy;
            route;
            state;
            task;
            timer;
            }
```

```
        }
    }
```

Hierarchy Level

```
[edit routing-options]
```

Description

Display the logical system control daemon information.

Options

<filename>— Specify the filename, in which trace information should be written.

files—Specify the maximum number of trace files. Range being from 2 to 1000.

no-world-readable—Do not allow any user to read the log file.

size—Specify the maximum size of trace file. Range being from 10240 to 4294967295 bytes.

world-readable—Allow any user to read the log file.

all—Trace everything

debug—Trace debug-level lrmuxd activities.

general—Trace general events.

normal—Trace normal events.

parse—Trace configuration parsing.

policy—Trace policy parsing.

route—Trace routing information.

state—Trace state transitions.

task—Trace routing protocol task processing.

timer—Trace routing protocol timer processing.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.

routing-control—To add this statement to the configuration.

CHAPTER 6

Operational Commands

Operational-Mode Commands | 385

clear cli logical-system | 388

set cli logical-system | 390

show cli authorization | 391

show interfaces detail | 397

Operational-Mode Commands

IN THIS SECTION

- CLI Operational Mode Command Overview | 385

CLI Operational Mode Command Overview

IN THIS SECTION

- CLI Operational Mode Command Categories | 385
- Commonly Used Operational Mode Commands | 386

You (the network administrator) can control all network operations using the Junos OS CLI operational mode commands described in this topic.

CLI Operational Mode Command Categories

CLI operational mode commands fall into the following broad categories:

- Operational mode commands for monitoring and troubleshooting—The following commands perform functions related to information and statistics about the software and to test network connectivity.
  - `clear`—Clear statistics and protocol database information.
  - `file`—Perform file operations.
  - `mtrace`—Trace a multicast path from source to receiver.
  - `monitor`—Perform real-time debugging of various software components, including the routing protocols and interfaces.
  - `ping`—Determine the reachability of a remote network host.

  - `show`—Display the current configuration and information about interfaces, routing protocols, routing tables, routing policy filters, system alarms, and the chassis.
  - `test`—Test the configuration and application of policy filters and autonomous system (AS) path regular expressions.
  - `traceroute`—Trace the route to a remote network host.
- Commands for restarting software processes—The commands in the `restart` hierarchy restart the various system processes, including the routing protocol, interface, and SNMP.
- A command—`request`—Perform system-level operations, including stopping and rebooting the router or switch and loading operating system images.
- A command—`start`—Exit the CLI and start a UNIX shell.
- A command—`configure`—Enter configuration mode, which provides a series of commands that configure the system, including the routing protocols, interfaces, network management, and user access.

For more information about the CLI operational mode commands, see the CLI Explorer. Alternatively, you can enter `?` at the operational mode command prompt to view a list of available commands.

Commonly Used Operational Mode Commands

The following table lists some operational commands you may find useful for monitoring router or switch operation.

Table 2: Commonly Used Operational Mode Commands

Items to Check	Description	Command
Software version	Versions of software running on the router or switch	`show version`
Log files	Contents of the log files	`monitor`
	Log files and their contents and recent user logins	`show log`
Remote systems	Host reachability and network connectivity	`ping`
	The route to a network system	`traceroute`

Table 2: Commonly Used Operational Mode Commands *(Continued)*

Items to Check	Description	Command
Configuration	Current system configuration	`show configuration`
File manipulation	List of files and directories on the router or switch	`file list`
	Contents of a file	`file show`
Interface information	Detailed information about interfaces	`show interfaces`
Chassis	Chassis alarm status	`show chassis alarms`
	Information currently on craft display	`show chassis craft-interface`
	Router or switch environment information	`show chassis environment`
	Hardware inventory	`show chassis hardware`
Routing table information	Information about entries in the routing tables	`show route`
Forwarding table information	Information about data in the kernel's forwarding table	`show route forwarding-table`

RELATED DOCUMENTATION

- Understanding Logical Systems for Routers and Switches

clear cli logical-system

IN THIS SECTION

- Syntax | 388
- Description | 388
- Options | 388
- Required Privilege Level | 389
- Output Fields | 389
- Sample Output | 389
- Release Information | 389

Syntax

```
clear cli logical-system
```

Description

Clear the logical system view and return to a full router view. In a logical system view, the output of the command displays information related to the logical system only.

Options

This command has no options.

Required Privilege Level

clear

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear cli logical-system

```
user@host:lr1> clear cli logical-system

Cleared default logical system


user@host>
```

Release Information

Command introduced before Junos OS Release 7.4.

RELATED DOCUMENTATION

set cli logical-system | 390

Logical Systems User Guide for Routers and Switches

set cli logical-system

IN THIS SECTION

- Syntax | 390
- Description | 390
- Options | 390
- Required Privilege Level | 390
- Output Fields | 391
- Sample Output | 391
- Release Information | 391

Syntax

```
set cli logical-system logical-system
```

Description

Set the CLI to the specified logical system view.

Options

logical-system	logical system name.

Required Privilege Level

view

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

set cli logical-system

```
user@host> set cli logical-system log-router-A
logical system: log-router-A
user@host:log-router-A>
```

Release Information

Command introduced before Junos OS Release 7.4.

RELATED DOCUMENTATION

clear cli logical-system | 388

Logical Systems User Guide for Routers and Switches

show cli authorization

IN THIS SECTION

- Syntax | 392
- Description | 392
- Options | 392
- Required Privilege Level | 392

- Output Fields | **392**
- Sample Output | **395**
- Release Information | **396**

Syntax

```
show cli authorization
```

Description

Display the permissions for the current user.

Options

This command has no options.

Required Privilege Level

view

Output Fields

Table 3 on page 393 lists the output fields for the `show cli authorization` command. In the table, all possible permissions are displayed and output fields are listed in alphabetical order.

Table 3: show cli authorization Output Fields

Field Name	Field Description
access	Can view access configuration information.
access-control	Can modify access configuration.
admin	Can view user account information.
admin-control	Can modify user account information.
clear	Can clear learned network information.
configure	Can enter configuration mode.
control	Can modify any configuration.
edit	Can edit configuration files.
field	Reserved for field (debugging) support.
firewall	Can view firewall configuration information.
firewall-control	Can modify firewall configuration information.
floppy	Can read from and write to removable media.
flow-tap	Can view flow-tap configuration information.
flow-tap-control	Can configure flow-tap configuration information.
idp-profiler-operation	Can configure Profiler data.

Table 3: show cli authorization Output Fields *(Continued)*

Field Name	Field Description
`interface`	Can view interface configuration information.
`interface-control`	Can modify interface configuration information.
`maintenance`	Can perform system maintenance.
`network`	Can access the network by entering the `ping`, `ssh`, `telnet`, and `traceroute` commands.
`pgcp-session-mirroring`	Can view Packet Gateway Control Protocol session mirroring configuration.
`pgcp-session-mirroring-control`	Can modify Packet Gateway Control Protocol session mirroring configuration all-control.
`reset`	Can reset or restart interfaces and system processes.
`rollback`	Can roll back to previous configurations.
`routing`	Can view routing configuration information.
`routing-control`	Can modify routing configuration information.
`secret`	Can view passwords and authentication keys in the configuration.
`secret-control`	Can modify passwords and authentication keys in the configuration.
`security`	Can view security configuration information.
`security-control`	Can modify security configuration information.

Table 3: show cli authorization Output Fields *(Continued)*

Field Name	Field Description
shell	Can start a local shell.
snmp	Can view SNMP configuration information.
snmp-control	Can modify SNMP configuration information.
system	Can view system configuration information.
system-control	Can modify system configuration information.
trace	Can view trace file settings information.
trace-control	Can modify trace file settings information.
view	Can view current values and statistics.
view-configuration	Can view all configuration information (not including secrets).

Sample Output

show cli authorization

```
user@host> show cli authorization
Current user: 'remote' login: 'user' class ''
Permissions:
    admin       -- Can view user accounts
    admin-control-- Can modify user accounts
    clear       -- Can clear learned network information
    configure   -- Can enter configuration mode
    control     -- Can modify any configuration
```

```
    edit        -- Can edit full files
    field       -- Special for field (debug) support
    floppy      -- Can read and write from the floppy
    interface   -- Can view interface configuration
    interface-control-- Can modify interface configuration
    network     -- Can access the network
    reset       -- Can reset/restart interfaces and daemons
    routing     -- Can view routing configuration
    routing-control-- Can modify routing configuration
    shell       -- Can start a local shell
    snmp        -- Can view SNMP configuration
    snmp-control-- Can modify SNMP configuration
    system      -- Can view system configuration
    system-control-- Can modify system configuration
    trace       -- Can view trace file settings
    trace-control-- Can modify trace file settings
    view        -- Can view current values and statistics
    maintenance -- Can become the super-user
    firewall    -- Can view firewall configuration
    firewall-control-- Can modify firewall configuration
    secret      -- Can view secret configuration
    secret-control-- Can modify secret configuration
    rollback    -- Can rollback to previous configurations
    security    -- Can view security configuration
    security-control-- Can modify security configuration
    access      -- Can view access configuration
    access-control-- Can modify access configuration
    view-configuration-- Can view all configuration (not including secrets)
    flow-tap    -- Can view flow-tap configuration
    flow-tap-control-- Can configure flow-tap service
Individual command authorization:
    Allow regular expression: none
    Deny regular expression: none
    Allow configuration regular expression: none
    Deny configuration regular expression: none
```

Release Information

Command introduced before Junos OS Release 7.4.

show interfaces detail

IN THIS SECTION

- Syntax | 397
- Description | 397
- Options | 397
- Additional Information | 398
- Required Privilege Level | 398
- Output Fields | 398
- Sample Output | 398
- Release Information | 402

Syntax

```
show interfaces detail
```

Description

Display detailed information about all interfaces configured on the router.

Options

This command has no options.

Additional Information

In a logical system, this command displays information only about the logical interfaces and not about the physical interfaces.

Required Privilege Level

view

Output Fields

For more information, see the output fields table for the particular interface type in which you are interested. For information about destination class and source class statistics, see the “Destination Class Field” section and the “Source Class Field” section under *Common Output Fields Description*. For sample output for specific interfaces, see the other chapters in this manual.

Sample Output

show interfaces detail (SONET)

```
user@host> show interfaces so-1/1/0 detail
Physical interface: so-1/1/0, Enabled, Physical link is Up
  Interface index: 142, SNMP ifIndex: 47, Generation: 143
  Link-level type: PPP, MTU: 4474, Clocking: Internal, SONET mode, Speed: OC12,
  Loopback: None, FCS: 16, Payload scrambler: Enabled
  Device flags   : Present Running
  Interface flags: Point-To-Point SNMP-Traps Internal: 0x4000
  Link flags     : Keepalives
  Hold-times     : Up 0 ms, Down 0 ms
  Keepalive settings: Interval 10 seconds, Up-count 1, Down-count 3
  Keepalive statistics:
    Input : 1934 (last seen 17:35:39 ago)
    Output: 1927 (last sent 17:35:48 ago)
  LCP state: Down
  NCP state: inet: Down, inet6: Not-configured, iso: Down, mpls: Not-configured
```

```
  CHAP state: Closed
  CoS queues     : 4 supported, 4 maximum usable queues
  Last flapped   : 2006-04-19 15:22:33 PDT (05:25:55 ago)
  Statistics last cleared: 2006-04-18 03:58:02 PDT (1d 16:50 ago)
  Traffic statistics:
   Input  bytes  :              7910882                    0 bps
   Output bytes  :              5632131                    0 bps
   Input  packets:                89460                    0 pps
   Output packets:               116043                    0 pps
  SONET alarms   : None
  SONET defects  : None
  Logical interface so-1/1/0.0 (Index 69) (SNMP ifIndex 61) (Generation 138)
    Flags: Hardware-Down Point-To-Point SNMP-Traps 0x4000 Encapsulation: PPP
      Protocol inet, MTU: 4470, Generation: 156, Route table: 2
      Flags: Protocol-Down, SCU-out
                                            Packets                Bytes
      Source class                   (packet-per-second)    (bits-per-second)
                           gold1                    0                    0
                                 (                  0) (                 0)
                           gold2                    0                    0
                                 (                  0) (                 0)
                           gold3                    0                    0
                                 (                  0) (                 0)
      Addresses, Flags: Dest-route-down Is-Preferred Is-Primary
        Destination: 10.27.248/24, Local: 10.27.248.1, Broadcast: 10.27.248.255,
        Generation: 152
    Protocol iso, MTU: 4470, Generation: 157, Route table: 2
      Flags: Protocol-Down, Is-Primary
```

show interfaces detail (MX Series Routers)

```
user@host> show interfaces xe-0/0/0 detail
Physical interface: xe-0/0/0, Enabled, Physical link is Up
  Interface index: 145, SNMP ifIndex: 592, Generation: 148
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Speed: 10Gbps, BPDU Error: None,
  Loopback: None, Source filtering: Disabled, Flow control: Enabled
  Pad to minimum frame size: Enabled
  Device flags   : Present Running
  Interface flags: Hardware-Down SNMP-Traps Internal: 0x0
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
```

```
  Hold-times     : Up 0 ms, Down 0 ms
  Current address: 08:81:f4:82:a3:f0, Hardware address: 08:81:f4:82:a3:f0
  Last flapped   : 2013-10-26 03:20:40 test (1w6d 04:47 ago)
  Statistics last cleared: Never
  Traffic statistics:
   Input  bytes  :                   0                   0 bps
   Output bytes  :                   0                   0 bps
   Input  packets:                   0                   0 pps
   Output packets:                   0                   0 pps
   IPv6 transit statistics:
    Input  bytes  :                  0
    Output bytes  :                  0
    Input  packets:                  0
    Output packets:                  0
  Egress queues: 8 supported, 4 in use
  Queue counters:       Queued packets  Transmitted packets      Dropped packets
    0                                 0                   0                   0
    1                                 0                   0                   0
    2                                 0                   0                   0
    3                                 0                   0                   0
  Queue number:         Mapped forwarding classes
    0                   best-effort
    1                   expedited-forwarding
    2                   assured-forwarding
    3                   network-control
  Active alarms  : LINK
  Active defects : LINK
  PCS statistics                      Seconds
    Bit errors                            79
    Errored blocks                        79
  Interface transmit statistics: Disabled
```

show interfaces detail (Junos OS Evolved)

```
user@host> show interfaces et-0/0/2:2 detail
Physical interface: et-0/0/2:2, Enabled, Physical link is Up
  Interface index: 1164, SNMP ifIndex: 592, Generation: 275
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Speed: 10Gbps,
  BPDU Error: None, MAC-REWRITE Error: None, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Enabled
  Device flags   : Present
```

```
  Interface flags: SNMP-Traps
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Hold-times     : Up 0 ms, Down 0 ms
  Damping        : half-life: 0 sec, max-suppress: 0 sec, reuse: 0, suppress: 0, state:
unsuppressed
  Current address: 28:8a:1c:68:84:ca, Hardware address: 28:8a:1c:68:84:ca
  Last flapped   : Never
  Statistics last cleared: Never
  Traffic statistics:
   Input  bytes  :                   0                   0 bps
   Output bytes  :                   0                   0 bps
   Input  packets:                   0                   0 pps
   Output packets:                   0                   0 pps
   IPv6 transit statistics:
    Input  bytes  :                  0
    Output bytes  :                  0
    Input  packets:                  0
    Output packets:                  0
  Input errors: 3, Output errors: 0
  Input errors:
    Errors: 0, Drops: 0, Framing errors: 3, Runts: 0, Policed discards: 0,
    L3 incompletes: 0, L2 channel errors: 0, L2 mismatch timeouts: 0,
    FIFO errors: 0, Resource errors: 0
  Output errors:
    Carrier transitions: 0, Errors: 0, Drops: 0, Collisions: 0, Aged packets: 0,
    FIFO errors: 0, HS link CRC errors: 0, MTU errors: 0, Resource errors: 0
  Active alarms  : None
  Active defects : None
  Egress queues: 8 supported, 4 in use
  Queue counters:       Queued packets  Transmitted packets      Dropped packets
    0                              0                   0                   0
    1                              0                   0                   0
    2                              0                   0                   0
    3                              0                   0                   0
  Queue number:         Mapped forwarding classes
    0                   best-effort
    1                   expedited-forwarding
    2                   assured-forwarding
    3                   network-control
  PCS statistics                    Seconds
```

```
    Bit errors                          0
    Errored blocks                      0
```

Release Information

Command introduced before Junos OS Release 7.4.