JUNOS® Software

Feature Guide

Release 9.5

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

Part Number: 530-029323-01, Revision 1

JUNOS® Software Feature Guide,
Release 9.5

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

Revision History
13 April 2009—530-029323-01 Revision 1

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Abbreviated Table of Contents

Table of Contents

Part 3 Services Interfaces

Chapter 10 Flow Monitoring 281

List of Figures

List of Tables

About This Guide

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

JUNOS Documentation and Release Notes

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

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

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

Table 1 on page xxix lists additional books on Juniper Networks solutions that you can order through your bookstore. A complete list of such books is available at http://www.juniper.net/books.

Table 1: Additional Books Available Through http://www.juniper.net/books

Book	Description
Interdomain Multicast Routing	Provides background and in-depth analysis of multicast routing using Protocol Independent Multicast sparse mode (PIM SM) and Multicast Source Discovery Protocol (MSDP); details any-source and source-specific multicast delivery models; explores multiprotocol BGP (MBGP) and multicast IS-IS; explains Internet Gateway Management Protocol (IGMP) versions 1, 2, and 3; lists packet formats for IGMP, PIM, and MSDP; and provides a complete glossary of multicast terms.

Table 1: Additional Books Available Through http://www.juniper.net/books *(continued)*

Book	Description
JUNOS Cookbook	Provides detailed examples of common JUNOS software configuration tasks, such as basic router configuration and file management, security and access control, logging, routing policy, firewalls, routing protocols, MPLS, and VPNs.
MPLS-Enabled Applications	Provides an overview of Multiprotocol Label Switching (MPLS) applications (such as Layer 3 virtual private networks [VPNs], Layer 2 VPNs, virtual private LAN service [VPLS], and pseudowires), explains how to apply MPLS, examines the scaling requirements of equipment at different points in the network, and covers the following topics: point-to-multipoint label switched paths (LSPs), DiffServ-aware traffic engineering, class of service, interdomain traffic engineering, path computation, route target filtering, multicast support for Layer 3 VPNs, and management and troubleshooting of MPLS networks.
OSPF and IS-IS: Choosing an IGP for Large-Scale Networks	Explores the full range of characteristics and capabilities for the two major link-state routing protocols: Open Shortest Path First (OSPF) and IS-IS. Explains architecture, packet types, and addressing; demonstrates how to improve scalability; shows how to design large-scale networks for maximum security and reliability; details protocol extensions for MPLS-based traffic engineering, IPv6, and multitopology routing; and covers troubleshooting for OSPF and IS-IS networks.
Routing Policy and Protocols for Multivendor IP Networks	Provides a brief history of the Internet, explains IP addressing and routing (Routing Information Protocol [RIP], OSPF, IS-IS, and Border Gateway Protocol [BGP]), explores ISP peering and routing policies, and displays configurations for both Juniper Networks and other vendors' routers.
The Complete IS-IS Protocol	Provides the insight and practical solutions necessary to understand the IS-IS protocol and how it works by using a multivendor, real-world approach.

Objectives

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

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

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

- Overview—Introduces the feature topic to the user.
- System Requirements—Lists the equipment and software needed to implement a feature.
- Terms and Acronyms—Explains the terminology used with each feature example.
- Feature Implementation—Discusses the background needed to understand how to implement the feature.

- Configuring the feature—Shows the steps that are needed to configure a feature.
- Example—Gives an actual configuration example, along with verification commands and output.
- For More Information—Provides additional resources and information for further understanding of the feature.

NOTE: This guide documents Release 9.5 of the JUNOS software. For additional information about the JUNOS software—either corrections to or information that might have been omitted from this guide—see the software release notes at http://www.juniper.net/.

Audience

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

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

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

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

Supported Routing Platforms

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

- J-series
- M-series
- MX-series
- T-series

Using the Indexes

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

Using the Examples in This Manual

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

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

Merging a Full Example

To merge a full example, follow these steps:

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

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

```
system {
    scripts {
        commit {
            file ex-script.xsl;
        }
    }
}
interfaces {
    fxp0 {
        disable;
```

```
        unit 0 {
            family inet {
                address 10.0.0.1/24;
            }
        }
    }
}
```

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

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

Merging a Snippet

To merge a snippet, follow these steps:

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

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

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

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

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

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

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

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

Documentation Conventions

Table 2 on page xxxiv defines notice icons used in this guide.

Table 2: Notice Icons

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

Table 3 on page xxxiv defines the text and syntax conventions used in this guide.

Table 3: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the **configure** command: user@host> **configure**
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host> show chassis alarms` `No alarms currently active`
Italic text like this	■ Introduces important new terms. ■ Identifies book names. ■ Identifies RFC and Internet draft titles.	■ A policy *term* is a named structure that defines match conditions and actions. ■ *JUNOS System Basics Configuration Guide* ■ RFC 1997, *BGP Communities Attribute*
Italic text like this	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*
Plain text like this	Represents names of configuration statements, commands, files, and directories; IP addresses; configuration hierarchy levels; or labels on routing platform components.	■ To configure a stub area, include the **stub** statement at the **[edit protocols ospf area area-id]** hierarchy level. ■ The console port is labeled **CONSOLE**.
< > (angle brackets)	Enclose optional keywords or variables.	**stub <default-metric** *metric***>;**

Table 3: Text and Syntax Conventions *(continued)*

Convention	Description	Examples
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	broadcast | multicast (*string1* | *string2* | *string3*)
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	rsvp { # Required for dynamic MPLS only
[ ] (square brackets)	Enclose a variable for which you can substitute one or more values.	community name members [*community-ids*]

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

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

Documentation Feedback

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

- Document name
- Document part number
- Page number
- Software release version (not required for *Network Operations Guides [NOGs])*

Requesting Technical Support

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

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

Self-Help Online Tools and Resources

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

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

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

Opening a Case with JTAC

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

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

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

Part 1
MPLS Applications

Chapter 1

GMPLS

This feature guide chapter covers these topics:

Overview

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

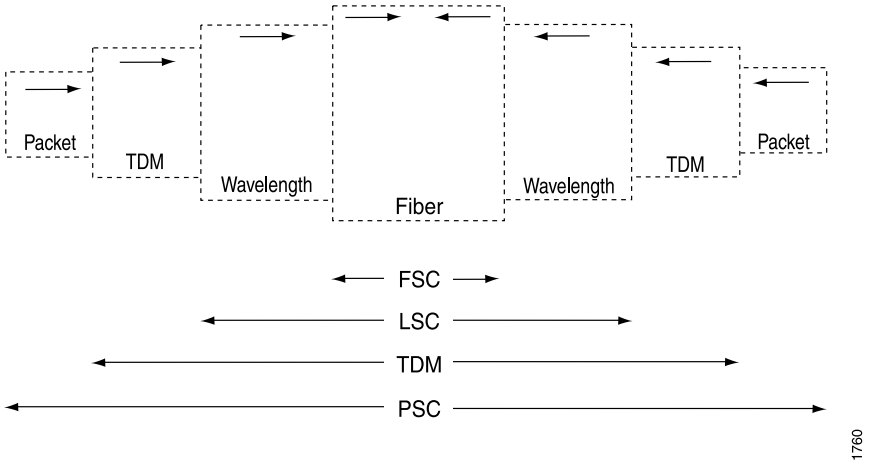
This document assumes you have a general understanding of MPLS, label switching concepts, and GMPLS Phase 1. For more information about MPLS, see the *JUNOS MPLS Applications Configuration Guide.* For more information about GMPLS Phase 1, see the *JUNOS 5.5 Feature Guide* at: http://www.juniper.net/techpubs/software/junos/junos55/feature-guide55/feature-guide-55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

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

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

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

Figure 1: GMPLS LSP Hierarchy



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

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

There are two service models for GMPLS. Each model determines how much visibility a client node, such as a router, has into the optical core or transport network. The first model is a user-to-network interface (UNI), which is often referred to as the overlay model. The second is known as the peer model. Juniper Networks supports both models.

To enable multilayer LSPs, GMPLS uses the following mechanisms:

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

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

System Requirements

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

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

Terms and Acronyms

C

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

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

F

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

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

G

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

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

L

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

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

P

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

T

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

traffic engineering link (TE link) A logical connection between GMPLS-enabled devices. TE links can have addresses or IDs and are associated with certain resources or interfaces. They also have certain inherent attributes, such as encoding-type, switching capability, and bandwidth. Each TE link represents a forwarding adjacency between a pair of devices.

GMPLS Phase 2 Implementation

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

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

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

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

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

The current JUNOS software release supports the following GMPLS functionality:

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

The following functionality is *not* supported in this release:

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

NOTE: There is not necessarily a one-to-one correspondence between a physical interface and a GMPLS interface. If a GMPLS connection uses a nonchannelized physical connector, the GMPLS label can use the physical port ID. However, the label for channelized interfaces often is based on a channel or timeslot. Consequently, it is best not to refer to GMPLS labels as "interfaces." To avoid confusion, refer to them as TE links and refer to the physical interfaces as resources.

GMPLS Operation

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

1. LMP notifies RSVP and OSPF of the control peer, the control adjacency, and resources for the TE link.
2. GMPLS extracts the LSP attributes from the configuration and requests RSVP to signal one or more specific paths, specified by the TE link addresses.
3. RSVP determines the local TE link, corresponding control adjacency and active control channel, and transmission parameters (such as IP destination). It requests that LMP allocate a resource from the TE link with the specified attributes. If LMP successfully finds a resource matching the attributes, label allocation succeeds. RSVP sends a PathMsg hop-by-hop until it reaches the target router.
4. The target router, on receiving the RSVP PathMsg, requests that LMP allocate a resource based on the signaled parameters. If label allocation succeeds, it sends back a ResvMsg.
5. If the signaling is successful, an optical path is provisioned.

Configuring GMPLS

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

Configuring Link Management Protocol Traffic Engineering Links

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

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

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

Configuring Link Management Protocol Peers

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

You can configure one or more control channels for a peer. The control channels must have point-to-point connectivity with the peer (for example, you can use a point-to-point link or a tunnel). You can also configure the generic routing encapsulation (GRE) tunnel interface of the Routing Engine as a control channel. To configure a static control channel, include the control-channel statement at the [edit protocols link-management peer *peer-name*] hierarchy level. To configure a control channel that uses control channel management and link property correlation, see "Option: Configuring an LMP Control Channel" on page 16. Without a control channel, the configuration will fail to commit.

```
[edit]
protocols {
   link-management {
      peer peer-name { # Configure the name of your network peer.
         address ip-address; # Include the router ID of the peer.
         control-channel interface; # Specify the interface for the control channel.
         te-link te-link-name; # Assign a TE link to this peer.
      }
   }
}
```

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

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

Configuring Peer Interfaces in OSPF and RSVP

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

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

```
[edit]
protocols {
   rsvp {
      peer-interface peer-name { # Configure the name of your LMP peer.
      }
      ospf {
         area  area-number  {
            peer-interface peer-name { # Configure the name of your LMP peer.
            }
         }
      }
   }
}
```

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

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

Establishing GMPLS LSP Path Information

When you configure LSP paths for GMPLS, you must use the TE link remote address as your next-hop address. When CSPF is supported, you can use any path option you wish. However, when CSPF is disabled with the no-cspf statement at the [edit protocols mpls label-switched-path *lsp-name*] hierarchy level, you must use strict paths.

```
[edit]
protocols {
    mpls {
        path path-name {
            next-hop-address (strict | loose);
        }
    }
}
```

Defining GMPLS Label-Switched Paths

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

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

```
[edit]
protocols {
    mpls {
        label-switched-path lsp-name {
            from ip-address;
            to ip-address;
            primary path-name;
            secondary path-name;
```

```
            no-cspf; # This statement to disable CSPF is optional.
            lsp-attributes { # Attributes determine the selection of an LSP.
               gpid type;  # Payload type, such as Ethernet or PPP.
               signal-bandwidth type; # Bandwidth encoding, such as DS3 or STM1.
               switching-type type; # Switching method, such as psc-1 or lambda.
            }
         }
      }
   }
```

NOTE: Because MPLS and GMPLS use the same configuration hierarchy for LSPs, it is helpful to know which LSP attributes control LSP functionality. Standard MPLS packet-switched LSPs are unidirectional, while GMPLS nonpacket LSPs are bidirectional.

If you use the default packet switching type of psc-1, your LSP becomes unidirectional. To enable a GMPLS bidirectional LSP, you must select a nonpacket switching type option, such as lambda or fiber, at the [edit mpls label-switched-path *lsp-name* lsp-attributes] hierarchy level.

Displaying Local Identifiers and Configuring Remote Identifiers

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

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

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

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

```
[edit]
protocols {
   link-management {
      te-link te-link-name {
         disable; # Disable the entire TE link.
         remote-id id-number; # TE link ID number of the peer device.
```

```
            interface interface-name { # Name of the interface used for data transfer.
                disable; # Disable an interface in the TE link.
                remote-id id-number; # ID number of the remote device.
            }
        }
    }
}
```

Option: Tearing Down GMPLS LSPs Gracefully

You can tear down a nonpacket GMPLS LSP in a two-step process that gracefully withdraws the RSVP session used by the LSP. For all neighbors that support graceful teardown, a request for the teardown is sent by the routing platform to the destination endpoint for the LSP and all RSVP neighbors in the path. The request is included within the ADMIN_STATUS field of the RSVP packet. When neighbors receive the request, they prepare for the RSVP session to be withdrawn. A second message is sent by the routing platform to complete the teardown of the RSVP session. If a neighbor does not support graceful teardown, the request is handled as a standard session teardown rather than a graceful one.

To perform a graceful teardown of a GMPLS LSP RSVP session, issue the clear rsvp session gracefully command. Optionally, you can specify the source and destination address of the RSVP session, the LSP identifier of the RSVP sender, and the tunnel identifier of the RSVP session. To use these qualifiers, include the connection-source, connection-destination, lsp-id, and tunnel-id options when you issue the clear rsvp session gracefully command.

You can also configure the amount of time that the routing platform waits for neighbors to receive the graceful teardown request before initiating the actual teardown. To configure, include the graceful-deletion-timeout statement at the [edit protocols rsvp] hierarchy level. The default graceful deletion timeout value is 30 seconds, with a minimum value of 1 second and a maximum value of 300 seconds. To view the current value configured for graceful deletion timeout, issue the show rsvp version operational command.

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

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

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

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

Option: Selecting the Peer Model for GMPLS

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

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

Option: Selecting the Overlay Model for GMPLS

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

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

Option: GMPLS Graceful Restart

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

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

To configure the maximum amount of time the routing platform retains information for restarting neighbors, include the maximum-helper-recovery-time statement at the [edit protocols rsvp graceful-restart] hierarchy level. The default helper recovery time is 180 seconds, with a minimum value of 1 second and a maximum value of 3600 seconds. To view the current value configured for the helper recovery time, issue the show rsvp version operational mode command.

To configure the maximum amount of time the routing platform waits until a neighbor is declared dead, include the maximum-helper-restart-time statement at the [edit protocols rsvp graceful-restart] hierarchy level. The default helper restart time is 20

seconds, with a minimum value of 1 second and a maximum value of 1800 seconds. To view the current value configured for the helper restart time, issue the show rsvp version operational command.

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

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

Option: Configuring an LMP Control Channel

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

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

To configure an LMP control channel, include the lmp-control-channel statement at the [edit protocols link-management peer *peer-name*] hierarchy level and specify the physical IP address of the peer as the remote address. You can also specify a hello interval, a hello dead interval, a retransmission interval, and a retry limit. Default values for these statements are shown in Table 4 on page 16.

Table 4: Default Values for LMP Protocol Fields

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

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

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

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

For a full example of an LMP protocol control channel configuration, see "Example: LMP Control Channel Configuration" on page 30.

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

Option: Configuring GMPLS Support for Unnumbered Links

The JUNOS software supports RSVP traffic engineering over unnumbered interfaces. With this feature, you do not have to configure IP addresses for each interface participating in the RSVP-signaled network. Traffic engineering information about unnumbered links is carried in the IGP traffic engineering extensions for OSPF and IS-IS, as described in RFC 4203, *OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)*. Unnumbered links can also be specified in

the MPLS traffic engineering signaling, as described in RFC 3477, *Signaling Unnumbered Links in Resource ReSerVation Protocol - Traffic Engineering (RSVP-TE)*.

To configure RSVP for unnumbered interfaces, you must configure the router with a router ID using the router-id statement specified at the [edit routing-options] hierarchy level. The router ID must be routable (you can typically use the loopback address). The RSVP control messages for the unnumbered links are sent using the router ID address (rather than a randomly selected address). To configure link protection and fast reroute on a router with unnumbered interfaces enabled, you must configure at least two addresses. In addition to the router ID, Juniper Networks recommends that you configure a secondary interface on the loopback:

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

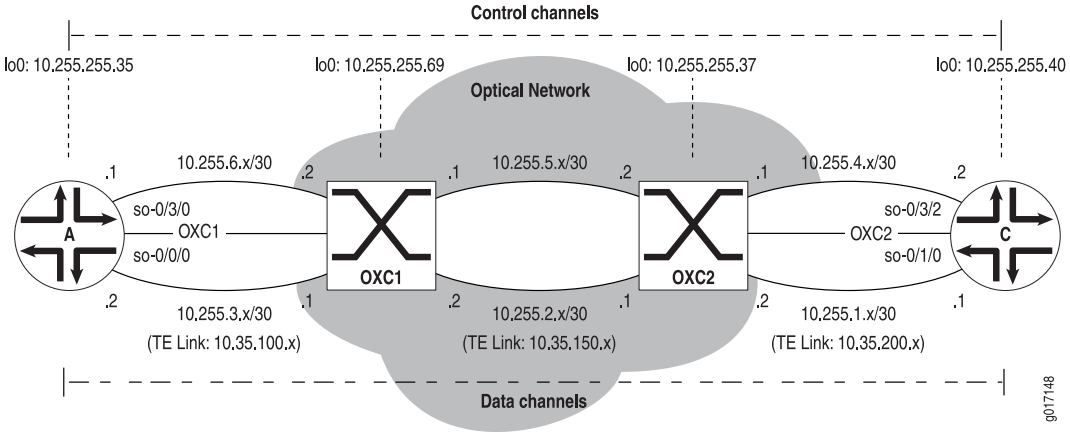
GMPLS Configuration Examples

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

- Example: GMPLS Configuration on page 18
- Example: Configuring TE Link and Interface Identifiers on page 29
- Example: LMP Control Channel Configuration on page 30

Example: GMPLS Configuration

Figure 2: GMPLS Topology Diagram



In Figure 2 on page 18, a control channel is established between Router A and OXC1, OXC1 and OXC2, and OXC2 and Router C. A data channel is enabled on a second connection between each pair of devices. The optical network cloud can contain OXCs, ADMs, or other lower-layer devices. In this example, OXC1 and OXC2 are in

the direct data path between Routers A and C and the two OXCs have point-to-point connectivity with each other and the directly connected peer routers.

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

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

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

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

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

Router A

```
[edit]
interfaces {
    so-0/0/0 {
        description “Data channel to OXC1”;
        encapsulation ppp;
        unit 0 {
            family inet {
                address 10.255.3.2/30 {
                    destination 10.255.3.1;
                }
            }
            family mpls;
        }
    }
    so-0/3/0 {
        description “Control channel to OXC1”;
        encapsulation ppp;
        unit 0 {
            family inet {
                address 10.255.6.1/30 {
```

```
            destination 10.255.6.2;
          }
        }
        family mpls;
      }
    }
    lo0 {
      unit 0 {
        family inet {
          address 10.255.255.35/32;
        }
      }
    }
}
protocols
rsvp {
    interface all;
    interface so-0/3/0.0 {
      disable;
    }
    peer-interface oxc1;
}
mpls {
    label-switched-path gmpls-lsp1 {
      to 10.255.255.40;
      lsp-attributes {
        signal-bandwidth stm-1;
        switching-type fiber;
        gpid ppp;
      }
      primary path-lsp1;
    }
    path path-lsp1 {
      10.35.100.1 strict; # This example does not disable CSPF,
      10.35.150.1 strict; # so this step is optional.
      10.35.200.1 strict;
    }
    interface all;
}
ospf {
    area 0.0.0.0 {
      interface lo0.0;
      interface fxp0.0 {
        disable;
      }
      peer-interface oxc1;
    }
}
link-management {
    te-link te-oxc1 {
      local-address 10.35.100.2;
      remote-address 10.35.100.1;
      remote-id 8256;
      interface t3-3/3/0:0 {
        local-address 10.35.100.2;
        remote-address 10.35.100.1;
```

```
            remote-id 65536;
        }
    }
    peer oxc1 {
        address 10.255.255.69;
        control-channel so-0/3/0.0;
        te-link te-oxc1;
    }
}
```

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

In the optical network between your OXCs, configure a TE link and a control channel between OXC1 and OXC2. Refer to the OXC vendor's instructions to configure this link. For the example shown in Figure 2 on page 18, you can assume a TE link with an address space of 10.255.150.x/30 has been enabled over a physical network with IP addresses 10.255.2.x/30. Also, a control channel has been created over the 10.255.4.x/30 link.

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

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

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

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

Router C

```
[edit]
interfaces {
    so-0/3/2 {
        description "Control channel to OXC2";
        unit 0 {
            family inet {
                address 10.255.4.2/30 {
                    destination 10.255.4.1;
                }
            }
```

```
            family mpls;
        }
    }
    so-0/1/0 {
        description “Data channel to OXC2”;
        encapsulation ppp;
        unit 0 {
            family inet {
                address 10.255.1.1/30 {
                    destination 10.255.1.2;
                }
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.255.40/32;
            }
        }
    }
}
protocols
rsvp {
    interface all;
    interface so-0/3/2.0 {
        disabled;
    }
    peer-interface oxc2;
}
mpls {
    interface all;
}
ospf {
    area 0.0.0.0 {
        interface fxp0.0 {
            disable;
        }
        interface lo0.0;
        peer-interface oxc2;
    }
}
link-management {
    te-link te-oxc2 {
        local-address 10.35.200.1;
        remote-address 10.35.200.2;
        remote id 41060;
        interface so-0/1/0 {
            local-address 10.35.200.1;
            remote-address 10.35.200.2;
            remote-id 22278;
        }
    }
    peer oxc2 {
        address 10.255.255.37;
```

```
        control-channel so-0/3/2.0;
        te-link te-oxc2;
    }
}
```

Verifying Your Work

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

- show link-management (te-link | peer)
- show link-management routing (te-link | peer)
- show mpls lsp (bidirectional | unidirectional)
- show mpls lsp (detail | extensive)
- show ospf interface
- show ospf neighbor
- show rsvp interface link-management
- show rsvp session (bidirectional | unidirectional)
- show rsvp session te-link
- show rsvp session detail
- show rsvp neighbor detail
- show ted database extensive
- traceroute (using the lsp flag with RSVP protocol–level trace options)

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

- Router A Status on page 23
- Router C Status on page 28

Router A Status

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

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

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

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

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

To further verify proper operation, use the following commands:

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

user@RouterA> show link-management routing te-link
 TE link name: te-oxc1, State: Up
  Local identifier: 8255, Remote identifier: 8256, Local address: 10.35.100.2,
Remote address: 10.35.100.1, Encoding: SDH/SONET,
  Minimum bandwidth: 155.52Mbps, Maximum bandwidth: 155.52Mbps, Total bandwidth:
 155.52Mbps, Available bandwidth: 0bps

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

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

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

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

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

10.255.255.40
  From: 10.255.255.35, State: Up, ActiveRoute: 0, LSPname: gmpls-lsp1
  Bidirectional
```

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

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

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

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

```
user@RouterA> show ospf interface
Interface           State     Area            DR ID           BDR ID       Nbrs
lo0.0               DR       0.0.0.0         10.255.255.35   0.0.0.0         0
oxc1                PtToPt   0.0.0.0         0.0.0.0         0.0.0.0         1
```

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

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

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

10.255.255.40
  From: 10.255.255.35, LSPstate: Up, ActiveRoute: 0
  LSPname: gmpls-lsp1, LSPpath: Primary
  Bidirectional, Upstream label in: 27676, Upstream label out: -
  Suggested label received: -, Suggested label sent: 27676
  Recovery label received: -, Recovery label sent: 60444
  Resv style: 1 FF, Label in: -, Label out: 60444
  Time left:    -,  Since: Thu Nov  7 15:47:11 2002
  Tspec: rate 0bps size 0bps peak 1.544Mbps m 20 M 1500
  Port number: sender 1 receiver 17 protocol 0
  PATH rcvfrom: localclient
```

```
  PATH sentto: 10.255.255.40 (oxc1) 157 pkts
  RESV rcvfrom: 10.255.255.40  (oxc1) 71 pkts
  Explct route: 10.35.100.1 10.35.150.1 10.35.200.1
  Record route: <self>  10.35.100.1 10.35.150.1 10.35.200.1
Total 1 displayed, Up 1, Down 0

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

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

user@RouterA>  show rsvp session bidirectional
Ingress RSVP: 1 sessions
To              From            State Rt Style Labelin Labelout LSPname
10.255.255.40   10.255.255.35   Up     0  1 FF       -    60444 gmpls-lsp1 Bidir
Total 1 displayed, Up 1, Down 0

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

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

user@RouterA>  show rsvp session te-link te-oxc1
Ingress RSVP: 1 sessions
To              From            State Rt Style Labelin Labelout LSPname
10.255.255.40   10.255.255.35   Up     0  1 FF       -    60444 gmpls-lsp1 Bidir
Total 1 displayed, Up 1, Down 0

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

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

user@RouterA>  show ted database extensive
TED database: 0 ISIS nodes 4 INET nodes
NodeID: 10.255.255.35
  Type: Rtr, Age: 2178 secs, LinkIn: 4, LinkOut: 5
  Protocol: OSPF(0.0.0.0)
    To: 10.255.255.69, Local: 10.35.100.2, Remote: 10.35.100.1
      Metric: 1
      Static BW: 155.52Mbps
      Reservable BW: 155.52Mbps
      Available BW [priority] bps:
       [0] 0bps         [1] 0bps         [2] 0bps         [3] 0bps
       [4] 0bps         [5] 0bps         [6] 0bps         [7] 0bps
      Interface Switching Capability Descriptor(1):
        Switching type: Fiber
        Encoding type: SDH/SONET
        Maximum LSP BW [priority] bps:
         [0] 155.52Mbps   [1] 155.52Mbps   [2] 155.52Mbps   [3] 155.52Mbps
         [4] 155.52Mbps   [5] 155.52Mbps   [6] 155.52Mbps   [7] 155.52Mbps
        Minimum LSP BW: 155.52Mbps
        Interface MTU: 2595
NodeID: 10.255.255.37
  Type: Rtr, Age: 2852 secs, LinkIn: 5, LinkOut: 5
  Protocol: OSPF(0.0.0.0)
    To: 10.255.255.69, Local: 10.35.150.1, Remote: 10.35.150.2
      Metric: 1
```

```
      Static BW: 622.08Mbps
      Reservable BW: 622.08Mbps
      Available BW [priority] bps:
       [0] 622.08Mbps   [1] 622.08Mbps   [2] 622.08Mbps   [3] 622.08Mbps
       [4] 622.08Mbps   [5] 622.08Mbps   [6] 622.08Mbps   [7] 622.08Mbps
      Interface Switching Capability Descriptor(1):
        Switching type: Fiber
        Encoding type: SDH/SONET
        Maximum LSP BW [priority] bps:
         [0] 622.08Mbps   [1] 622.08Mbps   [2] 622.08Mbps   [3] 622.08Mbps
         [4] 622.08Mbps   [5] 622.08Mbps   [6] 622.08Mbps   [7] 622.08Mbps
        Minimum LSP BW: 622.08Mbps
        Interface MTU: 2597
    To: 10.255.255.40, Local: 10.35.200.2, Remote: 10.35.200.1
      Metric: 1
      Static BW: 155.52Mbps
      Reservable BW: 155.52Mbps
      Available BW [priority] bps:
       [0] 0bps         [1] 0bps         [2] 0bps         [3] 0bps
       [4] 0bps         [5] 0bps         [6] 0bps         [7] 0bps
      Interface Switching Capability Descriptor(1):
        Switching type: Fiber
        Encoding type: SDH/SONET
        Maximum LSP BW [priority] bps:
         [0] 155.52Mbps   [1] 155.52Mbps   [2] 155.52Mbps   [3] 155.52Mbps
         [4] 155.52Mbps   [5] 155.52Mbps   [6] 155.52Mbps   [7] 155.52Mbps
        Minimum LSP BW: 155.52Mbps
        Interface MTU: 2600
NodeID: 10.255.255.40
  Type: Rtr, Age: 2854 secs, LinkIn: 2, LinkOut: 2
  Protocol: OSPF(0.0.0.0)
    To: 10.255.255.37, Local: 10.35.200.1, Remote: 10.35.200.2
      Metric: 1
      Static BW: 155.52Mbps
      Reservable BW: 155.52Mbps
      Available BW [priority] bps:
       [0] 0bps         [1] 0bps         [2] 0bps         [3] 0bps
       [4] 0bps         [5] 0bps         [6] 0bps         [7] 0bps
      Interface Switching Capability Descriptor(1):
        Switching type: Fiber
        Encoding type: SDH/SONET
        Maximum LSP BW [priority] bps:
         [0] 155.52Mbps   [1] 155.52Mbps   [2] 155.52Mbps   [3] 155.52Mbps
         [4] 155.52Mbps   [5] 155.52Mbps   [6] 155.52Mbps   [7] 155.52Mbps
        Minimum LSP BW: 155.52Mbps
        Interface MTU: 2600
NodeID: 10.255.255.69
  Type: Rtr, Age: 2832 secs, LinkIn: 8, LinkOut: 7
  Protocol: OSPF(0.0.0.0)
    To: 10.255.255.35, Local: 10.35.100.1, Remote: 10.35.100.2
      Metric: 1
      Static BW: 155.52Mbps
      Reservable BW: 155.52Mbps
      Available BW [priority] bps:
       [0] 0bps         [1] 0bps         [2] 0bps         [3] 0bps
       [4] 0bps         [5] 0bps         [6] 0bps         [7] 0bps
      Interface Switching Capability Descriptor(1):
        Switching type: Fiber
        Encoding type: SDH/SONET
        Maximum LSP BW [priority] bps:
         [0] 155.52Mbps   [1] 155.52Mbps   [2] 155.52Mbps   [3] 155.52Mbps
```

```
          [4] 155.52Mbps   [5] 155.52Mbps   [6] 155.52Mbps   [7] 155.52Mbps
        Minimum LSP BW: 155.52Mbps
        Interface MTU: 2595
    To: 10.255.255.37, Local: 10.35.150.2, Remote: 10.35.150.1
      Metric: 1
      Static BW: 622.08Mbps
      Reservable BW: 622.08Mbps
      Available BW [priority] bps:
       [0] 622.08Mbps   [1] 622.08Mbps   [2] 622.08Mbps   [3] 622.08Mbps
       [4] 622.08Mbps   [5] 622.08Mbps   [6] 622.08Mbps   [7] 622.08Mbps
      Interface Switching Capability Descriptor(1):
        Switching type: Fiber
        Encoding type: SDH/SONET
        Maximum LSP BW [priority] bps:
         [0] 622.08Mbps   [1] 622.08Mbps   [2] 622.08Mbps   [3] 622.08Mbps
         [4] 622.08Mbps   [5] 622.08Mbps   [6] 622.08Mbps   [7] 622.08Mbps
        Minimum LSP BW: 622.08Mbps
        Interface MTU: 2597

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

Router C Status

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

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

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

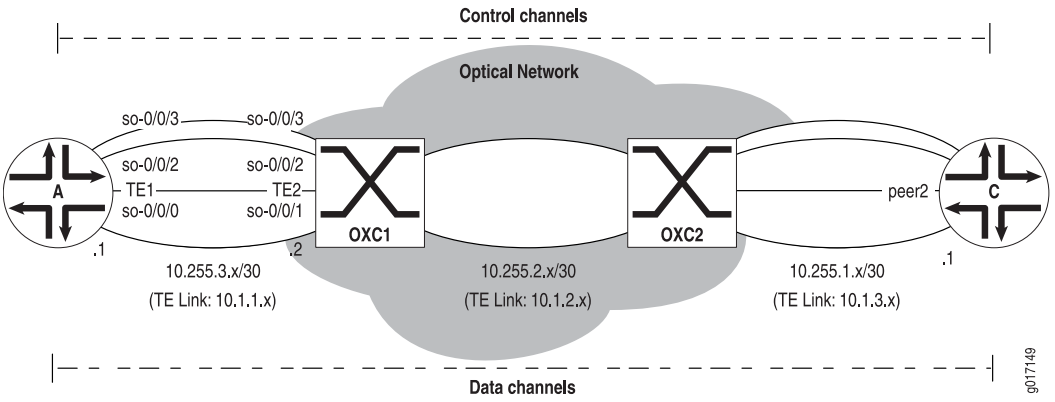
After you configure both remote IDs on both peers, the GMPLS TE links should work. Using the same command as before, you can determine whether the link is functional, with both remote and local IDs in place:

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

The other show commands operate like those in “Router A Status” on page 23.

Example: Configuring TE Link and Interface Identifiers

Figure 3: TE Link and Interface ID Example



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

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

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

Figure 3 on page 29 shows where the IDs come from and where you must assign them. This example highlights the connections between Router A and OXC1, but the same configuration concepts apply to all pairs of peers.

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

If Router A has a local ID of 12345 for its TE link and a local ID of 23456 for its interface, you must configure 12345 as the TE link remote-ID and 23456 as the interface remote-ID on the TE2 TE link of OXC1. Similarly, if OXC1 has local IDs of

98765 for its TE link and 54321 for its interface, you configure Router A's TE1 TE link with 98765 as the TE link remote-ID and 54321 as the interface remote-ID.

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

Example: LMP Control Channel Configuration

Figure 4: LMP Control Channel Topology Diagram

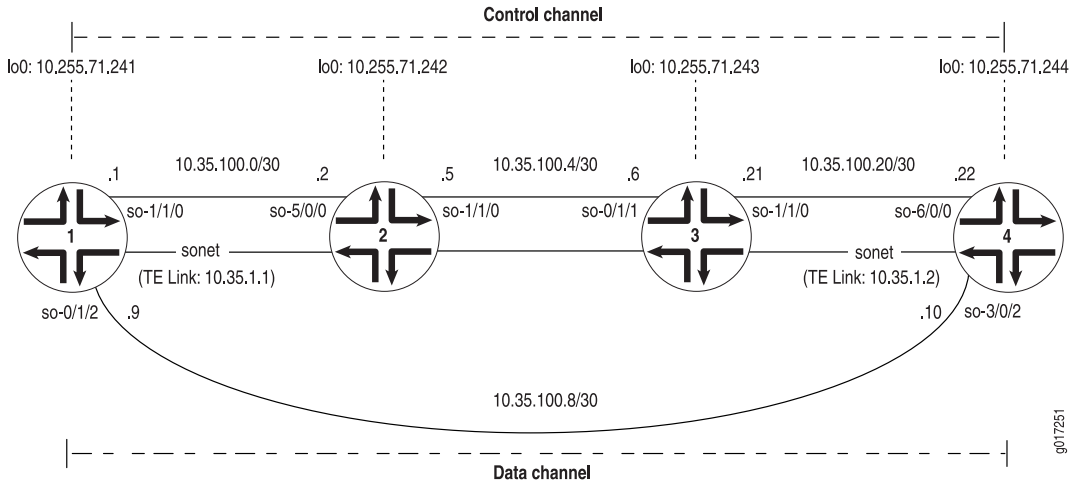


Figure 4 on page 30 shows a series of four routers that are used to establish an LMP control channel and TE link between Routers R1 and R4. The control channel originates on the so-1/1/0 interface of Router 1 and ends at the so-6/0/0 interface of Router 4. The TE link data channel is a point-to-point link from the so-0/1/2 interface of Router R1 to the so-3/0/2 interface of Router R4.

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

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

Router 1

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

```
            family mpls;
        }
    }
    so-1/1/0 {
        description “Control channel to Router 4”;
        unit 0 {
            family inet {
                address 10.35.100.1/30;
            }
            family iso;
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.71.241/32;
            }
            family iso {
                address 47.0005.0000.0000.0000.0000.0000.0102.5507.1241.00;
            }
        }
    }
}
routing-options {
    router-id 10.255.71.241;
}
protocols {
    rsvp {
        interface all;
    }
    mpls {
        label-switched-path gmpls-router1-router4 {
            to 10.255.71.244;
            lsp-attributes {
                switching-type fiber;
            }
            no-cspf;
            primary path1;
        }
        path path1 {
            10.35.1.2 strict;
        }
        interface so-1/1/0.0;
    }
    isis {
        interface so-1/1/0.0;
        interface fxp0.0 {
            disable;
        }
        interface lo0.0 {
            passive;
        }
    }
    link-management {
        te-link sonet {
```

```
        local-address 10.35.1.1;
        remote-address 10.35.1.2;
        remote-id 8070;
        interface so-0/1/2 {
            remote-id 21303;
        }
    }
    peer router4 {
        address 10.255.71.244;
        lmp-control-channel so-1/1/0.0 {
            remote-address 10.35.100.22;
        }
        te-link sonet;
    }
    traceoptions {
        file lmp.logs size 5m files 10 world-readable;
        flag hello-packets;
        flag packets;
        flag state;
    }
  }
}
```

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

Router 2

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

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

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

Router 3

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

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

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

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

Router 4 [edit]

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

```
            interface so-3/0/2 {
               remote-id 22279;
            }
         }
         peer router1 {
            address 10.255.71.241;
            lmp-control-channel so-6/0/0.0 {
               remote-address 10.35.100.1;
            }
            te-link sonet;
         }
         traceoptions {
            file lmp.logs size 5m files 10 world-readable;
            flag hello-packets;
            flag packets;
            flag state;
         }
      }
   }
```

Verifying Your Work

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

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

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

- Router 1 Status on page 36
- Router 4 Status on page 37

Router 1 Status

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

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

```
user@router1> show link-management statistics
Statistics for peer router4
    Received packets
        ConfigAck: 1
        Hello: 748
        LinkSummary: 13
        LinkSummaryAck: 1
    Small packets: 0
    Wrong protocol version: 0
    Messages for unknown peer: 0
    Messages for bad state: 0
    Stale acknowledgements: 0
    Stale negative acknowledgements: 0
    Sent packets
        Config: 24
        Hello: 748
        LinkSummary: 13
        LinkSummaryAck: 13
    Retransmitted packets
        Config: 18
        LinkSummary: 9
    Dropped packets
        Config: 5
        LinkSummary: 3

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

Router 4 Status

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

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

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

```
    Stale negative acknowledgements: 0
    Sent packets
        Config: 31
        ConfigAck: 1
        Hello: 255
        LinkSummary: 13
        LinkSummaryAck: 1
    Retransmitted packets
        Config: 23
        LinkSummary: 9
    Dropped packets
        Config: 7
        LinkSummary: 3

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

For More Information

For additional information about implementing GMPLS, see the following:

- *JUNOS MPLS Applications Configuration Guide*
- RFC 2205, *Resource ReSerVation Protocol (RSVP)*
- RFC 3471, *Generalized Multi-Protocol Label Switching (GMPLS)—Signaling Functional Description*
- RFC 3473, *Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions*
- RFC 4204, *Link Management Protocol (LMP)* (The JUNOS software supports sections 3 and 4)
- Internet draft draft-ietf-ccamp-gmpls-rsvp-te-ason-02.txt, *Generalized MPLS (GMPLS) RSVP-TE Signalling in support of Automatically Switched Optical Network (ASON)* (expires January 2005)
- Internet draft draft-ietf-ccamp-gmpls-sonet-sdh-08.txt, *GMPLS Extensions for SONET and SDH Control* (expires August 2003)
- Internet draft draft-ietf-ccamp-ospf-gmpls-extensions-12.txt, *OSPF Extensions in Support of Generalized MPLS* (expires April 2004)
- Internet draft draft-ietf-mpls-bundle-04.txt, *Link Bundling in MPLS Traffic Engineering* (expires January 2003)
- Internet draft draft-ietf-mpls-lsp-hierarchy-08.txt, *LSP Hierarchy with Generalized MPLS TE* (expires March 2003)
- Internet draft draft-ietf-ccamp-gmpls-routing-09.txt, *Routing Extensions in Support of Generalized MPLS* (expires April 2004)

Revision History

13 April 2009—9.5R1 Release. Roy Spencer.

15 January 2009—9.4R1 Release. Roy Spencer.

10 October 2008—9.3R1 Release. Roy Spencer.

10 July 2008—9.2R1 Release. Roy Spencer.

10 April 2008—9.1R1 Release. Roy Spencer.

1 February 2008—9.0R1 Release. Fawn Damitio.

27 March 2007—8.3R1 Release. Fawn Damitio.

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

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

29 June 2006—8.0R1 Release. Richard Hendricks.

27 March 2006—7.6R1 Release. Richard Hendricks.

9 January 2006—7.5R1 Release. Richard Hendricks.

14 September 2005—7.4R1 Release. Richard Hendricks.

13 June 2005—7.3R1 Release. Richard Hendricks.

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

2 February 2005—7.1R1 Release. Richard Hendricks.

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

6 July 2004—6.4R1 Release. Richard Hendricks.

5 April 2004—6.3R1 Release. Richard Hendricks.

22 December 2003—6.2R1 Release. Richard Hendricks.

22 September 2003—6.1R1 Release. Richard Hendricks.

30 June 2003—6.0R1 Release. Richard Hendricks.

2 April 2003—5.7R1 Release. Richard Hendricks.

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

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

19 July 2002—5.4R1 Release. Richard Hendricks.

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

6 May 2002—Initial document written. Richard Hendricks.

Chapter 2

Connecting IPv6 Islands with IPv4 MPLS

This feature guide covers these topics:

Overview

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

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

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

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

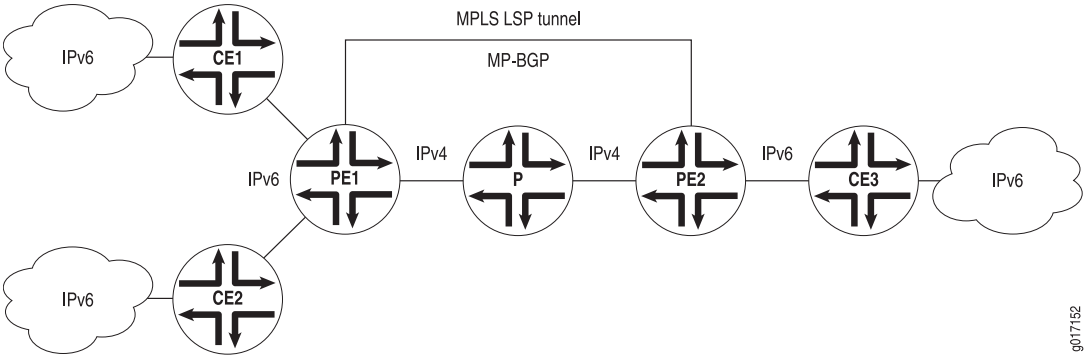
In Figure 5 on page 42, PE1 and PE2 are dual-stack Border Gateway Protocol (DS-BGP) routers. They implement IPv4 and IPv6 stacks simultaneously. The IPv6 clouds are separate islands that are connected to PE routers through a customer edge (CE) router.

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

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

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

Figure 5: Connecting IPv6 Islands over MPLS



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

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

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

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

In Figure 5 on page 42, PE1 receives an IPv6 packet from CE1 and performs a lookup in the IPv6 forwarding table. If the destination matches a prefix that was learned from CE2, no labels are necessary and the IPv6 packet is sent to CE2. If the destination matches a prefix that was learned from PE2, then PE1 places two labels on the packet and sends it to P. The inner label is 2 and the outer label is the LSP label needed to reach PE2. Since P is the penultimate-hop router for the LSP to PE2 and the received packet has more than one label, Router P pops the outer label and sends the packet to PE2. When PE2 receives the packet, it has a single label with a value of 2. PE2 strips off the label and treats the remaining packet as an IPv6 packet (since 2 is the IPv6 explicit null label) and performs a lookup in the IPv6 forwarding table.

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

Interconnecting IPv6 islands over an IPv4 MPLS tunnel requires:

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

System Requirements

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

- JUNOS Release 8.2 or later for MX-series routing platforms
- JUNOS Release 7.2 or later for J-series Services Routers

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

Terms and Acronyms

D

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

M

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

S

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

Configuring an IPv4 MPLS Tunnel to Carry IPv6 Traffic

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

- Configuring IPv6 on the Customer and Core-Facing Interfaces on page 44
- Configuring MPLS and RSVP from PE Router to PE Router to Create a Tunnel on page 45
- Enabling IPv6 Tunneling in MPLS on page 45
- Configuring Multiprotocol BGP to Carry IPv6 Traffic on page 45

Configuring IPv6 on the Customer and Core-Facing Interfaces

Configure family inet6 on all the CE-facing interfaces and on all the core-facing interfaces running MPLS. This enables the router to process any IPv6 packets it receives on these interfaces. You should not see any regular IPv6 traffic arrive on these interfaces, but you will receive MPLS packets tagged with label 2. Even though label 2 MPLS packets are sent in IPv4, these packets are treated as native IPv6 packets.

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

```
        }
      }
    }
}
```

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

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

Enabling IPv6 Tunneling in MPLS

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

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

Configuring Multiprotocol BGP to Carry IPv6 Traffic

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

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

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

Example: Connecting IPv6 Islands over an MPLS Tunnel Configuration

Figure 6: IPv6 over an MPLS Tunnel

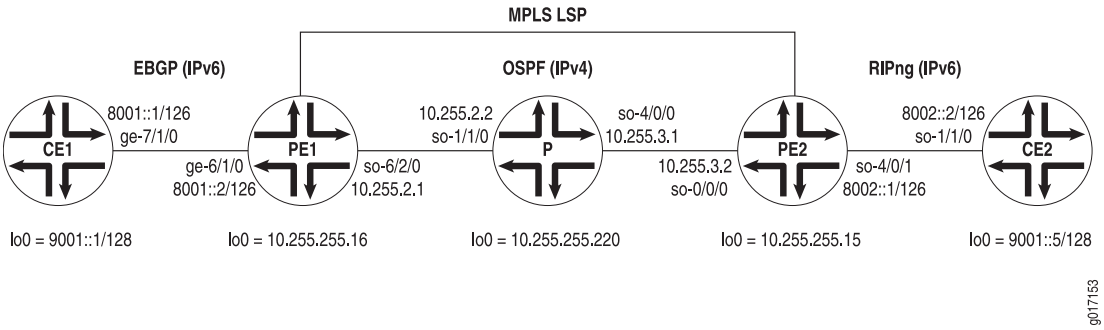


Figure 6 on page 46 shows a standard CE-PE-P-PE-CE MPLS-style network. CE1 and CE2 are the end customer CE routers using IPv6; PE1 and PE2 are the provider edge routers; and P is a provider core router. The IPv4 MPLS tunnel travels between PE1 and PE2, connecting IPv6 sites CE1 and CE2.

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

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

Router CE1

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

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

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

Router PE1

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

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

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

Router P [edit]

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

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

bgp family inet6] hierarchy level. Finally, create and apply policies that export BGP routes into RIPng and import RIPng routes to BGP.

Router PE2

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

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

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

Router CE2

```
[edit]
interfaces {
  so-1/1/0 {
    unit 0 {
      family inet6 {
        address 8002::2/126;
      }
    }
  }
  lo0 {
    unit 0 {
      family inet6 {
        address 9001::5/128;
      }
    }
  }
}
routing-options {
  autonomous-system 300;
```

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

Verifying Your Work

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

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

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

- Router CE1 Status on page 53
- Router PE1 Status on page 53
- Router PE2 Status on page 54
- Router CE2 Status on page 55

Router CE1 Status

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

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

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

Router PE1 Status

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

  inet6.0: 1/1/0

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

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

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

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

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

Router PE2 Status

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

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

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

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

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

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

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

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

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

Router CE2 Status

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

user@CE2> show route protocol ripng

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

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

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

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

For More Information

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

- *JUNOS MPLS Applications Configuration Guide*
- *JUNOS Routing Protocols Configuration Guide*
- RFC 3032, *MPLS Label Stack Encoding*
- RFC 3107, *Carrying Label Information in BGP-4*
- RFC 3513, *IP Version 6 Addressing Architecture*
- Internet draft draft-ietf-l3vpn-bgp-ipv6-07.txt, *BGP-MPLS IP VPN extension for IPv6 VPN* (expires January 2006)
- Internet draft draft-ooms-v6ops-bgp-tunnel-06.txt, *Connecting IPv6 Islands over IPv4 MPLS using IPv6 Provider Edge Routers (6PE)* (expires July 2006)

Revision History

13 April 2009—9.5R1 Release. Roy Spencer.

15 January 2009—9.4R1 Release. Roy Spencer.

10 October 2008—9.3R1 Release. Roy Spencer.

10 July 2008—9.2R1 Release. Roy Spencer.

10 April 2008—9.1R1 Release. Roy Spencer.

1 February 2008—9.0R1 Release. Fawn Damitio.

27 March 2007—8.3R1 Release. Fawn Damitio.

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

15 September 2006—8.1R1 Release. Richard Hendricks.

29 June 2006—8.0R1 Release. Richard Hendricks.

27 March 2006—7.6R1 Release. Richard Hendricks.

9 January 2006—7.5R1 Release. Richard Hendricks.

14 September 2005—7.4R1 Release. Richard Hendricks.

13 June 2005—7.3R1 Release. Richard Hendricks.

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

2 February 2005—7.1R1 Release. Richard Hendricks.

6 October 2004—7.0R1 Release. Richard Hendricks.

6 July 2004—6.4R1 Release. Richard Hendricks.

5 April 2004—6.3R1 Release. Richard Hendricks.

22 December 2003—6.2R1 Release. Richard Hendricks.

22 September 2003—6.1R1 Release. Richard Hendricks.

30 June 2003—6.0R1 Release. Richard Hendricks.

2 April 2003—5.7R1 Release. Richard Hendricks.

27 December 2002—5.6R1 Release. Richard Hendricks.

30 September 2002—5.5R1 Release. Richard Hendricks.

19 July 2002—5.4R1 Release. Richard Hendricks.

6 May 2002—Initial document written. Richard Hendricks.

Chapter 3

Multiple Instances for Label Distribution Protocol

This feature guide covers these topics:

Overview

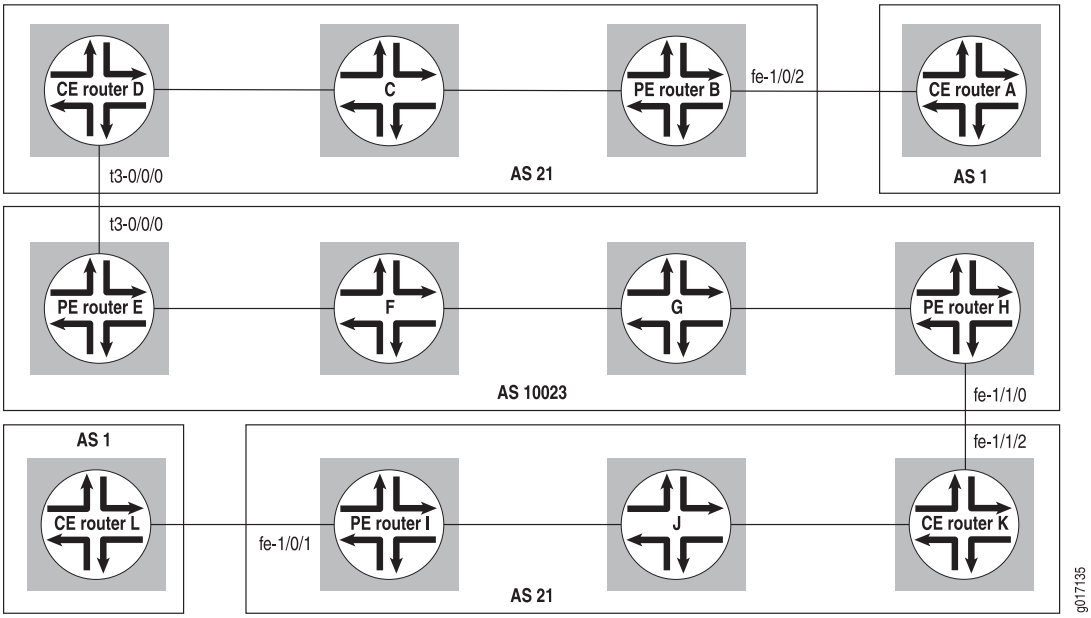
Previous versions of JUNOS software support multiple VPN routing and forwarding (VRF) instances of Border Gateway Protocol (BGP), Intermediate System-to-Intermediate System (IS-IS), Open Shortest Path First (OSPF), Protocol Independent Multicast (PIM), and Routing Information Protocol (RIP). JUNOS Release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

This support allows LDP to be used to advertise labels in a carrier-of-carriers scenario from a core provider edge (PE) router to a customer carrier edge (CE) router. This is especially useful when the carrier customer is a basic Internet service provider (ISP) and wants to restrict full Internet routes to its PE routers. By using LDP instead of BGP, the carrier customer shields its other internal routers from the Internet at large. Multiple-instance LDP is also useful when a carrier customer wants to provide Layer 2 VPN or Layer 3 VPN services to its customers.

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

In Figure 7 on page 60, the customer carrier in AS 21 can configure one instance of LDP for all routers in AS 21 instead of using full-mesh IBGP.

Figure 7: Carrier-of-Carriers Example



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

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

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

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

System Requirements

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

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

Terms and Acronyms

C

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

L

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

V

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

Example: Configuring Multiple-Instance LDP

The master LDP instance is configured at the [edit protocols] hierarchy level.

You can configure a specific instance of LDP by using the ldp statement at the [edit routing-instances *routing-instance-name* protocols] hierarchy level. This creates an instance of LDP for the particular VRF routing instance. You must specify all the required VRF statements and apply export and import policies to your LDP instance for the configuration to commit properly.

Most of the LDP hierarchy levels available in a master instance are also available for specific instances of LDP. However, the no-forwarding option does not work in a VRF-based instance of LDP.

Figure 8: Multiple-Instance LDP Topology Diagram

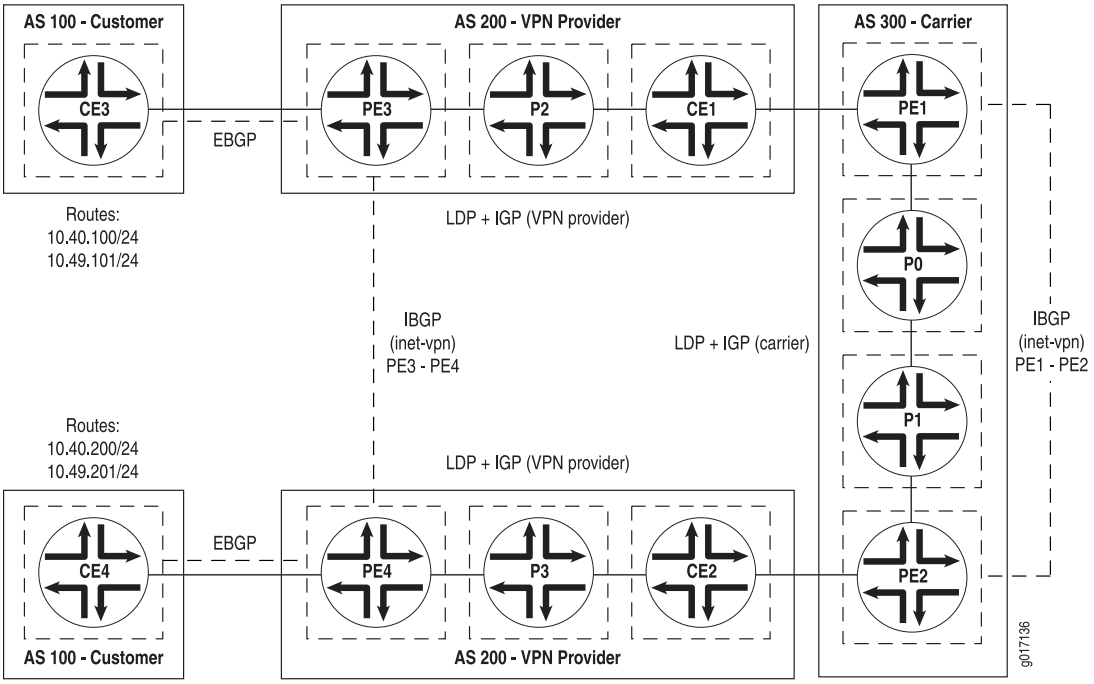


Figure 8 on page 62 shows an example of a carrier-of-carriers network. CE3 and CE4 are end customer CE routers residing in AS 100. The VPN provider in AS 200 has three types of routers: PE3 and PE4 are PE routers that connect to the end customer, CE1 and CE2 act as the intermediate carrier CE routers, and P2 and P3 are internal transit routers. PE1 and PE2 in AS 300 are PE routers servicing the intermediate VPN provider, and P0 and P1 are transit routers for the top tier carrier.

To make this configuration work, you must complete three major tasks:

1. Configure external BGP between the VPN customer CE and the VPN provider PE.
2. Configure internal BGP using the VPN family between both pairs of PE routers (one IBGP connection between PE1 and PE2 and a second IBGP connection between PE3 and PE 4).
3. Establish LDP and Interior Gateway Protocol (IGP) connections on all remaining links. This example uses OSPF as the IGP, but you can use the IGP of your choice.

Information supporting this carrier-of-carriers multiple-instance LDP example is summarized in Table 5 on page 62 and Table 6 on page 63.

Table 5: Multiple-Instance LDP Example—Routing Protocol Summary

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

Table 5: Multiple-Instance LDP Example—Routing Protocol Summary *(continued)*

Connection	Protocols
CE1 - PE1	OSPF and LDP
PE1 - P0 - P1 - PE2	OSPF and LDP
PE1 - PE2	IBGP family inet-vpn
PE2 - CE2	OSPF and LDP
CE2 - P3 - PE4	OSPF and LDP
PE4 - CE4	EBGP family inet
PE3 - PE4	IBGP family inet-vpn

Table 6: Multiple-Instance LDP Example—Loopback Addresses

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

Your configuration tasks start at CE3 and move router by router through the first part of the VPN provider network, into the carrier AS, through the second VPN provider cluster of AS 200, and end at the second VPN customer Router CE4.

Since CE3 is the first customer router, configure EBGP between CE3 and the connected VPN provider Router PE3. You must also advertise your loopback address into BGP with a routing policy to allow IP reachability with CE4.

Router CE3

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

```
        }
    }
}
```

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

Finally, set up an outbound VRF policy that places all BGP traffic and directly connected interfaces into a BGP community and an inbound VRF policy that accepts similar BGP community traffic from PE4.

Router PE3

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

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

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

On P2, enable LDP and the IGP used for transporting labels (in this case, OSPF). You will repeat these tasks on all transit core routers, both in the VPN provider network and the core carrier network.

Router P2

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

```
}
```

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

Router CE1

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

On core carrier Router PE1, configure a master instance for OSPF, LDP, MPLS, and IBGP (with the family inet-vpn option) to connect the router to neighbor PE2. Next, implement multiple-instance LDP by establishing a secondary instance. Enable LDP

and OSPF in this instance for PE1 to communicate with CE1. MPLS is not required in the secondary instance.

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

Router PE1

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

```
        area 0.0.0.0 {
            interface lo0.0 {
                passive;
            }
            interface so-0/0/0.0;
        }
    }
    ldp {
        interface so-0/0/0.0;
    }
}
policy-options {
    policy-statement vpn-provider-import {
        term 1 {
            from {
                protocol bgp;
                community vpn-provider-comm;
            }
            then accept;
        }
        term 2 {
            then reject;
        }
    }
    policy-statement vpn-provider-export {
        term 1 {
            from protocol ldp;
            then {
                community add vpn-provider-comm;
                accept;
            }
        }
        term 2 {
            then reject;
        }
    }
    policy-statement bgp-routes-to-export {
        term 1 {
            from {
                protocol bgp;
                community vpn-provider-comm;
            }
            then accept;
        }
        term 2 {
            then reject;
        }
    }
    community vpn-provider-comm members target:300:200;
}
routing-instances {
    vpn-provider {
        instance-type vrf;
        interface t3-0/2/1.0;
        route-distinguisher 10.255.255.171:1;
        vrf-import vpn-provider-import;
```

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

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

Router P0

```
[edit]
interfaces {
    so-0/1/0 {
        description "to pe1 so-0/0/0";
        unit 0 {
            family inet {
                address 192.255.197.22/30;
            }
            family mpls;
        }
    }
    so-1/0/0 {
        description "to p1 so-1/0/0";
        unit 0 {
            family inet {
                address 192.255.197.85/30;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.255.173/32;
            }
        }
    }
}
routing-options {
    autonomous-system 300;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/1/0.0;
            interface so-1/0/0.0;
            interface lo0.0 {
```

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

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

Router P1

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

```
    }
  }
```

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

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

Router PE2

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

```
        }
        peer-as 300;
        neighbor 10.255.255.171;
      }
    }
    ospf {
      area 0.0.0.0 {
        interface so-0/2/0.0;
        interface lo0.0 {
          passive;
        }
      }
    }
    ldp {
      interface so-0/2/0.0;
    }
  }
  policy-options {
    policy-statement vpn-provider-import {
      term 1 {
        from {
          protocol bgp;
          community vpn-provider-comm;
        }
        then accept;
      }
      term 2 {
        then reject;
      }
    }
    policy-statement vpn-provider-export {
      term 1 {
        from protocol ldp;
        then {
          community add vpn-provider-comm;
          accept;
        }
      }
      term 2 {
        then reject;
      }
    }
    policy-statement bgp-routes-to-export {
      term 1 {
        from {
          protocol bgp;
          community vpn-provider-comm;
        }
        then accept;
      }
      term 2 {
        then reject;
      }
    }
    community vpn-provider-comm members target:300:200;
  }
```

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

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

Router CE2

```
[edit]
interfaces {
   t1-0/0/0 {
      description "to pe2 t1-3/0/0";
      unit 0 {
         family inet {
            address 192.255.197.38/30;
         }
         family mpls;
      }
   }
   t3-0/3/3 {
      description "to p3 t3-0/0/3";
      unit 0 {
         family inet {
            address 192.255.198.26/30;
         }
         family mpls;
      }
      lo0 {
         unit 0 {
            family inet {
               address 10.255.255.180/32;
            }
         }
      }
   }
   routing-options {
      autonomous-system 200;
   }
   protocols {
      ospf {
```

```
            area 0.0.0.0 {
               interface t1-0/0/0.0;
               interface t3-0/3/3.0;
               interface lo0.0 {
                  passive;
               }
            }
         }
         ldp {
            interface t1-0/0/0.0;
            interface t3-0/3/3.0;
         }
      }
   }
```

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

Router P3

```
[edit]
interfaces {
   t3-0/0/3 {
      description "to ce2 t3-0/3/3";
      unit 0 {
         family inet {
            address 192.255.198.25/30;
         }
         family mpls;
      }
   }
   t1-0/1/1 {
      description "to pe4 t1-0/1/1";
      unit 0 {
         family inet {
            address 192.255.198.37/30;
         }
         family mpls;
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.255.255.176/32;
         }
      }
   }
}
routing-options {
   autonomous-system 200;
}
protocols {
   ospf {
      area 0.0.0.0 {
         interface t3-0/0/3.0;
         interface t1-0/1/1.0;
         interface lo0.0 {
            passive;
```

```
        }
      }
    }
    ldp {
      interface t3-0/0/3.0;
      interface t1-0/1/1.0;
    }
  }
```

On PE4, complete the IBGP connection initiated on PE3 to connect the edge routers in AS 200. Also, enable LDP and MPLS on the t1-0/0/1 interface pointing toward the VPN provider CE2 router and establish an EBGP connection to CE4 through use of a VRF instance.

Finally, set up an outbound VRF policy that places all BGP traffic and directly connected interfaces into a BGP community and an inbound VRF policy that accepts similar BGP community traffic from PE3.

Router PE4

```
[edit]
interfaces {
  t3-0/0/3 {
    description to ce4 t3-0/0/3";
    unit 0 {
      family inet {
        address 192.255.198.21/30;
      }
      family mpls;
    }
  }
  t1-0/1/1 {
    unit 0 {
      family inet {
        address 192.255.198.38/30;
      }
      family mpls;
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.255.255.178/32;
      }
    }
  }
}
routing-options {
  autonomous-system 200;
}
protocols {
  mpls {
    interface t3-0/0/3.0;
  }
  bgp {
    group internal {
      type internal;
```

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

```
        bgp {
          group customer {
            type external;
            peer-as 100;
            as-override;
            neighbor 192.255.198.22;
          }
        }
      }
    }
```

CE4 is the destination VPN customer router. Configure EBGP between CE4 and the connected VPN provider Router PE4 to complete the configuration. Remember to advertise the loopback address into BGP by using a routing policy to allow IP reachability with CE3.

Router CE4

```
[edit]
interfaces {
  t3-0/0/3 {
    description "to pe4 t3-0/0/3";
    unit 0 {
      family inet {
        address 192.255.198.22/30;
      }
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.255.255.182/32;
        address 10.49.200.1/32;
      }
    }
  }
}
routing-options {
  static {
    route 10.49.200.0/24 reject;
    route 10.49.201.0/24 reject;
  }
  autonomous-system 100;
}
protocols {
  bgp {
    group provider {
      type external;
      export static-to-bgp;
      peer-as 200;
      neighbor 192.255.198.21;
    }
  }
}
policy-options {
  policy-statement static-to-bgp {
    term 1 {
```

```
            from {
                protocol static;
                route-filter 10.49.200.0/24 exact;
                route-filter 10.49.201.0/24 exact;
            }
            then accept;
        }
        term 2 {
            from protocol direct;
            then accept;
        }
        term 3 {
            then reject;
        }
    }
}
```

Verifying Your Work

To verify the proper operation of your multiple-instance LDP configuration, use the following commands:

- show ldp database
- show ldp interface
- show ldp neighbor
- show ldp path
- show ldp route
- show ldp session
- show ldp statistics

The display output for these commands is the same as in previous JUNOS software releases, except for one difference. An instance name can now be used as an argument.

If you include an instance name with these commands, you display information for the specified LDP instance. For example, the command show ldp neighbor instance crockett shows all the LDP neighbors for a VRF instance named crockett. Conversely, show ldp neighbor without an instance name displays the LDP neighbors associated with the master instance.

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

- Router CE3 Status on page 81
- Router PE3 Status on page 81
- Router CE1 Status on page 83
- Router PE1 Status on page 84
- Router PE2 Status on page 86

- Router CE2 Status on page 91
- Router PE4 Status on page 93
- Router CE4 Status on page 95

Router CE3 Status

```
user@CE3> show bgp summary

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

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

Router PE3 Status

```
user@PE3> show bgp summary

Groups: 2 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l3vpn.0            6          6          0          0          0          0
Peer               AS      InPkt     OutPkt    OutQ   Flaps Last Up/DwnState|#Active/Received/Damped...
192.255.198.14    100        432        441       0       0     3:34:55 Establ
  vpn-customer.inet.0: 5/6/0
10.255.255.178    200         62         63       0       2       27:23 Establ
  bgp.l3vpn.0: 6/6/0
  vpn-customer.inet.0: 5/6/0

user@PE3>  show route protocol ldp
inet.0: 19 destinations, 20 routes (18 active, 0 holddown, 1 hidden)
inet.3: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.255.255.175/32  *[LDP/9] 03:35:45, metric 1
                    > via so-1/2/1.0
10.255.255.176/32  *[LDP/9] 00:29:32, metric 1
                    > via so-1/2/1.0, Push 100007
```

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

user@PE3>  show route protocol bgp
inet.0: 19 destinations, 20 routes (18 active, 0 holddown, 1 hidden)
inet.3: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
vpn-customer.inet.0: 12 destinations, 14 routes (12 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.49.100.0/24     *[BGP/170] 03:34:59, MED 0, localpref 100
                      AS path: 100 I
                    > to 192.255.198.14 via so-1/2/0.0
10.49.100.1/32     *[BGP/170] 03:34:59, localpref 100
                      AS path: 100 I
                    > to 192.255.198.14 via so-1/2/0.0
10.49.101.0/24     *[BGP/170] 03:34:59, MED 0, localpref 100
                      AS path: 100 I
                    > to 192.255.198.14 via so-1/2/0.0
10.49.200.0/24     *[BGP/170] 00:26:39, MED 0, localpref 100, from 10.255.255.178
                      AS path: 100 I
                    > via so-1/2/1.0, Push 100019, Push 100008(top)
10.49.200.1/32     *[BGP/170] 00:26:39, localpref 100, from 10.255.255.178
                      AS path: 100 I
                    > via so-1/2/1.0, Push 100019, Push 100008(top)
10.49.201.0/24     *[BGP/170] 00:26:39, MED 0, localpref 100, from 10.255.255.178
                      AS path: 100 I
                    > via so-1/2/1.0, Push 100019, Push 100008(top)
10.255.255.181/32  *[BGP/170] 03:34:59, localpref 100
                      AS path: 100 I
                    > to 192.255.198.14 via so-1/2/0.0
10.255.255.182/32  *[BGP/170] 00:26:39, localpref 100, from 10.255.255.178
                      AS path: 100 I
                    > via so-1/2/1.0, Push 100019, Push 100008(top)
192.255.14.0/24    *[BGP/170] 03:34:59, localpref 100
                      AS path: 100 I
                    > to 192.255.198.14 via so-1/2/0.0
                     [BGP/170] 00:26:39, localpref 100, from 10.255.255.178
                      AS path: 100 I
                    > via so-1/2/1.0, Push 100019, Push 100008(top)
192.255.198.12/30   [BGP/170] 03:34:59, localpref 100
                      AS path: 100 I
                    > to 192.255.198.14 via so-1/2/0.0
```

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

Router CE1 Status

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

user@CE1>  show route

inet.0: 21 destinations, 23 routes (20 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both
0.0.0.0/0          *[Static/5] 07:53:10, metric 0
                      Discard
10.255.255.175/32  *[OSPF/10] 00:31:44, metric 1
                    > via so-1/2/0.0
10.255.255.176/32  *[OSPF/150] 00:31:44, metric 1, tag 3489661228
                    > via t3-0/1/0.0
10.255.255.177/32  *[OSPF/10] 00:31:44, metric 2
                    > via so-1/2/0.0
10.255.255.178/32  *[OSPF/150] 00:31:44, metric 1, tag 3489661228
                    > via t3-0/1/0.0
10.255.255.179/32  *[Direct/0] 07:53:10
                    > via lo0.0
10.255.255.180/32  *[OSPF/150] 00:31:44, metric 1, tag 3489661228
                    > via t3-0/1/0.0
172.16.0.0/12      *[Static/5] 07:53:10
                    > to 192.255.14.254 via fxp0.0
```

```
192.255.0.0/18     *[Static/5] 07:53:10
                    > to 192.255.14.254 via fxp0.0
192.255.14.0/24    *[Direct/0] 07:53:10
                    > via fxp0.0
192.255.14.179/32  *[Local/0] 07:53:10
                      Local via fxp0.0
192.255.40.0/22    *[Static/5] 03:38:37
                    > to 192.255.14.254 via fxp0.0
192.255.64.0/18    *[Static/5] 03:38:37
                    > to 192.255.14.254 via fxp0.0
192.255.197.16/30  *[Direct/0] 03:37:42
                    > via t3-0/1/0.0
                    [OSPF/10] 00:31:44, metric 2
                    > via t3-0/1/0.0
192.255.197.18/32  *[Local/0] 07:52:01
                      Local via t3-0/1/0.0
192.255.198.0/30   *[Direct/0] 07:51:18
                    > via so-1/2/0.0
                    [OSPF/10] 00:31:44, metric 1
                    > via so-1/2/0.0
192.255.198.1/32   *[Local/0] 07:51:59
                      Local via so-1/2/0.0
192.255.198.8/30   *[OSPF/10] 00:31:44, metric 2
                    > via so-1/2/0.0
207.17.136.192/32  *[Static/5] 07:53:10
                    > to 192.255.14.254 via fxp0.0
224.0.0.5/32       *[OSPF/10] 07:53:14, metric 1
                      MultiRecv
inet.3: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.255.255.175/32  *[LDP/9] 01:00:52, metric 1
                    > via so-1/2/0.0
10.255.255.176/32  *[LDP/9] 00:33:24, metric 1
                    > via t3-0/1/0.0, Push 100020
10.255.255.177/32  *[LDP/9] 01:00:52, metric 1
                    > via so-1/2/0.0, Push 100000
10.255.255.178/32  *[LDP/9] 00:33:24, metric 1
                    > via t3-0/1/0.0, Push 100021
10.255.255.180/32  *[LDP/9] 01:00:52, metric 1
                    > via t3-0/1/0.0, Push 100015
mpls.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
100000             *[LDP/9] 03:38:31, metric 1
                    > via so-1/2/0.0, Pop
100000(S=0)        *[LDP/9] 03:38:31, metric 1
                    > via so-1/2/0.0, Pop
100001             *[LDP/9] 03:38:31, metric 1
                    > via so-1/2/0.0, Swap 100000
100002             *[LDP/9] 03:35:06, metric 1
                    > via t3-0/1/0.0, Swap 100015
100007             *[LDP/9] 00:33:24, metric 1
                    > via t3-0/1/0.0, Swap 100020
100008             *[LDP/9] 00:33:24, metric 1
                    > via t3-0/1/0.0, Swap 100021
```

Router PE1 Status

```
user@PE1> show ldp neighbor instance vpn-provider
```

```
Address            Interface          Label space ID         Hold time
192.255.197.18     t3-0/2/1.0         10.255.255.179:0         11

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

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

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

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

user@PE1>  show ldp statistics instance vpn-provider
Message type              Total                   Last 5 seconds
                      Sent      Received          Sent      Received
Hello                 2838          2839           1             2
Initialization           1             1           0             0
Keepalive             1240          1239           0             0
Notification             0             0           0             0
Address                  1             1           0             0
Address withdraw         0             0           0             0
Label mapping           10            10           0             0
```

```
Label request             0             0             0             0
Label withdraw            4             4             0             0
Label release             4             4             0             0
Label abort               0             0             0             0
All UDP                2837          2839             1             2
All TCP                1258          1251             0             0
Event type                             Total        Last 5 seconds
Sessions opened                            1                 0
Sessions closed                            0                 0
Topology changes                          21                 0
No router id                               0                 0
No address                                 0                 0
No interface                               0                 0
No session                                 0                 0
No adjacency                               0                 0
Unknown version                            0                 0
Malformed PDU                              0                 0
Malformed message                          0                 0
Unknown message type                       0                 0
Inappropriate message                      0                 0
Malformed TLV                              0                 0
Bad TLV value                              0                 0
Missing TLV                                0                 0
PDU too large                              0                 0
PDU too small                              0                 0

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

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

Router PE2 Status

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

user@PE2>  show ldp database instance vpn-provider
Input label database, 192.255.197.37:0--10.255.255.180:0
  Label     Prefix
      3     10.255.255.180/32
 100003     10.255.255.177/32
```

```
 100010     10.255.255.178/32
 100009     10.255.255.176/32
 100002     10.255.255.175/32
 100004     10.255.255.179/32
Output label database, 192.255.197.37:0--10.255.255.180:0
  Label     Prefix
 100026     10.255.255.175/32
 100028     10.255.255.179/32
 100027     10.255.255.177/32
 100021     10.255.255.180/32
 100039     10.255.255.178/32
 100037     10.255.255.176/32

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

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

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

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

user@PE2>  show ldp statistics instance vpn-provider
Message type              Total                    Last 5 seconds
                      Sent      Received          Sent      Received
Hello                 2948          2939            1             1
Initialization           1             1            0             0
Keepalive             1285          1285            0             0
Notification             0             0            0             0
Address                  1             1            0             0
Address withdraw         0             0            0             0
Label mapping           10            10            0             0
Label request            0             0            0             0
Label withdraw           4             4            0             0
Label release            4             4            0             0
Label abort              0             0            0             0
All UDP               2947          2939            1             1
All TCP               1297          1299            0             0
```

```
Event type                              Total        Last 5 seconds
Sessions opened                             1                    0
Sessions closed                             0                    0
Topology changes                           33                    0
No router id                                0                    0
No address                                  0                    0
No interface                                0                    0
No session                                  0                    0
No adjacency                                0                    0
Unknown version                             0                    0
Malformed PDU                               0                    0
Malformed message                           0                    0
Unknown message type                        0                    0
Inappropriate message                       0                    0
Malformed TLV                               0                    0
Bad TLV value                               0                    0
Missing TLV                                 0                    0
PDU too large                               0                    0
PDU too small
                            0                    0
user@PE2>  show ldp traffic-statistics instance vpn-provider
FEC                  Type            Packets          Bytes    Shared
 10.255.255.175/32   Transit               0              0    No
 10.255.255.176/32   Transit               0              0    No
 10.255.255.176/32   Ingress               0              0    No
 10.255.255.177/32   Transit            3131         274830    No
 10.255.255.178/32   Transit            1966         178256    No
 10.255.255.178/32   Ingress               0              0    No
 10.255.255.179/32   Transit               1             44    No
 10.255.255.180/32   Transit               0              0    No
 10.255.255.180/32   Ingress            2330         144838    No

user@PE2>  show bgp summary

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

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

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

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

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

```
vpn-provider.inet.3: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
mpls.0: 11 destinations, 11 routes (11 active, 0 holddown, 0 hidden)
vpn-provider.mpls.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
bgp.l3vpn.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.255.255.171:1:10.255.255.175/32
                   *[BGP/170] 03:33:34, MED 1, localpref 100, from 10.255.255.171

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

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

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

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

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

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

user@PE2>  show ldp route instance vpn-provider
Destination         Next-hop intf/lsp               Next-hop address
 10.255.255.175/32  so-0/2/0.0
 10.255.255.176/32  t1-3/0/0.0
 10.255.255.177/32  so-0/2/0.0
 10.255.255.178/32  t1-3/0/0.0
 10.255.255.179/32  so-0/2/0.0
```

```
 10.255.255.180/32  t1-3/0/0.0
 192.255.197.36/30  t1-3/0/0.0
 192.255.197.37/32
 192.255.198.24/30  t1-3/0/0.0
 192.255.198.36/30  t1-3/0/0.0
 224.0.0.5/32

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

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

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

user@PE2>  show bgp summary
Groups: 1 Peers: 1 Down peers: 0
```

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

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

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

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

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

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

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

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

Router CE2 Status

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

user@CE2>  show route
inet.0: 21 destinations, 23 routes (20 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both
0.0.0.0/0          *[Static/5] 07:53:49, metric 0
                      Discard
```

```
10.255.255.175/32  *[OSPF/150] 00:29:56, metric 1, tag 3489661228
                    > via t1-0/0/0.0
10.255.255.176/32  *[OSPF/10] 00:29:56, metric 2
                    > via t3-0/3/3.0
10.255.255.177/32  *[OSPF/150] 00:29:56, metric 1, tag 3489661228
                    > via t1-0/0/0.0
10.255.255.178/32  *[OSPF/10] 00:29:56, metric 67
                    > via t3-0/3/3.0
10.255.255.179/32  *[OSPF/150] 00:29:56, metric 1, tag 3489661228
                    > via t1-0/0/0.0
10.255.255.180/32  *[Direct/0] 07:53:49
                    > via lo0.0
172.16.0.0/12      *[Static/5] 07:53:49
                    > to 192.255.14.254 via fxp0.0
192.255.0.0/18     *[Static/5] 07:53:49
                    > to 192.255.14.254 via fxp0.0
192.255.14.0/24    *[Direct/0] 07:53:49
                    > via fxp0.0
192.255.14.180/32  *[Local/0] 07:53:49
                      Local via fxp0.0
192.255.40.0/22    *[Static/5] 06:07:28
                    > to 192.255.14.254 via fxp0.0
192.255.64.0/18    *[Static/5] 07:49:39
                    > to 192.255.14.254 via fxp0.0
192.255.197.36/30  *[Direct/0] 03:38:03
                    > via t1-0/0/0.0
                    [OSPF/10] 00:29:56, metric 65
                    > via t1-0/0/0.0
192.255.197.38/32  *[Local/0] 07:52:52
                      Local via t1-0/0/0.0
192.255.198.24/30  *[Direct/0] 03:33:17
                    > via t3-0/3/3.0
                    [OSPF/10] 00:29:56, metric 2
                    > via t3-0/3/3.0
192.255.198.26/32  *[Local/0] 07:52:49
                      Local via t3-0/3/3.0
192.255.198.36/30  *[OSPF/10] 00:29:56, metric 67
                    > via t3-0/3/3.0
207.17.136.192/32  *[Static/5] 07:53:49
                    > to 192.255.14.254 via fxp0.0
224.0.0.5/32       *[OSPF/10] 03:38:55, metric 1
                      MultiRecv
inet.3: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.255.255.175/32  *[LDP/9] 03:35:53, metric 1
                    > via t1-0/0/0.0, Push 100026
10.255.255.176/32  *[LDP/9] 00:34:13, metric 1
                    > via t3-0/3/3.0
10.255.255.177/32  *[LDP/9] 03:35:53, metric 1
                    > via t1-0/0/0.0, Push 100027
10.255.255.178/32  *[LDP/9] 00:34:13, metric 1
                    > via t3-0/3/3.0, Push 100014
10.255.255.179/32  *[LDP/9] 03:35:53, metric 1
                    > via t1-0/0/0.0, Push 100028
mpls.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
100002             *[LDP/9] 03:35:53, metric 1
                    > via t1-0/0/0.0, Swap 100026
100003             *[LDP/9] 03:35:53, metric 1
                    > via t1-0/0/0.0, Swap 100027
100004             *[LDP/9] 03:35:53, metric 1
```

```
                          > via t1-0/0/0.0, Swap 100028
                100009    *[LDP/9] 00:34:13, metric 1
                          > via t3-0/3/3.0, Pop
                100009(S=0)  *[LDP/9] 00:34:13, metric 1
                          > via t3-0/3/3.0, Pop
                100010    *[LDP/9] 00:34:13, metric 1
                          > via t3-0/3/3.0, Swap 100014
```

Router PE4 Status

```
user@PE4> show bgp summary

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

user@PE4>  show route protocol bgp
inet.0: 20 destinations, 21 routes (19 active, 0 holddown, 1 hidden)
inet.3: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
vpn-customer.inet.0: 12 destinations, 14 routes (12 active, 0 holddown,
0 hidden)
+ = Active Route, - = Last Active, * = Both
10.49.100.0/24     *[BGP/170] 00:23:27, MED 0, localpref 100, from 10.255.255.177
                      AS path: 100 I
                    > via t1-0/1/1.0, Push 100013, Push 100012(top)
10.49.100.1/32     *[BGP/170] 00:23:27, localpref 100, from 10.255.255.177
                      AS path: 100 I
                    > via t1-0/1/1.0, Push 100013, Push 100012(top)
10.49.101.0/24     *[BGP/170] 00:23:27, MED 0, localpref 100, from 10.255.255.177
                      AS path: 100 I
                    > via t1-0/1/1.0, Push 100013, Push 100012(top)
10.49.200.0/24     *[BGP/170] 03:29:00, MED 0, localpref 100
                      AS path: 100 I
                    > to 192.255.198.22 via t3-0/0/3.0
10.49.200.1/32     *[BGP/170] 03:29:00, localpref 100
                      AS path: 100 I
                    > to 192.255.198.22 via t3-0/0/3.0
10.49.201.0/24     *[BGP/170] 03:29:00, MED 0, localpref 100
                      AS path: 100 I
                    > to 192.255.198.22 via t3-0/0/3.0
10.255.255.181/32  *[BGP/170] 00:23:27, localpref 100, from 10.255.255.177
                      AS path: 100 I
                    > via t1-0/1/1.0, Push 100013, Push 100012(top)
10.255.255.182/32  *[BGP/170] 03:29:00, localpref 100
                      AS path: 100 I
                    > to 192.255.198.22 via t3-0/0/3.0
192.255.14.0/24    *[BGP/170] 03:29:00, localpref 100
                      AS path: 100 I
                    > to 192.255.198.22 via t3-0/0/3.0
                    [BGP/170] 00:23:27, localpref 100, from 10.255.255.177
                      AS path: 100 I
                    > via t1-0/1/1.0, Push 100013, Push 100012(top)
```

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

user@PE4>  show route protocol ldp
inet.0: 20 destinations, 21 routes (19 active, 0 holddown, 1 hidden)
inet.3: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.255.255.175/32  *[LDP/9] 00:29:08, metric 1
                    > via t1-0/1/1.0, Push 100011
10.255.255.176/32  *[LDP/9] 00:29:08, metric 1
                    > via t1-0/1/1.0
10.255.255.177/32  *[LDP/9] 00:29:08, metric 1
                    > via t1-0/1/1.0, Push 100012
10.255.255.179/32  *[LDP/9] 00:29:08, metric 1
                    > via t1-0/1/1.0, Push 100013
10.255.255.180/32  *[LDP/9] 00:29:08, metric 1
                    > via t1-0/1/1.0, Push 100010
vpn-customer.inet.0: 12 destinations, 14 routes (12 active, 0 holddown, 0 hidden)
mpls.0: 13 destinations, 13 routes (13 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
100014             *[LDP/9] 00:29:08, metric 1
                    > via t1-0/1/1.0, Pop
100014(S=0)        *[LDP/9] 00:29:08, metric 1
                    > via t1-0/1/1.0, Pop
100015             *[LDP/9] 00:29:08, metric 1
                    > via t1-0/1/1.0, Swap 100010
100016             *[LDP/9] 00:29:08, metric 1
                    > via t1-0/1/1.0, Swap 100011
100017             *[LDP/9] 00:29:08, metric 1
                    > via t1-0/1/1.0, Swap 100012
```

```
100018             *[LDP/9] 00:29:08, metric 1
                    > via t1-0/1/1.0, Swap 100013
bgp.l3vpn.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
```

Router CE4 Status

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

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

Related Topics For more information about proper configuration of VRF instances, see the *JUNOS VPNs Configuration Guide*. For the proper syntax related to policies, see the *JUNOS Policy Framework Configuration Guide*.

For More Information

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

- *JUNOS VPNs Configuration Guide*
- *JUNOS MPLS Applications Configuration Guide*
- *JUNOS Policy Framework Configuration Guide*
- RFC 3017, *Carrying Label Information in BGP-4*
- RFC 3036, *LDP Specification*

Revision History

13 April 2009—9.5R1 Release. Roy Spencer.

15 January 2009—9.4R1 Release. Roy Spencer.

10 October 2008—9.3R1 Release. Roy Spencer.

10 July 2008—9.2R1 Release. Roy Spencer.

10 April 2008—9.1R1 Release. Roy Spencer.

1 February 2008—9.0R1 Release. Fawn Damitio.

27 March 2007—8.3R1 Release. Fawn Damitio.

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

15 September 2006—8.1R1 Release. Richard Hendricks.

29 June 2006—8.0R1 Release. Richard Hendricks.

27 March 2006—7.6R1 Release. Richard Hendricks.

9 January 2006—7.5R1 Release. Richard Hendricks.

14 September 2005—7.4R1 Release. Richard Hendricks.

13 June 2005—7.3R1 Release. Richard Hendricks.

5 April 2005—7.2R1 Release. Richard Hendricks.

2 February 2005—7.1R1 Release. Richard Hendricks.

6 October 2004—7.0R1 Release. Richard Hendricks.

6 July 2004—6.4R1 Release. Richard Hendricks.

5 April 2004—6.3R1 Release. Richard Hendricks.

22 December 2003—6.2R1 Release. Richard Hendricks.

22 September 2003—6.1R1 Release. Richard Hendricks.

30 June 2003—6.0R1 Release. Richard Hendricks.

2 April 2003—5.7R1 Release. Richard Hendricks.

27 December 2002—5.6R1 Release. Richard Hendricks.

30 September 2002—5.5R1 Release. Richard Hendricks.

19 July 2002—5.4R1 Release. Richard Hendricks.

6 May 2002—Initial document written. Richard Hendricks.

Chapter 4

MPLS LSP Link Protection and Node-Link Protection

This feature guide covers these topics:

Overview

Multiprotocol Label Switching (MPLS) label-switched path (LSP) link protection and node-link protection are facility-based methods used to reduce the amount of time needed to reroute LSP traffic. These protection methods are often compared to fast reroute—the other JUNOS software LSP protection method.

While fast reroute can only protect LSPs on a one-to-one basis, link protection and node-link protection can protect multiple LSPs by using only a single, logical bypass

LSP. Link protection can provide robust backup support for a link, node-link protection can bypass a node or a link, and both types of protection are designed to interoperate with other vendor equipment. Such functionality makes link protection and node-link protection excellent choices for scalability, redundancy, and performance in MPLS-enabled networks.

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

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

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

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

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

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

- Packet Forwarding Engine local repair—This data plane method adds enhanced capabilities to the Packet Forwarding Engine subsystem and reduces the time needed for path switchover. With local repair, the Packet Forwarding Engine can correct a path failure before it receives recomputed paths from the Routing Engine. The Routing Engine pre-computes backup routes for every MPLS path and provides this information to the Packet Forwarding Engine before any failure. Packet Forwarding Engine local repair is enabled by default and requires no additional configuration.
- Fast reroute—The original control plane method for fast reroute of individual LSPs is described as “one-to-one” protection in the IETF Internet draft *Fast Reroute Extensions to RSVP-TE for LSP Tunnels*. JUNOS software calculates LSP detours for LSPs and implements the rerouted paths as needed. You can configure

the command fast-reroute at the [edit protocols mpls *lsp-name*] hierarchy level. For more information about MPLS LSP fast reroute, see the *JUNOS MPLS Applications Configuration Guide*.

- Link protection—Another control plane method discussed in this guide. In general, link protection is useful when you wish to protect LSPs after a supporting link is lost.
- Node-link protection—This is also a control plane method and is discussed in this guide. In general, link protection is useful when you wish to protect LSPs after a supporting node fails.

Link Protection

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

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

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

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

On a per-RSVP neighbor basis, the router keeps track of all the LSP sessions passing through a neighbor as well as the bypass LSP status. For the bypass LSP, the router maintains information about protected neighbors. For regular LSPs, the router monitors all threads containing the LSP. When a regular LSP is lost, the bypass LSP reroutes user traffic by using information about the next hop, egress Explicit Route Object (ERO), interface, and peer address.

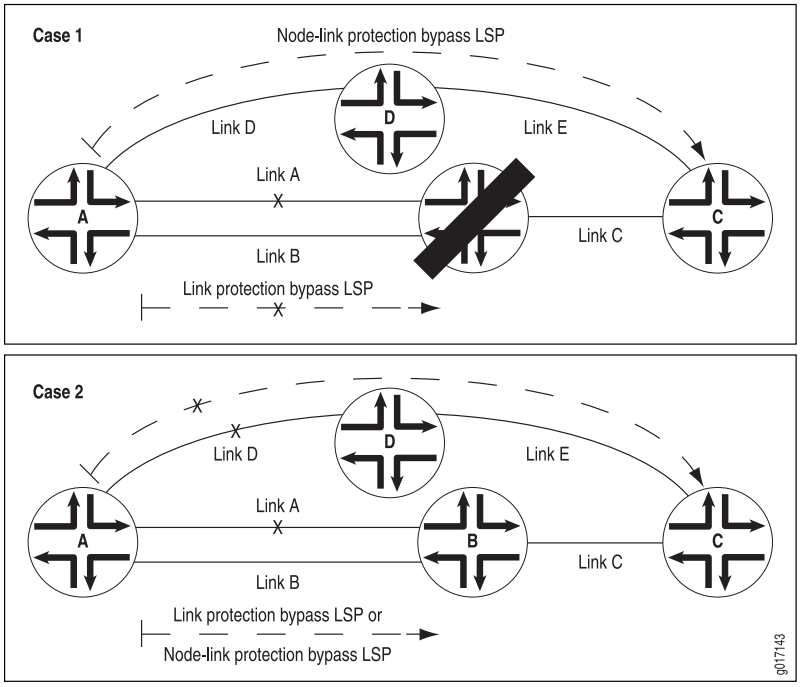
NOTE: Fast reroute, link protection, and node-link protection all rely on Constrained Shortest Path First (CSPF) to select bypass LSPs. The CSPF computation attempts to find an LSP that bypasses an affected node first, but can select an alternate link through the affected node if a node bypass LSP is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

Node-Link Protection

While link protection is useful for selecting an alternate path to the same router when a link fails, node-link protection establishes a bypass LSP through a different router altogether. For Case 1 in Figure 9 on page 102, link protection allows an LSP to switch to link B and immediately bypass failed link A. However, if Router B fails, link B will fail and the link-protected LSP will be lost.

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

Figure 9: Link Protection and Node-Link Protection Comparison



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

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

System Requirements

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

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

Terms and Acronyms

B

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

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

L

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

N

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

Configuring MPLS LSP Link Protection or Node-Link Protection

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

Configuring Link Protection or Node-Link Protection on the LSP

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

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

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

Configuring Link Protection on the RSVP Interfaces Traversed by the LSP

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

```
[edit]
protocols {
   rsvp {
      interface interface-name {
```

```
            link-protection {
               path  ip-address  {
                  (loose | strict);
               }
            }
         }
      }
   }
```

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

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

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

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

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

Link protection priority enables you to provide preferred combinations of priority and class in a traffic engineering class matrix. When selecting a bypass LSP, the routing platform selects the bypass containing the lowest priority. To configure link protection priority for all bypass LSP paths, include the priority statement at the [edit protocols rsvp interface *interface-name* link-protection] hierarchy level. To configure link protection priority for a manually specified bypass LSP path, include the priority statement at the [edit protocols rsvp interface *interface-name* link-protection bypass *bypass-name*] hierarchy level.

```
[edit protocols]
mpls {
```

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

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

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

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

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

Verifying MPLS LSP Link Protection and Node Link Protection

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

Action

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

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

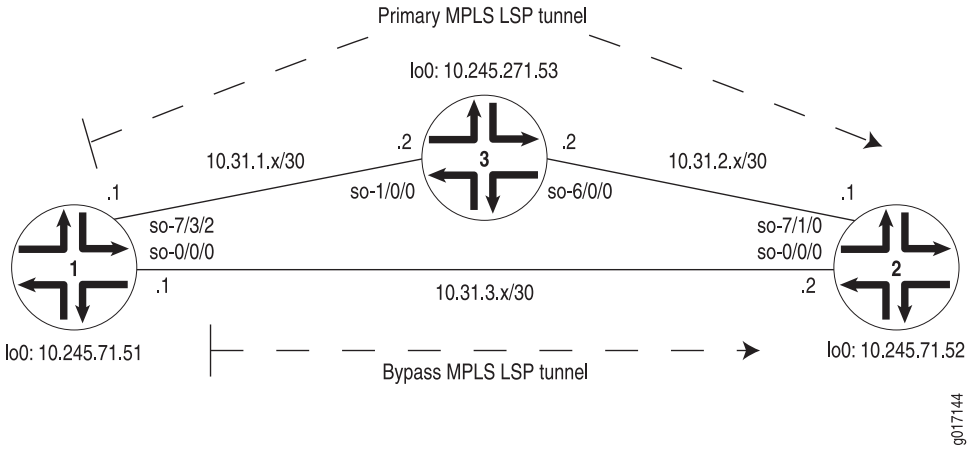
MPLS LSP Link Protection or Node-Link Protection Configuration Examples

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

- Example: Configuring MPLS LSP Link Protection on page 107
- Example: Node-Link Protection Configuration on page 127

Example: Configuring MPLS LSP Link Protection

Figure 10: MPLS LSP Link Protection Topology Diagram



In Figure 10 on page 107, a primary MPLS LSP is established from Router 1 through Router 3 to destination Router 2. To implement link protection, include the

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

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

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

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

Router 1

```
[edit]
interfaces {
    so-0/0/0 {
        unit 0 {
            family inet {
                address 10.31.3.1/30;
            }
            family iso;
            family mpls;
        }
    }
    so-7/3/2 {
        unit 0 {
            family inet {
                address 10.31.1.1/30;
            }
            family iso;
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.245.71.51/32;
            }
            family iso {
            }
        }
    }
    protocols {
        rsvp {
            interface so-7/3/2.0 {
```

```
            link-protection; # Enable link protection on the interface carrying the main
              LSP.
          }
          interface so-0/0/0.0 {
          }
          mpls {
            label-switched-path Protected_LSP {
              to 10.245.71.52;
              install 10.31.5.1/32 active; # This route is used for testing the LSPs.
              link-protection; # Enable link protection on the protected LSP.
              primary path1;
            }
            path path1 {
              10.31.1.2 strict;
            }
            interface so-0/0/0.0;
            interface so-7/3/2.0;
          }
          isis {
            level 2 wide-metrics-only;
            interface so-0/0/0.0 {
              level 1 disable;
              level 2 {
                metric 100;
                te-metric 100;
              }
            }
            interface so-7/3/2.0 {
              level 1 disable;
              level 2 {
                metric 10;
                te-metric 10;
              }
            }
            interface lo0.0 {
              passive;
            }
          }
        }
        routing-options {
          forwarding-table { # Apply this policy to the forwarding table only
            export pplb; # if Packet Forwarding Engine local repair is needed.
          }
        }
        policy-options {
          policy-statement pplb {
            then {# Configure this policy only if
              load-balance per-packet; # Packet Forwarding Engine local repair is needed.
            }
          }
        }
      }
    }
```

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

Router 2

```
[edit]
interfaces {
    so-0/0/0 {
        unit 0 {
            family inet {
                address 10.31.3.2/30;
            }
            family iso;
            family mpls;
        }
    }
    so-7/1/0 {
        unit 0 {
            family inet {
                address 10.31.2.1/30;
            }
            family iso;
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.245.71.53/32;
            }
            family iso {
            }
        }
    }
    protocols {
        rsvp {
            interface so-0/0/0.0;
            interface so-7/1/0.0;
        }
        mpls {
            interface so-0/0/0.0;
            interface so-7/1/0.0;
        }
        isis {
            level 2 wide-metrics-only;
            interface so-0/0/0.0 {
                level 1 disable;
                level 2 {
                    metric 100;
                    te-metric 100;
                }
            }
            interface so-7/1/0.0 {
                level 1 disable;
                level 2 {
                    metric 10;
                    te-metric 10;
                }
            }
            interface lo0.0 {
                passive;
```

```
        }
      }
    }
  }
```

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

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

Router 3

```
[edit]
interfaces {
   so-1/0/0 {
      unit 0 {
         family inet {
            address 10.31.1.2/30;
         }
         family iso;
         family mpls;
      }
   }
   so-6/0/0 {
      unit 0 {
         family inet {
            address 10.31.2.2/30;
         }
         family iso;
         family mpls;
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.245.271.53/32;
         }
         family iso {
         }
      }
   }
   protocols {
      rsvp {
         interface so-1/0/0.0;
         interface so-6/0/0.0 { # Primary interface going to Router 2.
            link-protection;
         }
      }
      mpls {
         interface so-1/0/0.0;
         interface so-6/0/0.0;
      }
      isis {
```

```
            level 2 wide-metrics-only;
            interface so-1/0/0.0 {
               level 1 disable;
               level 2 {
                  metric 10;
                  te-metric 10;
               }
            }
            interface so-6/0/0.0 {
               level 1 disable;
               level 2 {
                  metric 10;
                  te-metric 10;
               }
            }
            interface lo0.0 {
               passive;
            }
         }
      }
      policy-options {
         policy-statement pplb {
            then {
               load-balance per-packet; # If Packet Forwarding Engine local repair is needed.
            }
         }
      }
      routing-options {
         forwarding-table {
            export pplb; # If Packet Forwarding Engine local repair is needed.
         }
      }
   }
```

Verifying Your Work

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

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

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

Case 1: Normal Operation

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

Router 1

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

This is the main LSP.

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

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

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

user@Router1>  show rsvp session detail
Ingress RSVP: 2 sessions
10.245.71.52
   From: 10.245.71.51, LSPstate: Up, ActiveRoute: 1
  LSPname: Protected_LSP
```

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

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

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

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

user@Router1>  show rsvp interface detail
RSVP interface: 2 active
fxp0.0  Index 1, State Dis/Up
  NoAuthentication, NoAggregate, NoReliable, NoLinkProtection
  HelloInterval 9(second)
  Address 192.168.71.52
  PacketType              Total                 Last 5 seconds
                   Sent      Received        Sent      Received
  Path                0             0           0             0
  PathErr             0             0           0             0
  PathTear            0             0           0             0
  Resv                0             0           0             0
  ResvErr             0             0           0             0
  ResvTear            0             0           0             0
  Hello               0             0           0             0
  Ack                 0             0           0             0
  Srefresh            0             0           0             0
  EndtoEnd RSVP       0             0           0             0
so-0/0/0.0  Index 8, State Ena/Up
  NoAuthentication, NoAggregate, NoReliable, NoLinkProtection
  HelloInterval 20(second)
  Address 10.31.3.1, 10.245.71.51
  ActiveResv 2, PreemptionCnt 0, Update threshold 10%
  Subscription 100%, StaticBW 622.08Mbps, AvailableBW 622.08Mbps
  PacketType              Total                 Last 5 seconds
                   Sent      Received        Sent      Received
  Path              441             0           0             0
  PathErr             0             0           0             0
  PathTear            3             0           0             0
```

```
  Resv                0          431          0            0
  ResvErr             0            0          0            0
  ResvTear            0            0          0            0
  Hello             489          487          0            0
  Ack                 0            0          0            0
  Srefresh            0            0          0            0
  EndtoEnd RSVP       0            0          0            0
so-7/3/2.0  Index 11, State Ena/Up
  NoAuthentication, NoAggregate, NoReliable,  LinkProtection
```

Link protection is enabled.

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

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

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

```
    Bypass explicit route: 10.31.3.2 10.31.2.2

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

user@Router1>show route 10.31.5.1 extensive

inet.0: 24 destinations, 24 routes (22 active, 0 holddown, 2 hidden)
10.31.5.1/32 (1 entry, 1 announced)
```

```
        State: <FlashAll>
TSI:
KRT in-kernel 10.31.5.1/32 -> {0.0.0.0, 0.0.0.0}
        *RSVP   Preference: 7
                 Next hop: via so-7/3/2.0 weight 1, selected
```

This is the main LSP.

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

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

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

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

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

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

```
  Next-hop type: Push 100007, Push 100000(top)[0]
  Next-hop interface: so-0/0/0.0        Weight: 20001
# The Weight value for the backup starts at  20000 .
```

Router 3

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

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

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

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

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

The ingress bypass session to Router 2 from Router 1.

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

Egress RSVP: 1 sessions

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

The bypass from Router 1 to Router 3, arriving by way of Router 2.

```
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: -
  Resv style: 1 SE, Label in: 3, Label out: -
  Time left:   54,  Since: Thu Aug  8 12:26:48 2002
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 52 protocol 0
  PATH rcvfrom: 10.31.2.1  (so-6/0/0.0) 5 pkts
  PATH sentto: localclient
  RESV rcvfrom: localclient
  Record route: 10.31.3.1  10.31.2.1  <self>
Total 1 displayed, Up 1, Down 0
Transit RSVP: 1 sessions
10.245.71.52
  From: 10.245.71.51, LSPstate: Up, ActiveRoute: 1
  LSPname: Protected_LSP
```

This is the main LSP.

```
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 0
  Resv style: 1 SE, Label in: 100000, Label out: 0
  Time left:   41,  Since: Thu Aug  8 12:26:39 2002
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 2 receiver 33 protocol 0
  Link protection desired
  Type: Link protected LSP
```

```
  PATH rcvfrom: 10.31.1.1  (so-1/0/0.0) 9 pkts
  PATH sentto: 10.31.2.1 (so-6/0/0.0) 11 pkts
  RESV rcvfrom: 10.31.2.1  (so-6/0/0.0) 10 pkts
  Explct route: 10.31.2.1
  Record route: 10.31.1.1  <self>  10.31.2.1
Total 1 displayed, Up 1, Down 0

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

Backup routes = 1

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

user@Router3>  show rsvp interface detail
RSVP interface: 2 active
fxp0.0  Index 1, State Dis/Up
  NoAuthentication, NoAggregate, NoReliable, NoLinkProtection
  HelloInterval 9(second)
  Address 192.168.6.64
  PacketType              Total                 Last 5 seconds
                    Sent      Received        Sent      Received
  Path                 0             0           0             0
  PathErr              0             0           0             0
  PathTear             0             0           0             0
  Resv                 0             0           0             0
  ResvErr              0             0           0             0
  ResvTear             0             0           0             0
  Hello                0             0           0             0
  Ack                  0             0           0             0
  Srefresh             0             0           0             0
  EndtoEnd RSVP        0             0           0             0
so-1/0/0.0  Index 6, State Ena/Up
  NoAuthentication, NoAggregate, NoReliable, NoLinkProtection
  HelloInterval 3(second)
  Address 10.31.1.2, 10.245.271.53
  ActiveResv 1, PreemptionCnt 0, Update threshold 10%
  Subscription 100%, StaticBW 2.48832Gbps, AvailableBW 2.48832Gbps
  PacketType              Total                 Last 5 seconds
                    Sent      Received        Sent      Received
  Path                 5            14           0             1
  PathErr              0             0           0             0
  PathTear             0             0           0             0
```

```
  Resv                8             9           0             0
  ResvErr             0             0           0             0
  ResvTear            0             0           0             0
  Hello              23            25           2             2
  Ack                 0             0           0             0
  Srefresh            0             0           0             0
  EndtoEnd RSVP       0             0           0             0
so-6/0/0.0  Index 9, State Ena/Up
  NoAuthentication, NoAggregate, NoReliable,  LinkProtection
```

Link protection is enabled.

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

user@Router3>  show route forwarding-table family mpls
Routing table: ccc
MPLS:
Destination        Type RtRef Next hop          Type Index NhRef Netif
default            perm     0                   dscd     1     1
0                  user     0                   recv     3     3
1                  user     0                   recv     3     3
2                  user     0                   recv     3     3
100000             user     0                   ulst    77     1
```

This is the main LSP.

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

The bypass LSP from Router 1 to Router 2.

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

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

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

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

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

Router 1

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

The main LSP is up.

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

The bypass LSP from Router 3 to Router 2.

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

user@Router1>  show rsvp session detail
Ingress RSVP: 3 sessions
10.245.71.52
  From: 10.31.3.1, LSPstate: Up, ActiveRoute: 0
  LSPname: Protected_LSP
```

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

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

The original LSP is now down.

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

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

The bypass LSP from Router 1 to Router 2.

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

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

Transit RSVP: 1 sessions

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

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

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

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

The neighbor is down.

```
  Last changed time: 25, Idle: 25 sec, Up cnt: 2, Down cnt: 2
  Message received: 397
  Hello: sent 900, received: 890, interval: 3 sec
  Remote instance: 0x0, Local instance: 0x41b41b17
  Refresh reduction:  not operational
  Link protection:  enabled
    LSP name: Bypass->10.31.1.2
    Bypass LSP: operational,  Backup routes: 2,  Backup LSPs: 1
    Bypass explicit route: 10.31.3.2 10.31.2.2
Address: 10.31.3.2   via: so-0/0/0.0    status: Up
  Last changed time: 25:40, Idle: 5 sec, Up cnt: 4, Down cnt: 3
  Message received: 502
  Hello: sent 558, received: 538, interval: 20 sec
  Remote instance: 0x194fa7af, Local instance: 0x507b7c2a
  Refresh reduction:  not operational
  Link protection:  disabled
    Bypass LSP: does not exist,  Backup routes: 0,  Backup LSPs: 0

user@Router1>    show route 10.31.5.1
inet.0: 23 destinations, 23 routes (22 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both
10.31.5.1/32       *[RSVP/7] 00:03:04, metric 20
                    > via so-0/0/0.0, label-switched-path Bypass->10.31.1.2
# The route can be reached by way of the backup LSP.

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

so-0/0/0.0
# Double-stacked labels appear on the backup LSP from Router 1 to 2 to 3.
```

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

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

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

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

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

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

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

Router 3

```
user@Router3> show rsvp session
Ingress RSVP: 2 sessions
To              From            State Rt Style Labelin Labelout LSPname
```

```
10.245.71.52    10.245.271.53   Up     1  1 SE       -   100005
Bypass->10.31.2.1
10.245.71.52    10.31.1.2       Up     0  1 SE       -        0 Protected_LSP
```

The backup is signaled from Router 3 to Router 2.

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

The main LSP is down.

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

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

10.245.71.52
  From: 10.245.271.53, LSPstate: Up, ActiveRoute: 1
  LSPname: Bypass->10.31.2.1
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 100005
  Resv style: 1 SE, Label in: -, Label out: 100005
  Time left:    -,  Since: Thu Aug  8 12:31:09 2002
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 17 protocol 0
  Type: Bypass LSP
  PATH rcvfrom: localclient
  PATH sentto: 10.31.1.1 (so-1/0/0.0) 6 pkts
  RESV rcvfrom: 10.31.1.1  (so-1/0/0.0) 6 pkts
  Explct route: 10.31.1.1 10.31.3.2
  Record route: <self>  10.31.1.1  10.31.3.2

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

This is the backup LSP.

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

```
Total 2 displayed, Up 2, Down 0

Egress RSVP: 1 sessions

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

Transit RSVP: 1 sessions

10.245.71.52
  From: 10.245.71.51, LSPstate: Dn, ActiveRoute: 0
  LSPname: Protected_LSP
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: -
  Resv style: 0 -, Label in: 100000, Label out: -
  Time left:   53,  Since: Thu Aug  8 12:26:39 2002
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 2 receiver 33 protocol 0
  Link protection desired
  Type: Link protected LSP
  PATH rcvfrom: 10.31.1.1  (so-1/0/0.0) 30 pkts
  PATH sentto: [no route]
  Explct route: 10.31.2.1
  Record route: 10.31.1.1  <self>  ...incomplete
Total 1 displayed, Up 0, Down 1

user@Router3>  show route forwarding-table family mpls
Routing table: ccc
MPLS:
Destination        Type RtRef Next hop          Type Index NhRef Netif
default            perm     0                   dscd     1     1
0                  user     0                   recv     3     3
1                  user     0                   recv     3     3
2                  user     0                   recv     3     3
100000             user     0                   Swap  100005[0]  so-1/0/0.0
# This shows label swapping for the main LSP traveling over the backup LSP through
 Router 2.
```

Router 1

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

The main LSP is not affected.

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

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

Transit RSVP: 1 sessions

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

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

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

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

Swapping labels on the bypass LSP.

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

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

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

user@Router1>  show route 10.31.5.1 extensive

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

Only one entry is shown.

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

user@Router1>  show rsvp neighbor detail
RSVP neighbor: 2 learned
Address: 10.31.1.2   via: so-7/3/2.0    status: Up
  Last changed time: 2:58, Idle: 0 sec, Up cnt: 3, Down cnt: 2
  Message received: 433
  Hello: sent 965, received: 950, interval: 3 sec
  Remote instance: 0x74be7c42, Local instance: 0x41b41b17
  Refresh reduction:  not operational
  Link protection:  disabled
    Bypass LSP: does not exist,  Backup routes: 0,  Backup LSPs: 0
Address: 10.31.3.2   via: so-0/0/0.0    status: Up
  Last changed time: 30:54, Idle: 0 sec, Up cnt: 4, Down cnt: 3
  Message received: 533
  Hello: sent 593, received: 573, interval: 20 sec
  Remote instance: 0x194fa7af, Local instance: 0x507b7c2a
  Refresh reduction:  not operational
  Link protection:  disabled
    Bypass LSP: does not exist,  Backup routes: 0,  Backup LSPs: 0

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

Example: Node-Link Protection Configuration

Figure 11: Node-Link Protection Topology Diagram

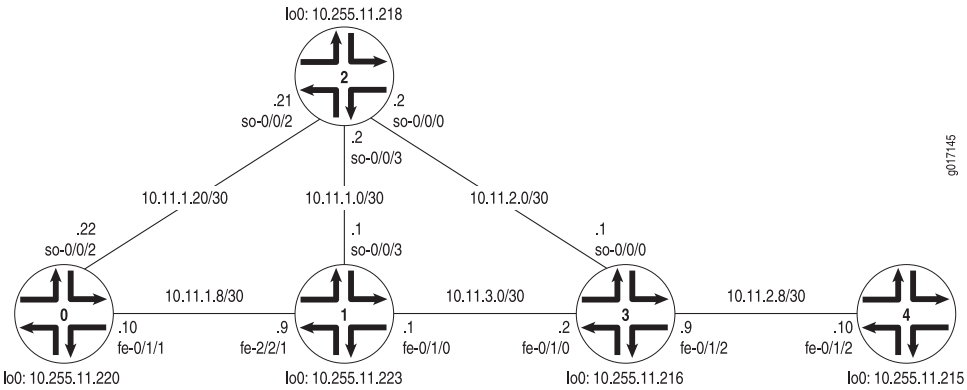


Figure 11 on page 127 shows an example of how you can implement node-link protection. An LSP is initiated on Router 0 with a strict path travelling through Router 1, Router 2, Router 3, and Router 4. You configure node-link protection within the LSP and link protection on all RSVP interfaces in the path.

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

Router 0

```
[edit]
interfaces {
   fe-0/1/1 {
      unit 0 {
         family inet {
            address 10.11.1.10/30;
         }
         family mpls;
      }
   }
   so-0/0/2 {
      unit 0 {
         family inet {
            address 10.11.1.22/30;
         }
         family mpls;
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.255.11.220/32;
         }
      }
   }
```

```
}
protocols {
   ospf {
      area 0.0.0.0 {
         interface lo0.0 {
            passive;
         }
         interface fe-0/1/1.0;
         interface so-0/0/2.0;
      }
      traffic-engineering;
   }
   mpls {
      statistics {
         file mplsStat.log size 1m;
         interval 30;
         display-id;
      }
      traffic-engineering bgp-igp-both-ribs;
      traceoptions {
         file mpls.log size 5m world-readable;
         flag cspf;
         flag cspf-link;
         flag cspf-node;
         flag state;
         flag error;
      }
      explicit-null;
      label-switched-path test_r0_r4 {
         from 10.255.11.220;
         to 10.255.11.215;
         node-link-protection; # Apply node-link protection to the LSP.
         primary pathP;
      }
      path pathP {# Define the LSP path across routers 1, 2, 3, and 4.
         10.11.1.9 strict;
         10.11.1.2 strict;
         10.11.2.1 strict;
         10.11.2.10 strict;
      }
      interface fe-0/1/1.0;
      interface so-0/0/2.0;
   }
   rsvp {
      traceoptions {
         file rsvp.log size 3m files 12 world-readable;
         flag event;
         flag state;
         flag error;
         flag packets detail;
      }
      interface fe-0/1/1.0 {
         link-protection; # Apply link protection to RSVP interfaces in the LSP path.
      }
      interface so-0/0/2.0;
   }
```

```
}
```

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

Router 1

```
[edit]
interfaces {
   fe-0/1/0 {
      unit 0 {
         family inet {
            address 10.11.3.1/30;
         }
         family mpls;
      }
   }
   fe-2/2/1 {
      unit 0 {
         family inet {
            address 10.11.1.9/30;
         }
         family mpls;
      }
   }
   so-0/0/3 {
      unit 0 {
         family inet {
            address 10.11.1.1/30;
         }
         family mpls;
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.255.11.223/32;
         }
      }
   }
}
protocols {
   ospf {
      area 0.0.0.0 {
         interface lo0.0 {
            passive;
         }
         interface fe-0/1/0.0;
         interface fe-2/2/1.0;
         interface so-0/0/3.0;
      }
      traffic-engineering;
   }
   mpls {
      traffic-engineering bgp-igp-both-ribs;
      explicit-null;
      interface fe-0/1/0.0;
      interface fe-2/2/1.0;
```

```
    interface so-0/0/3.0;
  }
  rsvp {
    interface fe-0/1/0.0;
    interface fe-2/2/1.0;
    interface so-0/0/3.0 {
      link-protection; # Apply link protection on all RSVP interfaces in the LSP path.
    }
  }
}
```

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

Router 2

```
[edit]
interfaces {
  so-0/0/0 {
    unit 0 {
      family inet {
        address 10.11.2.2/30;
      }
      family mpls;
    }
  }
  so-0/0/2 {
    unit 0 {
      family inet {
        address 10.11.1.21/30;
      }
      family mpls;
    }
  }
  so-0/0/3 {
    unit 0 {
      family inet {
        address 10.11.1.2/30;
      }
      family mpls;
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.255.11.218/32;
      }
    }
  }
}
protocols {
  ospf {
    area 0.0.0.0 {
      interface lo0.0 {
        passive;
      }
      interface so-0/0/0.0;
      interface so-0/0/2.0;
```

```
        interface so-0/0/3.0;
      }
      traffic-engineering;
    }
    mpls {
      traffic-engineering bgp-igp-both-ribs;
      explicit-null;
      interface so-0/0/0.0;
      interface so-0/0/2.0;
      interface so-0/0/3.0;
    }
    rsvp {
      interface so-0/0/0.0 {
        link-protection; # Apply link protection to RSVP interfaces in the LSP path.
      }
      interface so-0/0/2.0;
      interface so-0/0/3.0;
    }
  }
```

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

Router 3

```
[edit]
interfaces {
  fe-0/1/0 {
    unit 0 {
      family inet {
        address 10.11.3.2/30;
      }
      family mpls;
    }
  }
  fe-0/1/2 {
    unit 0 {
      family inet {
        address 10.11.2.9/30;
      }
      family mpls;
    }
  }
  so-0/0/0 {
    unit 0 {
      family inet {
        address 10.11.2.1/30;
      }
      family mpls;
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.255.11.216/32;
      }
    }
  }
```

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

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

Router 4

```
[edit]
interfaces {
   fe-0/1/2 {
      unit 0 {
         family inet {
            address 10.11.2.10/30;
         }
         family mpls;
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.255.11.215/32;
         }
      }
   }
}
protocols {
   ospf {
      area 0.0.0.0 {
         interface lo0.0 {
            passive;
```

```
        }
        interface fe-0/1/2.0;
      }
      traffic-engineering;
    }
    mpls {
      traffic-engineering bgp-igp-both-ribs;
      explicit-null;
      interface fe-0/1/2.0;
    }
    rsvp {
      interface fe-0/1/2.0;
    }
  }
```

Verifying Your Work

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

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

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

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

10.255.11.218
  From: 10.255.11.220, LSPstate: Up, ActiveRoute: 0
  LSPname: Bypass->10.11.1.9->10.11.1.2 # 2 next hops indicate node-link
protection.
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 0
  Resv style: 1 SE, Label in: -, Label out: 0
```

```
  Time left:    -, Since: Thu May  8 13:36:58 2003
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 56521 protocol 0
  Type: Bypass LSP
    Number of data route tunnel through: 4
    Number of RSVP session tunnel through: 0
  PATH rcvfrom: localclient
  PATH sentto: 10.11.1.21 (so-0/0/2.0) 1 pkts
  RESV rcvfrom: 10.11.1.21 (so-0/0/2.0) 1 pkts
  Explct route: 10.11.1.21
  Record route: <self> 10.11.1.21
Total 2 displayed, Up 2, Down 0

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

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

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

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

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

   65 May  8 13:37:28 CSPF: computation result ignored
   64 May  8 13:37:07 Record Route:  10.11.1.9(flag=9 Label=100128)
10.11.1.2(flag=1 Label=100080) 10.11.2.1(Label=100080) 10.11.2.10(Label=0)
   63 May  8 13:37:01 Record Route:  10.11.1.9(flag=9 Label=100128)
10.11.1.2(Label=100080) 10.11.2.1(Label=100080) 10.11.2.10(Label=0)
   62 May  8 13:37:01 Link-protection Up
   61 May  8 13:36:58 Selected as active path
   60 May  8 13:36:58 Record Route:  10.11.1.9(Label=100128)
10.11.1.2(Label=100080) 10.11.2.1(Label=100080) 10.11.2.10(Label=0)
   59 May  8 13:36:58 Up
   58 May  8 13:36:58 Originate Call
```

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

   38 May  8 13:27:41 CSPF failed: no route toward 10.11.2.10[2 times]
   37 May  8 13:27:39 CSPF: link down/deleted
10.11.2.9(eagle.00/10.255.11.216)->0.0.0.0(eagle.04/0.0.0.0)
   36 May  8 13:27:39 Link-protection Down
   35 May  8 13:27:39 Record Route:  10.11.1.9(Label=100064)
10.11.1.2(Label=100064) 10.11.2.1(Label=100064) 10.11.2.10(Label=0)
   34 May  8 13:27:39 CSPF failed: no route toward 10.11.2.10
   33 May  8 13:27:39 CSPF: link down/deleted
0.0.0.0(eagle.04/0.0.0.0)->0.0.0.0(papst.00/10.255.11.215)
   32 May  8 13:27:12 CSPF: computation result ignored[16 times]
   31 May  8 13:19:35 Record Route:  10.11.1.9(flag=9 Label=100064)
10.11.1.2(flag=9 Label=100048) 10.11.2.1(flag=1 Label=100048) 10.11.2.10(Label=0)

   30 May  8 13:19:22 Link-protection Up
   29 May  8 13:19:22 Record Route:  10.11.1.9(flag=9 Label=100064)
10.11.1.2(flag=9 Label=100048) 10.11.2.1(Label=100048) 10.11.2.10(Label=0)
   28 May  8 13:19:22 Up
   27 May  8 13:19:22 Link-protection Down
   26 May  8 13:19:22 CSPF: computation result accepted
   25 May  8 13:19:16 Link-protection Up
   24 May  8 13:19:16 Link-protection Down
   23 May  8 13:18:54 CSPF failed: no route toward 10.11.2.1
   22 May  8 13:18:54 CSPF: link down/deleted
0.0.0.0(eagle.04/0.0.0.0)->0.0.0.0(papst.00/10.255.11.215)
   21 May  8 13:18:53 CSPF failed: no route toward 10.11.2.1[2 times]
   20 May  8 13:18:46 CSPF: link down/deleted
10.11.2.9(eagle.00/10.255.11.216)->0.0.0.0(eagle.04/0.0.0.0)
   19 May  8 13:18:35 Record Route:  10.11.1.9(flag=9 Label=100032)
10.11.1.2(flag=9 Label=100032) 10.11.2.1(Label=100016) 10.11.2.10(Label=0)
   18 May  8 13:18:35 Record Route:  10.11.1.9(flag=9 Label=100032)
10.11.1.2(Label=100032) 10.11.2.1(Label=100016) 10.11.2.10(Label=0)
  Created: Thu May  8 13:13:28 2003
Total 1 displayed, Up 1, Down 0
Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

user@router0>  show rsvp neighbor detail
```

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

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

user@router0>  show route 10.255.11.215 detail
inet.0: 33 destinations, 34 routes (31 active, 0 holddown, 2 hidden)
10.255.11.215/32 (2 entries, 1 announced)
        State: <FlashAll>
        *RSVP   Preference: 7
                Next hop: 10.11.1.9 via fe-0/1/1.0 weight 1, selected
                Label-switched-path test_r0_r4
                Label operation: Push 100128, selfID=RSVP-7
                 Next hop: via so-0/0/2.0 weight 20001
                 Label-switched-path Bypass->10.11.1.9->10.11.1.2
                Label operation: Push 100080, selfID=RSVP-7, parentID=RSVP-8
                State: <Active Int>
                Local AS:    69
                Age: 13:14      Metric: 40
                Task: RSVP
                Announcement bits (2): 0-KRT 3-Resolve inet.0
                AS path: I
         IS-IS  Preference: 18
                Level: 2
                Next hop: 10.11.1.9 via fe-0/1/1.0, selected
                State: <Int>
                Inactive reason: Route Preference
                Local AS:    69
                Age: 13:20      Metric: 40
                Task: IS-IS
                AS path: I
inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
10.255.11.215/32 (1 entry, 1 announced)
        State: <FlashAll>
        *RSVP   Preference: 7
                Next hop: 10.11.1.9 via fe-0/1/1.0 weight 1, selected
                Label-switched-path test_r0_r4
                Label operation: Push 100128, selfID=RSVP-7
                Next hop: via so-0/0/2.0 weight 20001
                Label-switched-path Bypass->10.11.1.9->10.11.1.2
                Label operation: Push 100080, selfID=RSVP-7, parentID=RSVP-8
                State: <Active Int>
                Local AS:    69
                Age: 13:14      Metric: 40
                Task: RSVP
                Announcement bits (1): 1-Resolve inet.0
                AS path: I
```

For More Information

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

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

Revision History

13 April 2009—9.5R1 Release. Roy Spencer.

15 January 2009—9.4R1 Release. Roy Spencer.

10 October 2008—9.3R1 Release. Roy Spencer.

10 July 2008—9.2R1 Release. Roy Spencer.

10 April 2008—9.1R1 Release. Roy Spencer.

29 June 2007—8.4R1 Release. Fawn Damitio.

27 March 2007—8.31 Release. Fawn Damitio.

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

15 September 2006—8.1R1 Release. Richard Hendricks.

29 June 2006—8.0R1 Release. Richard Hendricks.

27 March 2006—7.6R1 Release. Richard Hendricks.

9 January 2006—7.5R1 Release. Richard Hendricks.

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

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

5 April 2005—7.2R1 Release. Richard Hendricks.

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

6 October 2004—7.0R1 Release. Richard Hendricks.

6 July 2004—6.4R1 Release. Richard Hendricks.

5 April 2004—6.3R1 Release. Richard Hendricks.

22 December 2003—6.2R1 Release. Richard Hendricks.

22 September 2003—6.1R1 Release. Richard Hendricks.

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

2 April 2003—5.7R1 Release. Richard Hendricks.

27 December 2002—5.6R1 Release. Richard Hendricks.

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

19 July 2002—5.4R1 Release. Richard Hendricks.

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

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

18 March 2002—Initial document written. Greg Ketell.

Chapter 5

RSVP LSP Tunnels

This feature guide covers these topics:

Overview

A Resource Reservation Protocol (RSVP) label-switched path (LSP) tunnel allows you to send RSVP LSPs inside other RSVP LSPs. This allows a network administrator to provide traffic engineering from one end of the network to the other. A useful application for this feature is to connect customer edge (CE) routers with provider edge (PE) routers by using an RSVP LSP, and then tunnel this edge LSP inside a second RSVP LSP traveling across the network core.

This document assumes you have a general understanding of Multiprotocol Label Switching (MPLS) and label switching concepts. For more information about MPLS, see the *JUNOS MPLS Applications Configuration Guide*.

An RSVP LSP tunnel adds the concept of a forwarding adjacency, similar to the one used for generalized Multiprotocol Label Switching (GMPLS). (For more information about GMPLS, see “GMPLS” on page 3.)

The forwarding adjacency creates a tunneled path for sending data between peer devices in an RSVP LSP network. Once a forwarding adjacency LSP (FA-LSP) has been established, other LSPs can be sent over the FA-LSP by using Constrained Shortest Path First (CSPF), Link Management Protocol (LMP), Open Shortest Path First (OSPF), and RSVP.

To enable an RSVP LSP tunnel, the JUNOS software uses the following mechanisms:

- LMP—Originally designed for GMPLS, LMP establishes forwarding adjacencies between RSVP LSP tunnel peers, and maintains and allocates resources for traffic engineering links (TE links).
- OSPF extensions—OSPF was designed to route packets to physical and logical interfaces related to a Physical Interface Card (PIC). This protocol has been extended to route packets to virtual peer interfaces defined in an LMP configuration.
- RSVP-TE extensions—RSVP-TE was designed to signal the setup of packet LSPs to physical interfaces. The protocol has been extended to request path setup for packet LSPs travelling to virtual peer interfaces defined in an LMP configuration.

The following limitations exist for LSP hierarchies:

- Circuit cross-connect (CCC)-based LSPs are not supported.
- Graceful restart is not supported.
- Link protection is not available for FA-LSPs or at the egress point of the forwarding adjacency.
- Point-to-multipoint LSPs are not supported across FA-LSPs.

System Requirements

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

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

Terms and Acronyms

F

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

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

L

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

T

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

RSVP LSP Tunneling Operation

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

1. LMP notifies RSVP and OSPF of the control peer, the control adjacency, and resources for the TE link.
2. GMPLS extracts the LSP attributes from the configuration and requests RSVP to signal one or more specific paths, specified by the TE link addresses.
3. RSVP determines the local TE link, corresponding control adjacency and active control channel, and transmission parameters (such as IP destination). It requests that LMP allocate a resource from the TE link with the specified attributes. If LMP successfully finds a resource matching the attributes, label allocation succeeds. RSVP sends a PathMsg hop-by-hop until it reaches the target router.
4. The target router, on receiving the RSVP PathMsg, requests that LMP allocate a resource based on the signaled parameters. If label allocation succeeds, it sends back a ResvMsg.
5. If the signaling is successful, an RSVP LSP tunnel is provisioned.

Configuring an RSVP LSP Tunnel

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

- Configuring Link Management Protocol Traffic Engineering Links on page 142
- Configuring Link Management Protocol Peers on page 142
- Configuring Peer Interfaces in OSPF and RSVP on page 143
- Establishing FA-LSP Path Information on page 143
- Defining Label-Switched Paths for the FA-LSP on page 144
- Creating End-to-End LSPs to Traverse the FA-LSP on page 144
- Option: Tearing Down RSVP LSPs Gracefully on page 144

Configuring Link Management Protocol Traffic Engineering Links

To begin your RSVP LSP tunnel configuration, configure LMP TE links on both the ingress and egress routing platforms. Because TE links define a unidirectional connection between peer devices, you must configure TE links in both directions between peers to enable the bidirectional transport of packets.

To configure TE links in LMP, include the te-link *te-link-name* statement at the [edit protocols link-management] hierarchy level. Define the TE link options shown below, especially the label-switched path to be used as the FA-LSP to reach the peer. Optionally, you can specify the traffic engineering metric for the TE link. By default, the traffic engineering metric is derived from the CSPF computation.

```
[edit]
protocols {
   link-management {
      te-link te-link-name { # Name of the TE link.
         label-switched-path lsp-name; # LSP used for the forwarding adjacency.
         local-address ip-address; # Local IP address associated with the TE link.
         remote-address ip-address; # Remote IP address mapped to the TE link.
         te-metric value; # Traffic engineering metric used for the TE link.
      }
   }
}
```

Configuring Link Management Protocol Peers

After you set up TE links, configure LMP network peers with the peer statement at the [edit protocols link-management] hierarchy level. A peer is the network device with which your routing platform communicates and establishes an FA-LSP. Designate a peer name, configure the peer router ID as the address (often a loopback address), and apply the TE link to be associated with this peer. Remember to configure both sides of a peering relationship to enable bidirectional communication.

Unlike GMPLS, you must not configure a control channel for a peer. If you include a control channel, the configuration will fail to commit.

```
[edit]
protocols {
   link-management {
      peer peer-name { # Configure the name of your network peer.
         address ip-address; # Include the router ID of the peer.
         te-link te-link-name; # Assign a TE link to this peer.
      }
   }
}
```

Configuring Peer Interfaces in OSPF and RSVP

After you establish LMP peers, you must add peer interfaces to OSPF and RSVP. A peer interface is a virtual interface used to support the control adjacency between two peers.

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

```
[edit]
protocols {
   rsvp {
      peer-interface peer-name { # Configure the name of your LMP peer.
      }
      ospf {
         area  area-number  {
            peer-interface peer-name { # Configure the name of your LMP peer.
            }
         }
      }
   }
}
```

Establishing FA-LSP Path Information

When you configure explicit LSP paths for an FA-LSP, you must use the TE link remote address as your next-hop address. When CSPF is supported, you can use any path option you wish. However, when CSPF is disabled with the no-cspf statement at the [edit protocols mpls label-switched-path *lsp-name*] hierarchy level, you must use strict paths.

```
[edit]
protocols {
   mpls {
      path path-name {
         next-hop-address (strict | loose);
      }
   }
}
```

NOTE: If the end-to-end LSP originates on the same routing platform as the FA-LSP, you must disable CSPF and use strict paths.

Defining Label-Switched Paths for the FA-LSP

Next, define your FA-LSP by including the label-switched-path statement at the [edit protocols mpls] hierarchy level. Include the router ID of the peer in the to statement at the [edit protocols mpls label-switched-path] hierarchy level. Because packet LSPs are unidirectional, you must create one FA-LSP to reach the peer and a second FA-LSP to return from the peer.

```
[edit]
protocols {
   mpls {
      label-switched-path lsp-name {
         from ip-address;
         to ip-address;
         primary path-name;
         secondary path-name;
         no-cspf; # This statement to disable CSPF is optional.
      }
   }
}
```

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

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

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

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

Option: Tearing Down RSVP LSPs Gracefully

You can tear down an RSVP LSP in a two-step process that gracefully withdraws the RSVP session used by the LSP. For all neighbors that support graceful teardown, a request for the teardown is sent by the routing platform to the destination endpoint for the LSP and all RSVP neighbors in the path. The request is included within the

ADMIN_STATUS field of the RSVP packet. When neighbors receive the request, they prepare for the RSVP session to be withdrawn. A second message is sent by the routing platform to complete the teardown of the RSVP session. If a neighbor does not support graceful teardown, the request is handled as a standard session teardown rather than a graceful one.

To perform a graceful teardown of an RSVP session, issue the clear rsvp session gracefully command. Optionally, you can specify the source and destination address of the RSVP session, the LSP identifier of the RSVP sender, and the tunnel identifier of the RSVP session. To use these qualifiers, include the connection-source, connection-destination, lsp-id, and tunnel-id options when you issue the clear rsvp session gracefully command.

You can also configure the amount of time that the routing platform waits for neighbors to receive the graceful teardown request before initiating the actual teardown. To configure, include the graceful-deletion-timeout statement at the [edit protocols rsvp] hierarchy level. The default graceful deletion timeout value is 30 seconds, with a minimum value of 1 second and a maximum value of 300 seconds. To view the current value configured for graceful deletion timeout, issue the show rsvp version operational mode command.

Example: RSVP LSP Tunnel Configuration

Figure 12: RSVP LSP Tunnel Topology Diagram

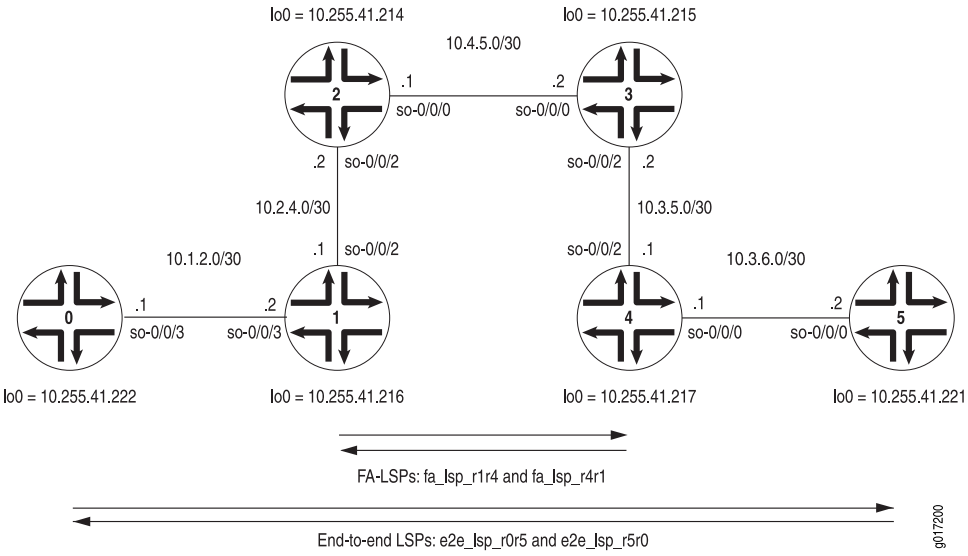


Figure 12 on page 145 shows an end-to-end RSVP LSP called e2e_lsp_r0r5 that originates on Router 0 and terminates on Router 5. In transit, this LSP traverses the FA-LSP fa_lsp_r1r4. The return path is represented by the end-to-end RSVP LSP e2e_lsp_r5r0 that travels over the FA-LSP fa_lsp_r4r1.

On Router 0, configure the end-to-end RSVP LSP that travels to Router 5. Use a strict path that traverses Router 1 and the LMP TE link traveling from Router 1 to Router 4.

Router 0 [edit]

```
interfaces {
   so-0/0/3 {
      unit 0 {
         family inet {
            address 10.1.2.1/30;
         }
         family mpls;
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.255.41.222/32;
         }
         family mpls;
      }
   }
}
routing-options {
   forwarding-table {
      export pplb;
   }
}
protocols {
   rsvp {
      interface all;
      interface fxp0.0 {
         disable;
      }
   }
   mpls {
      admin-groups {
         fa 1;
         backup 2;
         other 3;
      }
      label-switched-path e2e_lsp_r0r5 { # An end-to-end LSP traveling to Router 5.
         to 10.255.41.221;
         bandwidth 30k;
         primary path-fa; # Reference the requested path here.
      }
      path path-fa { # Configure the strict path here.
         10.1.2.2 strict;
         172.16.30.2 strict; # This traverses the TE link heading to Router 4.
      }
      interface all;
      interface fxp0.0 {
         disable;
      }
      interface so-3/2/1.0 {
         admin-group other;
      }
      interface so-0/0/3.0 {
         admin-group other;
      }
   }
```

```
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface fxp0.0 {
                disable;
            }
            interface all;
        }
    }
}
policy-options {
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
    }
}
```

On Router 1, configure an FA-LSP to reach Router 4. Establish an LMP TE link and LMP peer relationship with Router 4. Reference the FA-LSP in the TE link and add the peer interface into both OSPF and RSVP.

When the return path end-to-end LSP arrives at Router 1, the routing platform performs a routing lookup and can forward traffic to Router 0. Make sure you configure OSPF correctly between Routers 0 and 1.

Router 1

```
[edit]
interfaces {
    so-0/0/1 {
        unit 0 {
            family inet {
                address 10.2.3.1/30;
            }
            family mpls;
        }
    }
    so-0/0/2 {
        unit 0 {
            family inet {
                address 10.2.4.1/30;
            }
            family mpls;
        }
    }
    so-0/0/3 {
        unit 0 {
            family inet {
                address 10.1.2.2/30;
            }
            family mpls;
        }
    }
    fe-0/1/2 {
        unit 0 {
            family inet {
                address 10.2.5.1/30;
```

```
            }
            family mpls;
        }
    }
    at-1/0/0 {
        atm-options {
            vpi 1;
        }
        unit 0 {
            vci 1.100;
            family inet {
                address 10.2.3.5/30;
            }
            family mpls;
        }
    }
}
routing-options {
    forwarding-table {
        export [ pplb choose_lsp ];
    }
}
protocols {
    rsvp {
        interface all;
        interface fxp0.0 {
            disable;
        }
        peer-interface r4; # Apply the LMP peer interface here.
    }
    mpls {
        admin-groups {
            fa 1;
            backup 2;
            other 3;
        }
        label-switched-path fa_lsp_r1r4 { # Configure your FA-LSP to Router 4 here.
            to 10.255.41.217;
            bandwidth 400k;
            primary path_r1r4; # Apply the FA-LSP path here.
        }
        path path_r1r4 { # Configure the FA-LSP path here.
            10.2.4.2;
            10.4.5.2;
            10.3.5.1;
        }
        interface so-0/0/3.0 {
            admin-group other;
        }
        interface so-0/0/1.0 {
            admin-group fa;
        }
        interface at-1/0/0.0 {
            admin-group backup;
        }
        interface fe-0/1/2.0 {
```

```
            admin-group backup;
        }
        interface so-0/0/2.0 {
            admin-group fa;
        }
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface fxp0.0 {
                disable;
            }
            interface all;
            peer-interface r4; # Apply the LMP peer interface here.
        }
    }
    link-management { # Configure LMP statements here.
        te-link link_r1r4 { # Assign a name to the TE link here.
            local-address 172.16.30.1; # Configure a local address for the TE link.
            remote-address 172.16.30.2; # Configure a remote address for the TE link.
            te-metric 1; # Manually set a metric here if you are not relying on CSPF.
            label-switched-path fa_lsp_r1r4; # Reference the FA-LSP here.
        }
        peer r4 { # Configure LMP peers here.
            address 10.255.41.217; # Configure the loopback address of your peer here.
            te-link link_r1r4; # Apply the LMP TE link here.
        }
    }
}
policy-options {
    policy-statement choose_lsp {
        term A {
            from community choose_e2e_lsp;
            then {
                install-nexthop strict lsp e2e_lsp_r1r4;
                accept;
            }
        }
        term B {
            from community choose_fa_lsp;
            then {
                install-nexthop strict lsp fa_lsp_r1r4;
                accept;
            }
        }
    }
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
    }
    community choose_e2e_lsp members 1000:1000;
    community choose_fa_lsp members 2000:2000;
    community set_e2e_lsp members 1000:1000;
    community set_fa_lsp members 2000:2000;
}
```

On Router 2, configure OSPF, MPLS, and RSVP on all interfaces that transport the FA-LSPs across the core network.

Router 2

```
[edit]
interfaces {
   so-0/0/0 {
      unit 0 {
         family inet {
            address 10.4.5.1/30;
         }
         family mpls;
      }
   }
   so-0/0/1 {
      unit 0 {
         family inet {
            address 10.1.4.2/30;
         }
         family mpls;
      }
   }
   so-0/0/2 {
      unit 0 {
         family inet {
            address 10.2.4.2/30;
         }
         family mpls;
      }
   }
   fe-0/1/2 {
      unit 0 {
         family inet {
            address 10.3.4.2/30;
         }
         family mpls;
      }
   }
}
routing-options {
   forwarding-table {
      export pplb;
   }
}
protocols { # OSPF, MPLS, and RSVP form the core backbone for the FA-LSPs.
   rsvp {
      interface all;
      interface fxp0.0 {
         disable;
      }
   }
   mpls {
      admin-groups {
         fa 1;
         backup 2;
         other 3;
      }
```

```
        path path_r1 {
            10.2.4.1;
        }
        path path_r3r4 {
            10.4.5.2;
            10.3.5.1;
        }
        interface all;
        interface fxp0.0 {
            disable;
        }
        interface so-0/0/1.0 {
            admin-group other;
        }
        interface fe-0/1/2.0 {
            admin-group backup;
        }
        interface so-0/0/2.0 {
            admin-group fa;
        }
        interface so-0/0/0.0 {
            admin-group fa;
        }
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface fxp0.0 {
                disable;
            }
            interface all;
        }
    }
}
policy-options {
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
    }
}
```

On Router 3, configure OSPF, MPLS, and RSVP on all interfaces that transport the FA-LSPs across the core network.

Router 3

```
[edit]
interfaces {
    so-0/0/0 {
        unit 0 {
            family inet {
                address 10.4.5.2/30;
            }
            family mpls;
        }
    }
    so-0/0/1 {
```

```
        unit 0 {
            family inet {
                address 10.5.6.1/30;
            }
            family mpls;
        }
    }
    so-0/0/2 {
        unit 0 {
            family inet {
                address 10.3.5.2/30;
            }
            family mpls;
        }
    }
    fe-0/1/2 {
        unit 0 {
            family inet {
                address 10.2.5.2/30;
            }
            family mpls;
        }
    }
}
routing-options {
    forwarding-table {
        export pplb;
    }
}
protocols { # OSPF, MPLS, and RSVP form the core backbone for the FA-LSPs.
    rsvp {
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
    mpls {
        admin-groups {
            fa 1;
            backup 2;
            other 3;
        }
        path path_r4 {
            10.3.5.1;
        }
        path path_r2r1 {
            10.4.5.1;
            10.2.4.1;
        }
        interface all;
        interface fxp0.0 {
            disable;
        }
        interface so-0/0/2.0 {
            admin-group fa;
        }
```

```
        interface fe-0/1/2.0 {
            admin-group backup;
        }
        interface so-0/0/1.0 {
            admin-group other;
        }
        interface so-0/0/0.0 {
            admin-group fa;
        }
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface fxp0.0 {
                disable;
            }
            interface all;
        }
    }
}
policy-options {
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
    }
}
```

On Router 4, configure a return path FA-LSP to reach Router 1. Establish an LMP TE link and LMP peer relationship with Router 1. Reference the FA-LSP in the TE link and add the peer interface into both OSPF and RSVP.

When the initial end-to-end LSP arrives at Router 4, the routing platform performs a routing lookup and can forward traffic to Router 5. Make sure you configure OSPF correctly between Routers 4 and 5.

Router 4

```
[edit]
interfaces {
    so-0/0/0 {
        unit 0 {
            family inet {
                address 10.3.6.1/30;
            }
            family mpls;
        }
    }
    so-0/0/1 {
        unit 0 {
            family inet {
                address 10.2.3.2/30;
            }
            family mpls;
        }
    }
    so-0/0/2 {
        unit 0 {
```

```
            family inet {
               address 10.3.5.1/30;
            }
            family mpls;
         }
      }
      fe-0/1/2 {
         unit 0 {
            family inet {
               address 10.3.4.1/30;
            }
            family mpls;
         }
      }
      at-1/0/0 {
         atm-options {
            vpi 1;
         }
         unit 0 {
            vci 1.100;
            family inet {
               address 10.2.3.6/30;
            }
            family mpls;
         }
      }
   }
   routing-options {
      forwarding-table {
         export [ pplb choose_lsp ];
      }
   }
   protocols {
      rsvp {
         interface all;
         interface fxp0.0 {
            disable;
         }
         peer-interface r1; # Apply the LMP peer interface here.
      }
      mpls {
         admin-groups {
            fa 1;
            backup 2;
            other 3;
         }
         label-switched-path fa_lsp_r4r1 { # Configure your FA-LSP here.
            to 10.255.41.216;
            bandwidth 400k;
            primary path_r4r1; # Apply the FA-LSP path here.
         }
         path path_r4r1 { # Configure the FA-LSP path here.
            10.3.5.2;
            10.4.5.1;
            10.2.4.1;
         }
```

```
        interface all;
        interface fxp0.0 {
            disable;
        }
        interface at-1/0/0.0 {
            admin-group backup;
        }
        interface so-0/0/2.0 {
            admin-group fa;
        }
        interface fe-0/1/2.0 {
            admin-group backup;
        }
        interface so-0/0/0.0 {
            admin-group other;
        }
        interface so-0/0/1.0 {
            admin-group fa;
        }
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface fxp0.0 {
                disable;
            }
            interface all;
            peer-interface r1; # Apply the LMP peer interface here.
        }
    }
    link-management { # Configure LMP statements here.
        te-link link_r4r1 { # Assign a name to the TE link here.
            local-address 172.16.30.2; # Configure a local address for the TE link.
            remote-address 172.16.30.1; # Configure a remote address for the TE link.
            te-metric 1; # Manually set a metric here if you are not relying on CSPF.
            label-switched-path fa_lsp_r4r1; # Reference the FA-LSP here.
        }
        peer r1 { # Configure LMP peers here.
            address 10.255.41.216; # Configure the loopback address of your peer here.
            te-link link_r4r1; # Apply the LMP TE link here.
        }
    }
}
policy-options {
    policy-statement choose_lsp {
        term A {
            from community choose_e2e_lsp;
            then {
                install-nexthop strict lsp e2e_lsp_r4r1;
                accept;
            }
        }
        term B {
            from community choose_fa_lsp;
            then {
                install-nexthop strict lsp fa_lsp_r4r1;
```

```
            accept;
          }
        }
      }
      policy-statement pplb {
        then {
          load-balance per-packet;
        }
      }
      community choose_e2e_lsp members 1000:1000;
      community choose_fa_lsp members 2000:2000;
      community set_e2e_lsp members 1000:1000;
      community set_fa_lsp members 2000:2000;
    }
```

On Router 5, configure the return path end-to-end RSVP LSP that travels to Router 0. Use a strict path that traverses Router 4 and the LMP TE link traveling from Router 4 to Router 1.

Router 5

```
[edit]
interfaces {
    so-0/0/2 {
        unit 0 {
            family inet {
                address 10.3.6.2/30;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.41.221/32;
            }
        }
    }
}
routing-options {
    forwarding-table {
        export pplb;
    }
}
protocols {
    rsvp {
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
    mpls {
        admin-groups {
            fa 1;
            backup 2;
            other 3;
        }
        label-switched-path e2e_lsp_r5r0 { # An end-to-end LSP returning to Router 0.
```

```
            to 10.255.41.222;
            bandwidth 30k;
            primary path-fa; # Reference the requested path here.
        }
        path path-fa { # Configure the strict path here.
            10.3.6.1 strict;
            172.16.30.1 strict; # This traverses the TE link heading to Router 1.
        }
        interface all;
        interface fxp0.0 {
            disable;
        }
        interface so-0/0/2.0 {
            admin-group other;
        }
        interface so-0/0/1.0 {
            admin-group other;
        }
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface fxp0.0 {
                disable;
            }
            interface all;
        }
    }
}
policy-options {
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
    }
}
```

Verifying Your Work

To verify that your RSVP LSP tunnel is working correctly, issue the following commands:

- show ted database (extensive)
- show rsvp session name (extensive)
- show link-management
- show link-management te-link name (detail)

To see these commands used with the configuration example, see the following sections:

- Router 0 on page 158
- Router 1 on page 162

Router 0

On Router 0, you can verify that the FA-LSPs appear as valid paths in the traffic engineering database. In this case, look for the paths from Router 1 (10.255.41.216) and Router 4 (10.255.41.217) that reference the LMP TE link addresses of 172.16.30.1 and 172.16.30.2. You can also issue the show rsvp session extensive command to look for the path of the end-to-end LSP as it travels to Router 5 over the FA-LSP.

```
user@router0> show ted database
TED database: 0 ISIS nodes 8 INET nodes
ID                          Type Age(s) LnkIn LnkOut Protocol
10.255.41.214               Rtr     486     4      4 OSPF(0.0.0.0)
    To: 10.255.41.222, Local: 10.1.4.2, Remote: 10.1.4.1
    To: 10.255.41.216, Local: 10.2.4.2, Remote: 10.2.4.1
    To: 10.255.41.215, Local: 10.4.5.1, Remote: 10.4.5.2
    To: 10.3.4.1-1, Local: 10.3.4.2, Remote: 0.0.0.0
ID                          Type Age(s) LnkIn LnkOut Protocol
10.255.41.215               Rtr     187     4      4 OSPF(0.0.0.0)
    To: 10.255.41.214, Local: 10.4.5.2, Remote: 10.4.5.1
    To: 10.255.41.217, Local: 10.3.5.2, Remote: 10.3.5.1
    To: 10.255.41.221, Local: 10.5.6.1, Remote: 10.5.6.2
    To: 10.2.5.1-1, Local: 10.2.5.2, Remote: 0.0.0.0
ID                          Type Age(s) LnkIn LnkOut Protocol
10.255.41.216                Rtr     396     6      6 OSPF(0.0.0.0)
    To: 10.255.41.222, Local: 10.1.2.2, Remote: 10.1.2.1
    To: 10.255.41.214, Local: 10.2.4.1, Remote: 10.2.4.2
    To: 10.255.41.217, Local: 10.2.3.1, Remote: 10.2.3.2
     To: 10.255.41.217, Local: 172.16.30.1, Remote: 172.16.30.2
    To: 10.255.41.217, Local: 10.2.3.5, Remote: 10.2.3.6
    To: 10.2.5.1-1, Local: 10.2.5.1, Remote: 0.0.0.0
ID                          Type Age(s) LnkIn LnkOut Protocol
10.255.41.217                Rtr     404     6      6 OSPF(0.0.0.0)
    To: 10.255.41.216, Local: 10.2.3.2, Remote: 10.2.3.1
     To: 10.255.41.216, Local: 172.16.30.2, Remote: 172.16.30.1
    To: 10.255.41.216, Local: 10.2.3.6, Remote: 10.2.3.5
    To: 10.255.41.215, Local: 10.3.5.1, Remote: 10.3.5.2
    To: 10.255.41.221, Local: 10.3.6.1, Remote: 10.3.6.2
    To: 10.3.4.1-1, Local: 10.3.4.1, Remote: 0.0.0.0
ID                          Type Age(s) LnkIn LnkOut Protocol
10.255.41.221               Rtr     481     2      2 OSPF(0.0.0.0)
    To: 10.255.41.215, Local: 10.5.6.2, Remote: 10.5.6.1
    To: 10.255.41.217, Local: 10.3.6.2, Remote: 10.3.6.1
ID                          Type Age(s) LnkIn LnkOut Protocol
10.255.41.222               Rtr    2883     2      2 OSPF(0.0.0.0)
    To: 10.255.41.216, Local: 10.1.2.1, Remote: 10.1.2.2
    To: 10.255.41.214, Local: 10.1.4.1, Remote: 10.1.4.2

user@router0>  show ted database 10.255.41.216 extensive
TED database: 0 ISIS nodes 8 INET nodes
NodeID: 10.255.41.216
  Type: Rtr, Age: 421 secs, LinkIn: 6, LinkOut: 6
  Protocol: OSPF(0.0.0.0)
    To: 10.255.41.222, Local: 10.1.2.2, Remote: 10.1.2.1
      Color: 0x8 other
      Metric: 1
      Static BW: 155.52Mbps
      Reservable BW: 155.52Mbps
```

```
  Available BW [priority] bps:
      [0] 155.4Mbps    [1] 155.4Mbps   [2] 155.4Mbps   [3] 155.4Mbps
      [4] 155.4Mbps    [5] 155.4Mbps   [6] 155.4Mbps   [7] 155.4Mbps
  Interface Switching Capability Descriptor(1):
    Switching type: Packet
    Encoding type: Packet
    Maximum LSP BW [priority] bps:
      [0] 155.4Mbps    [1] 155.4Mbps   [2] 155.4Mbps   [3] 155.4Mbps
      [4] 155.4Mbps    [5] 155.4Mbps   [6] 155.4Mbps   [7] 155.4Mbps
To: 10.255.41.214, Local: 10.2.4.1, Remote: 10.2.4.2
  Color: 0x2 fa
  Metric: 1
  Static BW: 155.52Mbps
  Reservable BW: 155.52Mbps
  Available BW [priority] bps:
      [0] 155.12Mbps   [1] 155.12Mbps  [2] 155.12Mbps  [3] 155.12Mbps
      [4] 155.12Mbps   [5] 155.12Mbps  [6] 155.12Mbps  [7] 155.12Mbps
  Interface Switching Capability Descriptor(1):
    Switching type: Packet
    Encoding type: Packet
    Maximum LSP BW [priority] bps:
      [0] 155.12Mbps   [1] 155.12Mbps  [2] 155.12Mbps  [3] 155.12Mbps
      [4] 155.12Mbps   [5] 155.12Mbps  [6] 155.12Mbps  [7] 155.12Mbps
To: 10.255.41.217, Local: 10.2.3.1, Remote: 10.2.3.2
  Color: 0x2 fa
  Metric: 1
  Static BW: 155.52Mbps
  Reservable BW: 155.52Mbps
  Available BW [priority] bps:
      [0] 155.52Mbps   [1] 155.52Mbps  [2] 155.52Mbps  [3] 155.52Mbps
      [4] 155.52Mbps   [5] 155.52Mbps  [6] 155.52Mbps  [7] 155.52Mbps
  Interface Switching Capability Descriptor(1):
    Switching type: Packet
    Encoding type: Packet
    Maximum LSP BW [priority] bps:
      [0] 155.52Mbps   [1] 155.52Mbps  [2] 155.52Mbps  [3] 155.52Mbps
      [4] 155.52Mbps   [5] 155.52Mbps  [6] 155.52Mbps  [7] 155.52Mbps
To: 10.255.41.217,  Local: 172.16.30.1, Remote: 172.16.30.2
  Metric: 1
  Static BW: 400kbps
  Reservable BW: 400kbps
  Available BW [priority] bps:
      [0] 370kbps      [1] 370kbps     [2] 370kbps     [3] 370kbps
      [4] 370kbps      [5] 370kbps     [6] 370kbps     [7] 370kbps
  Interface Switching Capability Descriptor(1):
    Switching type: Packet
    Encoding type: Packet
    Maximum LSP BW [priority] bps:
      [0] 370kbps      [1] 370kbps     [2] 370kbps     [3] 370kbps
      [4] 370kbps      [5] 370kbps     [6] 370kbps     [7] 370kbps
To: 10.255.41.217, Local: 10.2.3.5, Remote: 10.2.3.6
  Color: 0x4 backup
  Metric: 1
  Static BW: 155.52Mbps
  Reservable BW: 155.52Mbps
  Available BW [priority] bps:
      [0] 155.52Mbps   [1] 155.52Mbps  [2] 155.52Mbps  [3] 155.52Mbps
      [4] 155.52Mbps   [5] 155.52Mbps  [6] 155.52Mbps  [7] 155.52Mbps
  Interface Switching Capability Descriptor(1):
    Switching type: Packet
    Encoding type: Packet
```

```
        Maximum LSP BW [priority] bps:
          [0] 155.52Mbps   [1] 155.52Mbps  [2] 155.52Mbps  [3] 155.52Mbps
          [4] 155.52Mbps   [5] 155.52Mbps  [6] 155.52Mbps  [7] 155.52Mbps
    To: 10.2.5.1-1, Local: 10.2.5.1, Remote: 0.0.0.0
      Color: 0x4 backup
      Metric: 1
      Static BW: 100Mbps
      Reservable BW: 100Mbps
      Available BW [priority] bps:
          [0] 100Mbps      [1] 100Mbps     [2] 100Mbps     [3] 100Mbps
          [4] 100Mbps      [5] 100Mbps     [6] 100Mbps     [7] 100Mbps
      Interface Switching Capability Descriptor(1):
        Switching type: Packet
        Encoding type: Packet
        Maximum LSP BW [priority] bps:
          [0] 100Mbps      [1] 100Mbps     [2] 100Mbps     [3] 100Mbps
          [4] 100Mbps      [5] 100Mbps     [6] 100Mbps     [7] 100Mbps

user@router0>  show ted database 10.255.41.217 extensive
TED database: 0 ISIS nodes 8 INET nodes
NodeID: 10.255.41.217
  Type: Rtr, Age: 473 secs, LinkIn: 6, LinkOut: 6
  Protocol: OSPF(0.0.0.0)
    To: 10.255.41.216, Local: 10.2.3.2, Remote: 10.2.3.1
      Color: 0x2 fa
      Metric: 1
      Static BW: 155.52Mbps
      Reservable BW: 155.52Mbps
      Available BW [priority] bps:
          [0] 155.52Mbps   [1] 155.52Mbps  [2] 155.52Mbps  [3] 155.52Mbps
          [4] 155.52Mbps   [5] 155.52Mbps  [6] 155.52Mbps  [7] 155.52Mbps
      Interface Switching Capability Descriptor(1):
        Switching type: Packet
        Encoding type: Packet
        Maximum LSP BW [priority] bps:
          [0] 155.52Mbps   [1] 155.52Mbps  [2] 155.52Mbps  [3] 155.52Mbps
          [4] 155.52Mbps   [5] 155.52Mbps  [6] 155.52Mbps  [7] 155.52Mbps
    To: 10.255.41.216,  Local: 172.16.30.2, Remote: 172.16.30.1
      Metric: 1
      Static BW: 400kbps
      Reservable BW: 400kbps
      Available BW [priority] bps:
          [0] 370kbps      [1] 370kbps     [2] 370kbps     [3] 370kbps
          [4] 370kbps      [5] 370kbps     [6] 370kbps     [7] 370kbps
      Interface Switching Capability Descriptor(1):
        Switching type: Packet
        Encoding type: Packet
        Maximum LSP BW [priority] bps:
          [0] 370kbps      [1] 370kbps     [2] 370kbps     [3] 370kbps
          [4] 370kbps      [5] 370kbps     [6] 370kbps     [7] 370kbps
    To: 10.255.41.216, Local: 10.2.3.6, Remote: 10.2.3.5
      Color: 0x4 backup
      Metric: 1
      Static BW: 155.52Mbps
      Reservable BW: 155.52Mbps
      Available BW [priority] bps:
          [0] 155.52Mbps   [1] 155.52Mbps  [2] 155.52Mbps  [3] 155.52Mbps
          [4] 155.52Mbps   [5] 155.52Mbps  [6] 155.52Mbps  [7] 155.52Mbps
      Interface Switching Capability Descriptor(1):
        Switching type: Packet
        Encoding type: Packet
```

```
        Maximum LSP BW [priority] bps:
          [0] 155.52Mbps   [1] 155.52Mbps  [2] 155.52Mbps  [3] 155.52Mbps
          [4] 155.52Mbps   [5] 155.52Mbps  [6] 155.52Mbps  [7] 155.52Mbps
    To: 10.255.41.215, Local: 10.3.5.1, Remote: 10.3.5.2
      Color: 0x2 fa
      Metric: 1
      Static BW: 155.52Mbps
      Reservable BW: 155.52Mbps
      Available BW [priority] bps:
          [0] 155.12Mbps   [1] 155.12Mbps  [2] 155.12Mbps  [3] 155.12Mbps
          [4] 155.12Mbps   [5] 155.12Mbps  [6] 155.12Mbps  [7] 155.12Mbps
      Interface Switching Capability Descriptor(1):
        Switching type: Packet
        Encoding type: Packet
        Maximum LSP BW [priority] bps:
          [0] 155.12Mbps   [1] 155.12Mbps  [2] 155.12Mbps  [3] 155.12Mbps
          [4] 155.12Mbps   [5] 155.12Mbps  [6] 155.12Mbps  [7] 155.12Mbps
    To: 10.255.41.221, Local: 10.3.6.1, Remote: 10.3.6.2
      Color: 0x8 other
      Metric: 1
      Static BW: 155.52Mbps
      Reservable BW: 155.52Mbps
      Available BW [priority] bps:
          [0] 155.52Mbps   [1] 155.52Mbps  [2] 155.52Mbps  [3] 155.52Mbps
          [4] 155.52Mbps   [5] 155.52Mbps  [6] 155.52Mbps  [7] 155.52Mbps
      Interface Switching Capability Descriptor(1):
        Switching type: Packet
        Encoding type: Packet
        Maximum LSP BW [priority] bps:
          [0] 155.52Mbps   [1] 155.52Mbps  [2] 155.52Mbps  [3] 155.52Mbps
          [4] 155.52Mbps   [5] 155.52Mbps  [6] 155.52Mbps  [7] 155.52Mbps
    To: 10.3.4.1-1, Local: 10.3.4.1, Remote: 0.0.0.0
      Color: 0x4 backup
      Metric: 1
      Static BW: 100Mbps
      Reservable BW: 100Mbps
      Available BW [priority] bps:
          [0] 100Mbps      [1] 100Mbps     [2] 100Mbps     [3] 100Mbps
          [4] 100Mbps      [5] 100Mbps     [6] 100Mbps     [7] 100Mbps
      Interface Switching Capability Descriptor(1):
        Switching type: Packet
        Encoding type: Packet
        Maximum LSP BW [priority] bps:
          [0] 100Mbps      [1] 100Mbps     [2] 100Mbps     [3] 100Mbps
          [4] 100Mbps      [5] 100Mbps     [6] 100Mbps     [7] 100Mbps

user@router0>  show rsvp session name e2e_lsp_r0r5 extensive
Ingress RSVP: 1 sessions
10.255.41.221
  From: 10.255.41.222, LSPstate: Up, ActiveRoute: 2
  LSPname: e2e_lsp_r0r5, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 101584
  Resv style: 1 FF, Label in: -, Label out: 101584
  Time left:    -, Since: Wed Sep  7 19:02:56 2005
  Tspec: rate 30kbps size 30kbps peak Infbps m 20 M 1500
  Port number: sender 2 receiver 29458 protocol 0
  PATH rcvfrom: localclient
  Adspec: sent MTU 1500
  Path MTU: received 1500
  PATH sentto: 10.1.2.2 (so-0/0/3.0) 15 pkts
```

```
  RESV rcvfrom: 10.1.2.2 (so-0/0/3.0) 16 pkts
  Explct route: 10.1.2.2 172.16.30.2 10.3.6.2
   Record route: <self> 10.1.2.2 172.16.30.2 10.3.6.2
Total 1 displayed, Up 1, Down 0

Egress RSVP: 1 sessions
Total 0 displayed, Up 0, Down 0

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

Router 1

On Router 1, verify that your LMP TE link configuration is working and that the end-to-end LSP is traversing the TE link by issuing the show link-management set of commands. You can also issue the show rsvp session extensive command to confirm that the FA-LSP is operational.

```
user@router1> show link-management
Peer name: r4 , System identifier: 10758
 State: Up, Control address: 10.255.41.217
   TE links:
   link_r1r4

 TE link name: link_r1r4, State: Up
  Local identifier: 16299, Remote identifier: 0, Local address: 172.16.30.1,
Remote address: 172.16.30.2,
  Encoding: Packet, Switching: Packet, Minimum bandwidth: 0bps, Maximum bandwidth:
 400kbps,
  Total bandwidth: 400kbps, Available bandwidth: 370kbps
    Name        State Local ID  Remote ID      Bandwidth Used  LSP-name
    fa_lsp_r1r4 Up       22642          0        400kbps  Yes  e2e_lsp_r0r5

user@router1>  show link-management te-link name link_r1r4 detail
 TE link name: link_r1r4, State: Up
  Local identifier: 16299, Remote identifier: 0, Local address: 172.16.30.1,
Remote address: 172.16.30.2,
  Encoding: Packet, Switching: Packet, Minimum bandwidth: 0bps, Maximum bandwidth:
 400kbps,
  Total bandwidth: 400kbps, Available bandwidth: 370kbps
    Resource: fa_lsp_r1r4, Type: LSP, System identifier: 2147483683, State: Up,
Local identifier: 22642,
    Remote identifier: 0
    Total bandwidth: 400kbps, Unallocated bandwidth: 370kbps
   Traffic parameters: Encoding: Packet, Switching: Packet, Granularity: Unknown

     Number of allocations: 1, In use: Yes
      LSP name: e2e_lsp_r0r5,  Allocated bandwidth: 30kbps

user@router1>  show rsvp session name fa_lsp_r1r4 extensive
Ingress RSVP: 1 sessions
10.255.41.217
  From: 10.255.41.216, LSPstate: Up, ActiveRoute: 0
  LSPname: fa_lsp_r1r4, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 100816
  Resv style: 1 FF, Label in: -, Label out: 100816
  Time left:    -, Since: Wed Sep  7 19:02:33 2005
  Tspec: rate 400kbps size 400kbps peak Infbps m 20 M 1500
```

```
  Port number: sender 2 receiver 5933 protocol 0
  PATH rcvfrom: localclient
  Adspec: sent MTU 1500
  Path MTU: received 1500
  PATH sentto: 10.2.4.2 (so-0/0/2.0) 28 pkts
  RESV rcvfrom: 10.2.4.2 (so-0/0/2.0) 26 pkts
  Explct route: 10.2.4.2 10.4.5.2 10.3.5.1
  Record route: <self> 10.2.4.2 10.4.5.2 10.3.5.1
Total 1 displayed, Up 1, Down 0

Egress RSVP: 1 sessions
Total 0 displayed, Up 0, Down 0

Transit RSVP: 2 sessions
Total 0 displayed, Up 0, Down 0
```

For More Information

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

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

Revision History

13 April 2009—9.5R1 Release. Roy Spencer.

15 January 2009—9.4R1 Release. Roy Spencer.

10 October 2008—9.3R1 Release. Roy Spencer.

10 July 2008—9.2R1 Release. Roy Spencer.

10 April 2008—9.1R1 Release. Roy Spencer.

1 February 2008—9.0R1 Release. Fawn Damitio.

29 June 2007—8.4R1 Release. Fawn Damitio.

27 March 2007—8.3R1 Release. Fawn Damitio.

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

15 September 2006—8.1R1 Release. Richard Hendricks.

29 June 2006—8.0R1 Release. Richard Hendricks.

27 March 2006—7.6R1 Release. Richard Hendricks.

9 January 2006—7.5R1 Release. Richard Hendricks.

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

Chapter 6

Simplified Interinstance Route Sharing

This feature guide covers these topics:

Overview

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

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

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

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

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

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

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

System Requirements

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

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

Terms and Acronyms

R

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

V

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

Simplified Interinstance Configuration

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

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

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

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

For nonforwarding instances, routes are imported into the instance so routing protocols can announce them. For more information, see “Configuring Nonforwarding Instances” on page 181.

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

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

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

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

- The instance-import and instance-export commands cannot be used with VRF instances. They are equivalent to the vrf-import and vrf-export VRF-specific commands.
- Traceoptions in the master instance apply to the routing table export task. Consequently, such traceoptions are propagated to all other instances, although they can be modified as needed.
- The auto-export statement applies to VRF and non-VRF instances.
- Use of the command instance-import automatically enables auto-export for non-VRF instances.
- Some network administrators use the instance-import functionality to create communities of interest. By setting up different VPNs and sharing routes between instances as needed, administrators can tailor services to the needs of their customers. For an example, see “Configuring Nonforwarding Instances” on page 181.

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

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

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

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

Instance Export Using an IGP Export Policy

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

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

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

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

Configuring Overlapping VPNs

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

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

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

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

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

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

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

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

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

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

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

```
[edit]
routing-options {
  rib-groups {
    inet-access {
```

```
            import-rib inet.0;
        }
    }
}
routing-instances {
    VPN-A {
        routing-options {
            auto-export {
                family inet {
                    unicast {
                        rib-group inet-access;
                    }
                }
            }
        }
    }
    VPN-AB {
        routing-options {
            auto-export {
                family inet {
                    unicast {
                        rib-group inet-access;
                    }
                }
            }
        }
    }
}
```

There is a significant difference in how routing table groups are used in this case and how they are used more generally. Typically, routing table groups require the exporting routing table to be referenced as the primary import routing table in the rib-group configuration. In this case, the restriction is lifted and the routing table group functions as an additional list of tables that export routes.

NOTE: When upgrading to JUNOS Release 5.4 or later, be aware that route export behavior differs when using the auto-export command instead of rib-group export:

- When routes are exported between routing tables by using the rib-group statements, both primary routes (routes in the originating routing table) and secondary routes (routes imported from other routing tables) are exported to the remote PE routers. When the auto-export statement is used, only the primary routes from the originating routing table are exported.
- Routes exported from an originating VRF instance to another on the same PE now honor export policy changes to route attributes. When you use auto-export, you must add the originating route target(s) to the exported routes. With rib-group statements, no additional configuration is necessary.

Example: Configuring Overlapping VPNs

Figure 13: Overlapping VPNs Topology Diagram

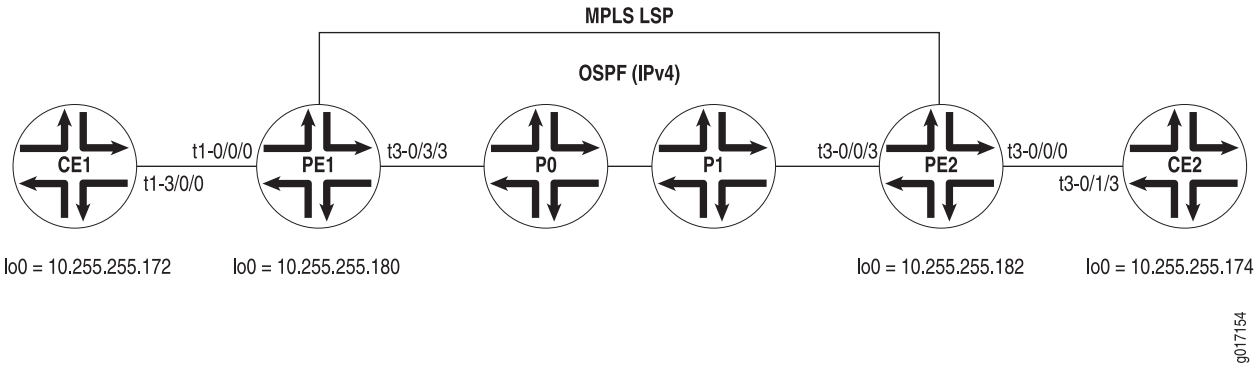


Figure 13 on page 173 shows a standard Multiprotocol Label Switching (MPLS) VPN topology. Routers PE1 and PE2 are acting as PE routers, CE1 and CE2 are CE routers, and P0 and P1 are core provider routers. You will establish three VRF instances: A, B, and AB. You will also configure auto-export as the method of sharing routing information between instances.

This example focuses on the interinstance and policy statements. As a result, some information has been omitted.

- Because PE1 uses static routing instances, the router configuration for CE1 is not included in this example.
- Most routers display a minimal configuration. Interface addresses and loopback addresses are assumed to have been enabled properly.

For more information about VPNs, see the *JUNOS VPNs Configuration Guide*.

Routers PE1 and PE2 contain the bulk of the configuration. At PE1, initiate an IBGP connection to PE2 and open a VPN connection to CE Router CE1 through three VRF instances: A, B, and AB.

The auto-export policy is applied to all instances simultaneously by means of a configuration group. Another method of enabling this option is to configure auto-export individually on each VRF instance.

Finally, the policy statements add the appropriate communities to each instance and accept traffic coming from the desired community. For example, the policy for VRF A sets community A on all outbound traffic leaving the instance, and only accepts traffic from PE2 that is tagged with community A.

Router PE1

```
[edit]
groups {
    vrf-export on {
        routing-instances {
            <*> {
                routing-options {
```

```
                auto-export;
            }
        }
    }
}
interfaces {
    t1-0/0/0
    description " to vpn02 t1-3/0/0";
    dce;
    encapsulation frame-relay;
    unit 0 {
        dlci 100;
        family inet {
            address 192.255.197.38/30;
        }
    }
    unit 1 {
        dlci 101;
        family inet {
            address 10.3.0.1/30;
        }
    }
    unit 2 {
        dlci 102;
        family inet {
            address 10.3.0.5/30;
        }
    }
}
lo0
unit 0
family inet {
    address 10.255.255.180/32;
}
protocols {
    mpls {
        interface all;
    }
    bgp {
        group pepe {
            type internal;
            neighbor 10.255.255.182 {
                family inet-vpn {
                    unicast;
                }
            }
        }
    }
    ospf {
        area 0.0.0.0 {
            interface t3-0/3/3.0;
            interface lo0.0 {
                passive;
            }
        }
```

```
    }
    ldp {
        interface all;
    }
}
policy-options {
    policy-statement A-in {
        from community A;
        then accept;
    }
    policy-statement A-out {
        then {
            community add A;
            accept;
        }
    }
    policy-statement B-in {
        from community B;
        then accept;
    }
    policy-statement B-out {
        then {
            community add B;
            accept;
        }
    }
    policy-statement AB-in {
        from community [A B];
        then accept;
    }
    policy-statement AB-out {
        then {
            community add A;
            community add B;
            accept;
        }
    }
    community A members target:69:1;
    community B members target:69:2;
}
routing-instances {
    apply-groups vrf-export-on;
    A {
        instance-type vrf;
        interface t1-0/0/0.0;
        route-distinguisher 10.255.255.180:69;
        vrf-import A-in;
        vrf-export A-out;
        routing-options {
            static {
                route 1.1.1.1/32 next-hop t1-0/0/0.0;
                route 1.1.1.2/32 next-hop t1-0/0/0.0;
            }
        }
    }
    AB {
```

```
        instance-type vrf;
        interface t1-0/0/0.2;
        route-distinguisher 10.255.255.180:69;
        vrf-import AB-in;
        vrf-export AB-out;
        routing-options {
            static {
                route 1.1.3.1/32 next-hop t1-0/0/0.2;
                route 1.1.3.2/32 next-hop t1-0/0/0.2;
            }
        }
    }
    B {
        instance-type vrf;
        interface t1-0/0/0.1;
        route-distinguisher 10.255.255.180:69;
        vrf-import B-in;
        vrf-export B-out;
        routing-options {
            static {
                route 1.1.2.1/32 next-hop t1-0/0/0.1;
                route 1.1.2.2/32 next-hop t1-0/0/0.1;
            }
        }
    }
}
```

As a provider core transit router, Router P0 only needs to provide connectivity to the PE routers. You configure OSPF, MPLS, and LDP on the interfaces pointing to both PE routers.

Router P0

```
[edit]
protocols {
    mpls {
        interface all;
    }
    ospf {
        area 0.0.0.0 {
            interface t3-0/0/3.0;
            interface t1-0/1/1.0;
        }
    }
    ldp {
        interface all;
    }
}
```

Like Router P0, Router P1 also needs to provide basic core connectivity for the PE routers. You can configure OSPF, MPLS, and LDP on the interfaces pointing toward routers P0 and PE2.

Router P1

```
[edit]
protocols {
    mpls {
        interface all;
```

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

At Router PE2, complete your IBGP connection to PE1 and finish the VPN connection to CE Router CE2 through VRF instance AB. The VRF import policy named AB-in is the same as the export policy used for the OSPF protocol in the AB VRF instance. The policy statements add communities A and B to all outbound routes and accept any traffic coming from these communities.

Router PE2

```
[edit]
interfaces {
  lo0
  unit 0
  family inet {
    address 10.255.255.182/32;
  }
}
protocols {
  mpls {
    interface all;
  }
  bgp {
    keep all;
    group pepe {
      type internal;
      neighbor 10.255.255.180 {
        family inet-vpn {
          unicast;
        }
      }
    }
  }
  ospf {
    area 0.0.0.0 {
      interface t3-0/0/3.0;
      interface lo0.0 {
        passive;
      }
    }
  }
  ldp {
    interface all;
  }
}
policy-options {
  policy-statement AB-in {
```

```
        from community [A B];
        then accept;
      }
      policy-statement AB-out {
        then {
          community add A;
          community add B;
          accept;
        }
      }
      community A members target:69:1;
      community B members target:69:2;
    }
    routing-instances {
      AB {
        instance-type vrf;
        interface t3-0/0/0.0;
        route-distinguisher 10.255.255.182:69;
        vrf-import AB-in;
        vrf-export AB-out;
        protocols {
          ospf {
            export AB-in;
            area 0.0.0.0 {
              interface all;
            }
          }
        }
      }
    }
```

At Router CE2, advertise the 10.255.255.174 loopback address into the VPN. Look for this route when you check the routing tables for the A, B, and AB instances on Router PE1. If the route appears in these instances, interinstance route sharing is successful.

Router CE2

```
[edit]
interfaces {
  lo0
  unit 0
  family inet {
    address 10.255.255.174/32;
  }
}
protocols {
  ospf {
    area 0.0.0.0 {
      interface t3-0/1/3.0;
      interface lo0.0;
    }
  }
}
```

Verifying Your Work

To verify that your overlapping VPN configuration is functioning properly, use the following commands:

- show route export table *table-name* (brief | detail)
- show route export instance *instance-name* (brief | detail)
- show route export vrf-target (community *community-regular-expression*) (brief | detail)

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

Router PE1 Status

```
user@PE1> show route export
Table                            Export           Routes
A.inet.0                         Y                     4
AB.inet.0                        Y                     4
B.inet.0                         Y                     4

user@PE1>  show route export detail
A.inet.0                                           Routes:        4
  Flags: <vrf>
AB.inet.0                                          Routes:        4
  Flags: <vrf>
B.inet.0                                           Routes:        4
  Flags: <vrf>

user@PE1>  show route export instance detail
Instance: A                                Type: vrf
  Flags: <config> Options: <unicast multicast>
Instance: AB                               Type: vrf
  Flags: <config> Options: <unicast multicast>
Instance: B                               Type: vrf
  Flags: <config> Options: <unicast multicast>

user@PE1>  show route table A.inet.0

A.inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
1.1.1.1/32         *[Static/5] 02:08:14
                    > via t1-0/0/0.0
1.1.1.2/32         *[Static/5] 02:08:14
                    > via t1-0/0/0.0
1.1.3.1/32         *[Static/5] 02:08:14
                    > via t1-0/0/0.2
1.1.3.2/32         *[Static/5] 02:08:14
                    > via t1-0/0/0.2
10.3.0.4/30        *[Direct/0] 02:08:14
                    > via t1-0/0/0.2
10.3.0.5/32        *[Local/0] 02:08:14
                      Local via t1-0/0/0.2
10.255.255.174/32   *[BGP/170] 00:18:08, MED 2, localpref 100, from 10.255.255.182
                      AS path: I
```

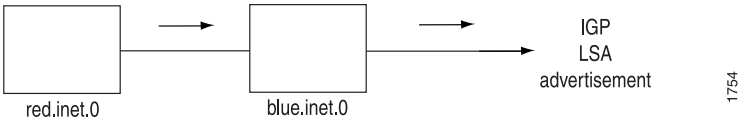
```
                    > via t3-0/3/3.0, Push 100004, Push 100017(top)
192.255.197.36/30  *[Direct/0] 02:08:14
                    > via t1-0/0/0.0
192.255.197.38/32  *[Local/0] 02:08:14
                      Local via t1-0/0/0.0
192.255.197.248/30 *[BGP/170] 00:18:18, localpref 100, from 10.255.255.182
                      AS path: I
                    > via t3-0/3/3.0, Push 100003, Push 100017(top)

user@PE1>  show route table B.inet.0

B.inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
1.1.2.1/32         *[Static/5] 02:09:28
                    > via t1-0/0/0.1
1.1.2.2/32         *[Static/5] 02:09:28
                    > via t1-0/0/0.1
1.1.3.1/32         *[Static/5] 02:09:28
                    > via t1-0/0/0.2
1.1.3.2/32         *[Static/5] 02:09:28
                    > via t1-0/0/0.2
10.3.0.0/30        *[Direct/0] 02:09:28
                    > via t1-0/0/0.1
10.3.0.1/32        *[Local/0] 02:09:28
                      Local via t1-0/0/0.1
10.3.0.4/30        *[Direct/0] 02:09:28
                    > via t1-0/0/0.2
10.3.0.5/32        *[Local/0] 02:09:28
                      Local via t1-0/0/0.2
10.255.255.174/32   *[BGP/170] 00:19:22, MED 2, localpref 100, from 10.255.255.182

                      AS path: I
                    > via t3-0/3/3.0, Push 100004, Push 100017(top)
192.255.197.248/30 *[BGP/170] 00:19:32, localpref 100, from 10.255.255.182
                      AS path: I
                    > via t3-0/3/3.0, Push 100003, Push 100017(top)

user@PE1>  show route table AB.inet.0
AB.inet.0: 14 destinations, 14 routes (14 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
1.1.1.1/32         *[Static/5] 02:09:43
                    > via t1-0/0/0.0
1.1.1.2/32         *[Static/5] 02:09:43
                    > via t1-0/0/0.0
1.1.2.1/32         *[Static/5] 02:09:43
                    > via t1-0/0/0.1
1.1.2.2/32         *[Static/5] 02:09:43
                    > via t1-0/0/0.1
1.1.3.1/32         *[Static/5] 02:09:43
                    > via t1-0/0/0.2
1.1.3.2/32         *[Static/5] 02:09:43
                    > via t1-0/0/0.2
10.3.0.0/30        *[Direct/0] 02:09:43
                    > via t1-0/0/0.1
10.3.0.1/32        *[Local/0] 02:09:43
                      Local via t1-0/0/0.1
10.3.0.4/30        *[Direct/0] 02:09:43
                    > via t1-0/0/0.2
10.3.0.5/32        *[Local/0] 02:09:43
                      Local via t1-0/0/0.2
10.255.255.174/32   *[BGP/170] 00:19:37, MED 2, localpref 100, from 10.255.255.182
```

```
                    AS path: I
                  > via t3-0/3/3.0, Push 100004, Push 100017(top)
192.255.197.36/30  *[Direct/0] 02:09:43
                    > via t1-0/0/0.0
192.255.197.38/32  *[Local/0] 02:09:43
                      Local via t1-0/0/0.0
192.255.197.248/30 *[BGP/170] 00:19:47, localpref 100, from 10.255.255.182
                      AS path: I
                    > via t3-0/3/3.0, Push 100003, Push 100017(top)

user@PE1>  show route export vrf-target detail
Target: 69:1                              inet     unicast
   Import table(s): A.inet.0 AB.inet.0
  Export table(s): A.inet.0 AB.inet.0
Target: 69:2                              inet     unicast
  Import table(s): AB.inet.0 B.inet.0
  Export table(s): AB.inet.0 B.inet.0
```

Configuring Nonforwarding Instances

In nonforwarding instances implemented in JUNOS Release 5.3 and earlier, you could configure interinstance export through use of import routing table groups. A secondary routing instance would import routes from the primary routing instance. Then, IGPs would advertise the routes received from the second instance table as shown in the example in Figure 14 on page 181.

Figure 14: Nonforwarding Instance Concept



In JUNOS Release 5.4 and later, you can use the instance-import and instance-export policy keywords to perform nonforwarding, interinstance route sharing. The keywords are assigned at the [edit routing-instances *instance-name* routing-options] hierarchy level. These statements are similar to VRF import and VRF export policies used for VRF instances.

The "rt-export" module examines the from *instance* statements present in an instance import policy to construct the list of import tables for a particular exporting instance. The following example illustrates the configuration hierarchy for this feature:

```
[edit]
policy-options {
   policy-statement {
      red-import {
         from instance blue;
         then {
            tag 1;
            accept;
         }
      }
      blue-import {
```

```
            from instance red;
            then {
               tag 2;
               accept;
            }
         }
      }
   }
   routing-instances {
      red {
         routing-options {
            instance-import red-import;
         }
      }
      blue {
         routing-options {
            instance-import blue-import;
         }
      }
   }
```

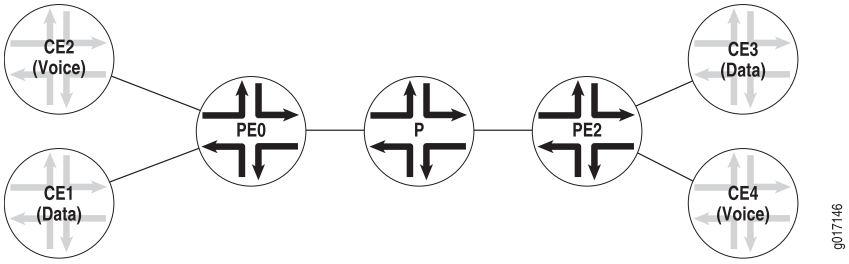
To advertise instance **blue** routes through an instance **red** IGP such as OSPF, you would add an export policy to OSPF to advertise routes from the local table.

```
[edit}
policy-options {
   policy-statement ospf-export {
      from /* some criteria */
      then accept;
   }
}
routing-instances {
   red {
      protocols ospf {
         export ospf-export;
      }
   }
}
```

When an instance import policy is configured, the policy is allowed to modify route attributes other than **next-hop**.

Example: Nonforwarding Instances Configuration

Figure 15: Nonforwarding Instances Topology Diagram



In Figure 15 on page 183, routers CE1, CE2, CE3, and CE4 are CE routers, PE0 and PE2 are PE routers, and Router P is the provider core transit router. CE1 and CE3 are part of a "community of interest" group called *data*, whereas CE2 and CE4 belong to a group called *voice*. Your goal is to connect the members of each group to each other by using a nonforwarding instance at the PE routers.

Note that routers PE0, CE1, and CE2 mirror the configurations on PE2, CE3, and CE4, respectively. Therefore, the latter routers are not shown in this example. The loopback addressing scheme for this network is shown in Table 7 on page 183.

Table 7: Nonforwarding Instances—Loopback Addresses

Router	Loopback Address
CE1	10.255.255.172
CE2	10.255.255.180
PE0	10.255.255.176
P	10.255.255.178
PE2	10.255.255.174
CE3	10.255.255.182
CE4	10.255.255.181

Routers CE1, CE2, CE3, and CE4 only need basic connectivity to their directly connected PE router. You enable OSPF on the interface that connects the CE routers to the PE routers. Since the configurations for all the CE routers are almost identical, only CE3 and CE4 are shown.

Router CE3

```
[edit]
protocols {
   ospf {
      area 0.0.0.0 {
         interface t3-0/0/0.0;
```

```
    }
  }
}
```

Router CE4

```
[edit]
protocols {
  ospf {
    area 0.0.0.0 {
      interface t3-0/0/2.0;
    }
  }
}
```

PE router configuration is next. Because the configurations for routers PE0 and PE2 mirror each other, only Router PE2 is displayed.

You must enable auto-export at the routing-options level for both the main configuration and the nonforwarding instances, establish policies that set tags on packets arriving from the CE routers, and accept packets into a specific instance that match the corresponding outbound tags. Specifically, you configure the router to attach a *data* tag to all packets coming from Router CE3 and a *voice* tag to all packets arriving from Router CE4. Also, forward any OSPF traffic coming from the core with a *data* tag to Router CE3, while OSPF core traffic with a *voice* tag is sent to Router CE4.

Router PE2

```
[edit]
routing-options {
  auto-export;
}
protocols {
  ospf {
    export [tag-voice tag-data];
    area 0.0.0.0 {
      interface t3-0/1/1.0;
    }
  }
}
routing-instances {
  data {
    instance-type no-forwarding;
    interface t3-0/1/3.0;
    routing-options {
      auto-export;
    }
    protocols {
      ospf {
        export import-data;
        area 0.0.0.0 {
          interface all;
        }
      }
    }
  }
  voice {
    instance-type no-forwarding;
    interface t3-0/1/0.0;
```

```
        routing-options {
            auto-export
        }
        protocols {
            ospf {
                export import-voice;
                area 0.0.0.0 {
                    interface all;
                }
            }
        }
    }
}
policy-options {
    policy-statement tag-voice {
        from instance voice;
        then {
            tag 11;
            accept;
        }
    }
    policy-statement tag-data {
        from instance data;
        then {
            tag 12;
            accept;
        }
    }
    policy-statement import-voice {
        from {
            instance master;
            protocol ospf;
            tag 11;
        }
        then accept;
    }
    policy-statement import-data {
        from {
            instance master;
            protocol ospf;
            tag 12;
        }
        then accept;
    }
}
```

On Router P, the provider core router configuration is simple. Include the interfaces that connect to the two PE routers (PE0 and PE2) in the OSPF process.

Router P

```
[edit]
protocols {
    ospf {
        area 0.0.0.0 {
            interface t1-0/1/1.0;
            interface t3-0/0/1.0;
        }
```

```
    }
  }
```

If all the configurations are correct, routers CE1 and CE3 (the *data* tagged routers) can send traffic to one another and routers CE2 and CE4 (the *voice* tagged routers) can communicate bidirectionally, but routers with different tag types cannot reach each other.

Verifying Your Work

To verify that the nonforwarding instances configuration is functioning properly, you can use the following commands:

- show ospf database
- show route detail
- ping

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

- Router PE2 Status on page 186
- Router CE3 Status on page 187

Router PE2 Status

```
user@PE2> show ospf database
    OSPF link state database, area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router *10.255.255.174    10.255.255.174    0x80000014   180  0x2  0x14b3  60
Router 10.255.255.176    10.255.255.176    0x80000010   592  0x2  0x14c1  60
Router 10.255.255.178    10.255.255.178    0x80000007  1074  0x2  0x9329  84
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Extern   10.255.255.172    10.255.255.176    0x8000000f   489  0x2  0xd258  36
Extern   10.255.255.180    10.255.255.176    0x8000000f   189  0x2  0x948d  36
Extern  *10.255.255.181    10.255.255.174    0x8000000f   780  0x2  0x968c  36
Extern  *10.255.255.182    10.255.255.174    0x8000000f   480  0x2  0x7aa8  36

user@PE2> show ospf database instance voice
    OSPF link state database, area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router   10.255.255.181    10.255.255.181    0x80000008  1112  0x2  0x29ac  60
Router  *192.255.197.117  192.255.197.117  0x8000000c  2681  0x2  0x5d7a  48
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Extern  *10.255.255.180    192.255.197.117  0x80000001  2681  0x2  0x5cf7  36

user@PE2> show ospf database instance data
    OSPF link state database, area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router   10.255.255.182    10.255.255.182    0x8000000b  1117  0x2  0x53d   60
Router  *192.255.197.249  192.255.197.249  0x8000000e  2686  0x2  0xbd05  48
    OSPF AS SCOPE link state database
```

```
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Extern  *10.255.255.172    192.255.197.249  0x80000002  2686  0x2  0x7d5a  36
```

Router CE3 Status

```
user@CE3> ping 10.255.255.172
PING 10.255.255.172 (10.255.255.172): 56 data bytes
64 bytes from 10.255.255.172: icmp_seq=0 ttl=252 time=2.978 ms
64 bytes from 10.255.255.172: icmp_seq=1 ttl=252 time=2.903 ms
^C
--- 10.255.255.172 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 2.903/2.941/2.978/0.037 ms

user@CE3> ping 10.255.255.180
PING 10.255.255.180 (10.255.255.180): 56 data bytes
^C
--- 10.255.255.180 ping statistics ---
2 packets transmitted, 0 packets received, 100% packet loss

user@CE3> show ospf database
    OSPF link state database, area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router  *10.255.255.182    10.255.255.182    0x8000000b  1164  0x2  0x53d   60
Router   192.255.197.249  192.255.197.249  0x8000000e  2735  0x2  0xbd05  48
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Extern   10.255.255.172    192.255.197.249  0x80000002  2735  0x2  0x7d5a  36

user@CE3> show route 10.255.255.172 detail

inet.0: 31 destinations, 32 routes (30 active, 0 holddown, 1 hidden)
10.255.255.172/32 (1 entry, 1 announced)
        *OSPF   Preference: 150
                Next hop: via t3-0/0/0.0, selected
                State: <Active Int Ext>
                Local AS:    69
                Age: 47:23      Metric: 2       Tag: 12
                Task: OSPF
                Announcement bits (1): 0-KRT
                AS path: I
```

For More Information

For additional information about interinstance route sharing, see the following resources:

- *JUNOS VPNs Configuration Guide*
- *JUNOS Routing Protocols Configuration Guide*

Revision History

13 April 2009—9.5R1 Release. Roy Spencer.

15 January 2009—9.4R1 Release. Roy Spencer.

10 October 2008—9.3R1 Release. Roy Spencer.

10 July 2008—9.2R1 Release. Roy Spencer.

10 April 2008—9.1R1 Release. Roy Spencer.

1 February 2008—9.0R1 Release. Fawn Damitio.

29 June 2007—8.4R1 Release. Fawn Damitio.

27 March 2007—8.3R1 Release. Fawn Damitio.

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

15 September 2006—8.1R1 Release. Richard Hendricks.

29 June 2006—8.0R1 Release. Richard Hendricks.

27 March 2006—7.6R1 Release. Richard Hendricks.

9 January 2006—7.5R1 Release. Richard Hendricks.

14 September 2005—7.4R1 Release. Richard Hendricks.

13 June 2005—7.3R1 Release. Richard Hendricks.

5 April 2005—7.2R1 Release. Richard Hendricks.

2 February 2005—7.1R1 Release. Richard Hendricks.

6 October 2004—7.0R1 Release. Richard Hendricks.

6 July 2004—6.4R1 Release. Richard Hendricks.

5 April 2004—6.3R1 Release. Richard Hendricks.

22 December 2003—6.2R1 Release. Richard Hendricks.

22 September 2003—6.1R1 Release. Richard Hendricks.

30 June 2003—6.0R1 Release. Richard Hendricks.

2 April 2003—5.7R1 Release. Richard Hendricks.

27 December 2002—5.6R1 Release. Richard Hendricks.

30 September 2002—5.5R1 Release. Richard Hendricks.

19 July 2002—5.4R1 Release. Richard Hendricks.

6 May 2002—Initial document written. Richard Hendricks.

Part 2
Routing Protocols

Chapter 7

Logical Systems

This feature guide chapter covers these topics:

Overview

For many years, engineers have combined power supplies, routing hardware and software, forwarding hardware and software, and physical interfaces into a networking device known as a router. Networking vendors have created large routers and small routers, but all routers have been placed into service as individual devices. As a result, the router has been considered a single physical device for most of its history.

The concept of logical systems breaks with this tradition. With JUNOS software, you can partition a single router into multiple logical devices that perform independent routing tasks. Because logical systems perform a subset of the tasks once handled by the main router, logical systems offer an effective way to maximize the use of a single routing or switching platform.

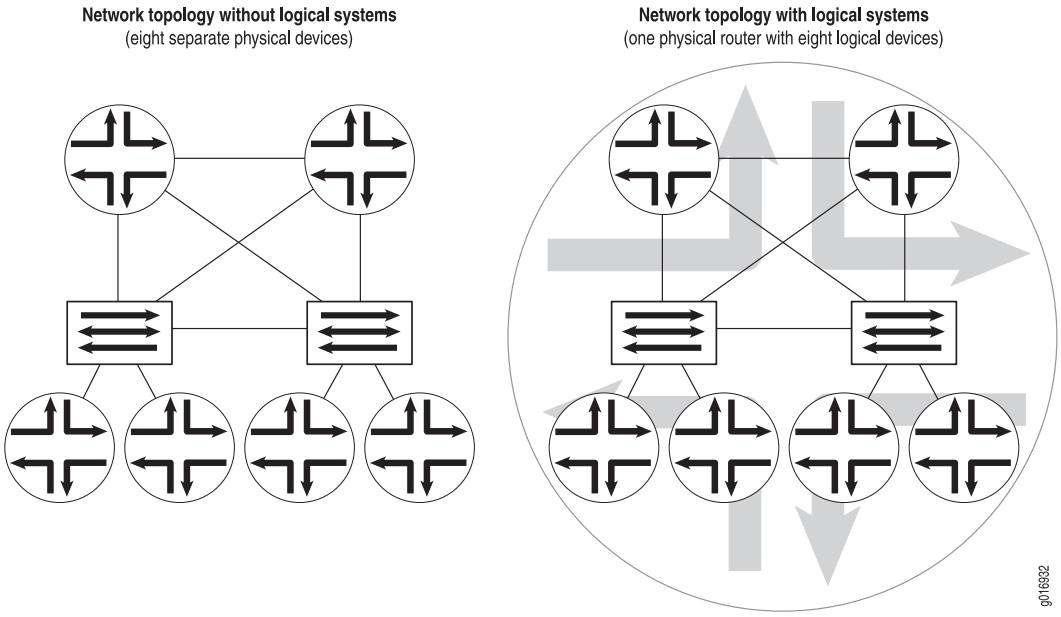
NOTE: Beginning with JUNOS software Release 9.3, the logical router feature has been renamed logical system.

All configuration statements, operational commands, show command outputs, error messages, log messages, and SNMP MIB objects that contain the string logical-router or logical-routers have been changed to logical-system and logical-systems, respectively.

Traditionally, service provider network design requires multiple layers of switches and routers. These devices transport packet traffic between customers. As seen on the left side of Figure 16 on page 192, access devices are connected to edge devices, which are in turn connected to core devices.

However, this complexity can lead to challenges in maintenance, configuration, and operation. To reduce such complexity, Juniper Networks support logical systems. Logical systems perform a subset of the actions of the main router and have their own unique routing tables, interfaces, policies, and routing instances. As shown on the right side of Figure 16 on page 192, a set of logical systems within a single router can handle the functions previously performed by several small routers.

Figure 16: Logical Systems Concept



The following protocols and functions are supported on logical systems:

- Open Shortest Path First (OSPF), Intermediate System-to-Intermediate System (IS-IS), Routing Information Protocol (RIP), RIP next generation (RIPng), Border Gateway Protocol (BGP), Resource Reservation Protocol (RSVP), Label Distribution Protocol (LDP), static routes, and Internet Protocol version 4 (IPv4) and version 6 (IPv6) are supported at the [edit logical-systems *logical-system-name* protocols] hierarchy level.
- Multiprotocol Label Switching (MPLS) provider edge (PE) and core provider router functions, such as Layer 2 virtual private networks (VPNs), Layer 3 VPNs, circuit cross-connect (CCC), Layer 2 circuits, and virtual private LAN service (VPLS) are supported at the [edit logical-systems *logical-system-name* routing-instances] hierarchy level.
- Multicast protocols, such as Protocol Independent Multicast (PIM) and Distance Vector Multicast Routing Protocol (DVMRP) are supported at the [edit logical-systems *logical-system-name* protocols] hierarchy level. Rendezvous point (RP) and source designated router (DR) functionality for multicast protocols within a logical system is also supported.
- All policy-related statements available at the [edit policy-options] hierarchy level are supported at the [edit logical-systems policy-options] hierarchy level.
- Most routing options statements available at the [edit routing-options] hierarchy level are supported at the [edit logical-systems routing-options] hierarchy level.
- Graceful Routing Engine switchover (GRES) is supported on J-series, M-series, MX-series, and T-series routers. For more information about GRES, see the *JUNOS High Availability Configuration Guide*.
- You can assign most interface types to a logical system, including SONET/SDH interfaces, Ethernet interfaces, Asynchronous Transfer Mode (ATM) interfaces, ATM2 intelligent queuing (IQ) interfaces, channelized IQ and Gigabit Ethernet IQ interfaces, aggregated interfaces, Link Services interfaces, and Multilink Services interfaces.
- Source class usage, destination class usage, unicast reverse-path forwarding, class of service, firewall filters, class-based forwarding, and policy-based accounting work with logical systems when you configure these features on the main router.
- Simple Network Management Protocol (SNMP) has been extended to support logical systems and routing instances. A network management system receives instance-aware information in the following format:

  logical-system-name/routing-instance@community

  As a result, a network manager can gather statistics for a specific community within a routing instance within a logical system. For more information on SNMP for logical systems, see the *JUNOS Network Management Configuration Guide*.
- SNMP support for logical systems and routing instances has been enhanced. The SNMP manager for a routing instance can now request and manage SNMP data only for that routing instance and other routing instances in the same logical system. As in previous releases, by default the SNMP manager for the default routing instance in the main router (inet.0) can access SNMP data from all routing instances. To restrict that manager's access to the default routing instance only,

include the routing-instance-access statement at the [edit snmp] hierarchy level. For more information, see the *JUNOS Network Management Configuration Guide*.

The following restrictions apply to logical systems:

- You can configure a maximum of 15 logical systems plus the master logical system on a router. When a configuration session is in use, users who are tied to the same logical system cannot commit configuration changes.
- The router has only one running configuration database, which contains configuration information for the main router and all associated logical systems. When configuring a logical system, users have their own candidate configuration database, which does not become part of the running configuration database until the user issues the commit statement.
- If a logical system experiences an interruption of its routing protocol process (rpd), the core dump output is placed in a file in the following location: /var/tmp/rpd_*logical-system-name*.core-tarball.*number*.tgz. Likewise, if you issue the restart routing command in a logical system, only the routing protocol process (rpd) for the logical system is restarted.
- If you configure trace options for a logical system, the output log file is stored in the following location: /var/log/*logical-system-name*. To monitor a log file within a logical system, issue the monitor start *logical-system-name/filename* command.
- The following Physical Interface Cards (PICs) are not supported with logical systems: Adaptive Services, ES, Monitoring Services, and Monitoring Services II.
- Generalized MPLS (GMPLS), IP Security (IPSec), point-to-multipoint label-switched paths (LSPs), port mirroring, and sampling are not supported.
- LSP ping and traceroute for autonomous system (AS) number lookup are not supported.
- Class of service (CoS) on logical tunnel (lt) or virtual loopback tunnel (vt) interfaces in a logical system is not supported.
- You cannot include the vrf-table-label statement on multiple logical systems if the core-facing interfaces are channelized or configured with multiple logical interfaces (Frame Relay DLCIs or Ethernet VLANs).
- The master administrator must configure global interface properties/physical interface properties at the [edit interfaces] hierarchy level. Logical system administrators can only configure and verify configurations for the logical systems to which they are assigned. For more information on configuring interfaces, see the *JUNOS Network Interfaces Configuration Guide*.

NOTE: A virtual router does not have the same capabilities as a logical system. A virtual router is a type of simplified routing instance that has a single routing table. A logical system is a partition of the main router and can contain multiple routing instances and routing tables. For example, a logical system can contain multiple virtual router routing instances. As a result, these two entities are not equivalent.

System Requirements

To implement logical systems, your system must meet these minimum requirements:

- JUNOS Release 8.5 or later for logical system administrator support.
- JUNOS Release 8.4 or later for SNMP enhancements and limits.
- JUNOS Release 8.3 or later for Bidirectional Forward (BFD) on logical systems.
- JUNOS Release 8.2 or later for support on MX-series routers
- JUNOS Release 7.5 or later for SNMP support within a logical system
- JUNOS Release 7.4 or later for multicast protocol RP and source DR functionality within a logical system
- JUNOS Release 7.0 or later to implement a logical tunnel (lt) interface on an integrated Adaptive Services Module in an M7i router
- JUNOS Release 6.1 or later, a Tunnel Services PIC, and an Enhanced FPC on M-series or T-series routing platforms to implement a logical tunnel (lt) interface
- JUNOS Release 6.0 or later for basic logical system functionality
- One or more Juniper Networks M-series, MX-series, or T-series routing platforms
- On M-series and T-series routers, a variety of PICs to assign interfaces to each logical system

Terms and Acronyms

L

logical system Segmentation of a system into multiple logical devices. Logical system configuration statements are found at the [edit logical-systems] hierarchy level.

logical system administrator JUNOS user with configuration and verification privileges for only the logical systems to which that user is assigned.

M

main router The standard concept of a router. Main router configuration statements are found at the [edit] hierarchy level.

master administrator JUNOS user with superuser configuration and verification privileges.

Configuring Logical Systems

To implement logical systems, you must configure the following:

Configuring Logical System Administrators (Master Administrator)

The master administrator can assign one or more logical system administrators to each logical system. Logical system administrators are confined to the context of the logical system to which they are assigned. This means that any global configuration statements are restricted from them. This also means that command output is restricted to the context to which the logical system administrators are assigned.

To configure logical system administrators, include the **logical-system** *logical-system-name* statement at the [**edit system login class** *class-name*] hierarchy level:

```
[edit]
system {
   login {
      class admin1 {
         permissions all;
         logical-system ls1;
      }
      class admin2 {
         permissions view; # Gives users assigned to class admin2 the ability to view
         logical-system ls2; # but not to change the configuration.
      }
      user user1 {
         class admin1;
      }
      user user2 {
         class admin2;
      }
   }
}
```

Configuring Logical System Interface Properties (Master Administrator)

Before adding interfaces to a logical system, the master administrator must configure physical interface properties on the main router. Common physical interface properties include encapsulation types and interface-related options.

To configure physical interface properties, the master administrator must include the statements at the [**edit interfaces** *interface-name*] hierarchy level.

Assigning Logical Interfaces to the Logical System (Master or Logical System Administrator)

After the interfaces are configured, the master administrator can assign logical interfaces to a logical system. To configure, include the **unit** statement at the [**edit logical-systems** *logical-system-name* **interfaces** *interface-name*] hierarchy level. Once you assign logical interfaces to a logical system, they are considered part of the logical system. Any logical interface can only be assigned one logical system, including the main router.

```
[edit]
logical-systems logical-system-name {
    interfaces {
        interface-name {
            logical-interface-statements;
            unit unit-number {
                family inet {
                    address ip-address;
                }
            }
        }
    }
    interfaces {
        interface-name {
            physical-interface-statements;
        }
    }
}
```

Configuring Protocols, Routing, and Policy Statements for the Logical System (Master or Logical System Administrator)

You can configure routing protocols (such as OSPF, BGP, and MPLS), policies (such as next-hop or load-balancing), routing options, and routing instances for a logical system.

To configure routing protocols, include the **protocols** statement at the [**edit logical-systems** *logical-system-name*] hierarchy level. To configure policies, include the **policy-options** statement at the [**edit logical-systems** *logical-system-name*] hierarchy level. To configure routing options, include the **routing-options** statement at the [**edit logical-systems** *logical-system-name*] hierarchy level. To configure routing instances, include the **routing-instances** statement at the [**edit logical-systems** *logical-system-name*] hierarchy level.

```
[edit]
logical-systems logical-system-name {
    protocols {
        ...
    }
    policy-options {
        ...
```

```
    }
    routing-options {
        ...
    }
    routing-instances {
        ...
    }
}
```

Configuring Other Logical System Statements

You can configure a variety of additional statements in conjunction with a logical system:

- Logical tunnel (lt) interface—You can connect different logical systems together within the same router with an lt interface. On M-series and T-series routing platforms, you can create an lt interface if you have a Tunnel Services PIC installed on an Enhanced FPC in your routing platform. On an M7i router, logical tunnel interfaces can be created by using the integrated Adaptive Services Module. On an MX-series router, the master administrator must configure logical tunnel interfaces by including the tunnel-services statement at the [edit chassis fpc *slot-number* pic *number*] hierarchy level. For more information about configuring tunnel interfaces on MX-series routers, see the *JUNOS System Basics Configuration* Guide.

  You must treat each interface like a point-to-point connection because you can only connect one logical tunnel interface to another at any given time. Also, you must select an interface encapsulation type, specify a DLCI number or VLAN identifier, configure a corresponding protocol family, and set the logical interface unit number of the peering lt interface. To configure, include the dlci, encapsulation, family, peer-unit,and vlan-id statements at the following hierarchy levels:

  - M-series, MX-series, or T-series router (master administrator only)—[edit interfaces lt-*fpc/pic/*0 unit *unit-number*]
  - Logical system—[edit logical-systems *logical-system-name* interfaces *lt-fpc/pic/*0 unit *unit-number*]

```
[edit]
logical-systems logical-system-name {
    interfaces {
        lt-fpc/pic/0 {
            unit unit-number {
                encapsulation (ethernet | ethernet-ccc | ethernet-vpls | frame-relay
                    |
                    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 master administrator. To configure, include a second IPv6 address with the address statement at the [edit interfaces lt-*fpc/pic/port* unit *unit-number* family inet6] hierarchy level. Link local addresses typically begin with the numbers fe80 (such as fe80::1111:1/64).

- Dynamic Host Control Protocol (DHCP) relay (master administrator only)—In a logical system, you can configure a DHCP or BOOTP server, and allow TFTP and DNS packets to be forwarded. To configure a DHCP or BOOTP server in a logical system, include the logical-system statement at the [edit forwarding-options helpers bootp interface *interface-name* server *ip-address*] hierarchy level. To configure TFTP packet forwarding in a logical system, include the logical-system statement at the [edit forwarding-options helpers tftp interface *interface-name* server *ip-address*] hierarchy level. To configure DNS packet forwarding in a logical system, include the logical-system statement at the [edit forwarding-options helpers domain interface *interface-name* server *ip-address*] hierarchy level. For more information about DHCP relay, BOOTP, TFTP, or DNS, see the *JUNOS Policy Framework Configuration Guide*.
- Filter-based forwarding (master administrator only)—You can configure filter-based forwarding for a logical system or a routing instance within a logical system. To configure filter-based forwarding for a logical system, include the logical-system statement at the [edit firewall filter *filter-name* term *term-name* then] hierarchy level. To configure filter-based forwarding for a routing instance within a logical system, include the routing-instance option at the [edit firewall filter *filter-name* term *term-name* then logical-system *logical-system-name*] hierarchy level. For more information, see the *JUNOS Policy Framework Configuration Guide*.
- Bidirectional forwarding—You can configure Bidirectional Forwarding Detection Protocol (BFD) for a logical system or a routing instance within a logical system. To configure BFD for a logical system, include the bfd-liveness-detection statement at the [edit logical-systems *logical-system-name* protocols] hierarchy level. To configure BFD for a routing instance within a logical system, include the bfd-liveness-detection statement at the [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols] hierarchy level. This feature is supported for the following protocols: RIP, BGP, OSPF, and IS-IS. For more information, see the *JUNOS Routing Protocols Configuration Guide*.
- You can place yourself into the context of a specific logical system. To configure a logical system context, issue the set cli logical-system *logical-system-name* command.

  When you enter logical system context mode and enter an operational mode command, the output of the command displays information related to the logical system only. For example, when you issue the show route command, the output shows only the routes that are assigned to the logical system.

  ```
  user@P0> set cli logical-system ls1
  Logical system: ls1
  ```

```
user@P0:ls1># Note that the user is now restricted to a logical system context.
```

To clear the logical system context and return to a full router (master router) context, issue the clear cli logical-system command.

```
user@P0:ls1> clear cli logical-system
Cleared default logical system
user@P0>#  Note that the user can now view the entire router again.
```

To achieve the same effect when using a JUNOScript client application, include the < set-logical-system > tag:

```
<rpc>
<set-logical-system>
<logical-system>ls1</logical-system>
</set-logical-system>
</rpc>
```

For more information about JUNOScript, see the *JUNOScript API Guide*.

- Enhanced SNMP support for logical systems (master administrator only)—By default, the SNMP manager can access SNMP data from all routing instances. To restrict the SNMP manager to data from the default routing instance only, include the routing-instance-access statement at the [edit snmp] hierarchy level. For more information about configuring SNMP, see the *JUNOS Network Management Configuration Guide*.
- SNMP community strings for routing instances and logical systems—To specify a routing instance when you add a client to an SNMP community, include the routing-instance *routing-instance-name* statement at the [edit snmp community *community-name*] hierarchy level. To specify a routing instance that is defined within a logical system, include the routing-instance routing-instance-name statement at the [edit snmp community community-name logical-system logical-system-name] hierarchy level.
- SNMPv3 trap targets for routing instances and logical systems—To specify a routing instance when you add a client to an SNMPv3 trap target, include the routing-instance *routing-instance-name* statement at the [edit snmp v3 target-address *target-address*] hierarchy level. To specify a logical system when you add a client in an SNMPv3 trap target, include the logical-system *logical-system-name* statement at the [edit snmp v3 target-address *target-address*] hierarchy level. For more information about configuring SNMP, see the *JUNOS Network Management Configuration Guide*.
- SNMP trap packets for routing instances—To specify a routing instance for SNMP trap packets sent by the router, include the routing-instance *routing-instance-name* statement at the [edit snmp trap-options] hierarchy level. To specify a routing instance that is defined within a logical system, include the routing-instance *routing-instance-name* statement at the [edit snmp trap-options logical-system *logical-system-name*] hierarchy level. For more information about configuring SNMP, see the *JUNOS Network Management Configuration Guide*.
- SNMP trap groups for routing instances and logical systems—To specify a routing instance of an SNMP trap group, include the routing-instance *routing-instance-name* statement at the [edit snmp trap-group *trap-group-name*] hierarchy level. To specify a logical system for an SNMP trap group, include the logical-system

logical-system-name statement at the [edit snmp trap-group *trap-group-name*] hierarchy level. For more information about configuring SNMP, see the *JUNOS Network Management Configuration Guide*.

In addition, you can configure only Frame Relay interface encapsulation on a logical tunnel interface when it is configured with an IPv6 address.

Example: Configuring Logical Systems

Figure 17: Logical System Topology Diagram

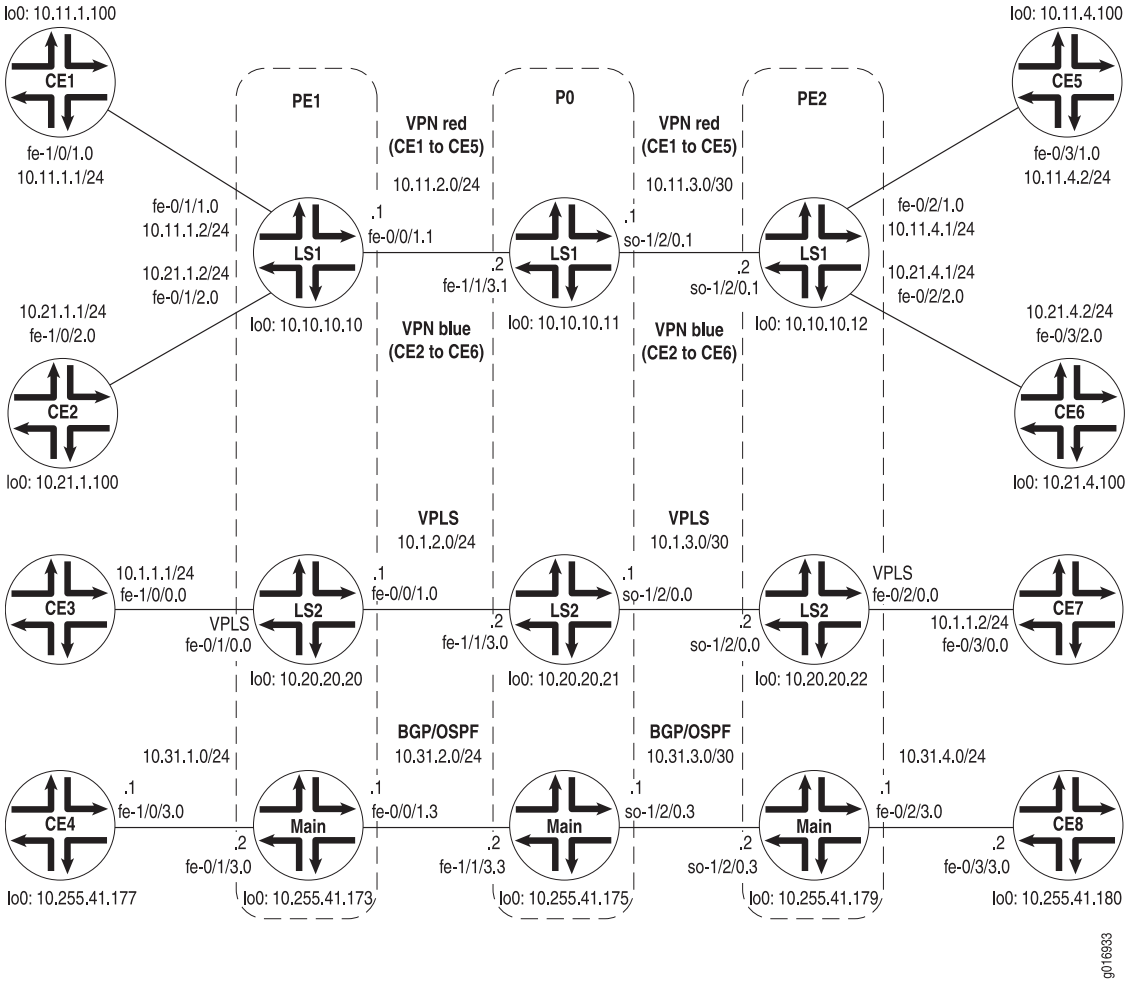


Figure 17 on page 201 shows four pairs of customer edge (CE) routers that are connected across an MPLS backbone. Routers CE1 and CE5 are part of the red VPN, routers CE2 and CE6 are in the blue VPN, routers CE3 and CE7 belong to a VPLS domain, and routers CE4 and CE8 are connected with standard protocols. Two logical systems are configured on provider edge (PE) routers PE1 and PE2 and provider core Router P0. Each of these three routers has two logical systems: LS1 and LS2. To illustrate the concept of a logical system, both VPNs are part of logical system LS1,

the VPLS instance belongs to LS2, and the remaining routers use the main router portion of routers PE1, P0, and PE2.

On Router CE1, configure OSPF to connect to the red VPN in logical system LS1 on Router PE1:

Router CE1

```
[edit]
interfaces {
    fe-1/0/1 {
        vlan-tagging;
        unit 0 {
            description "routing-instance red CE";
            vlan-id 101;
            family inet {
                address 10.11.1.1/24;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.11.1.100/32;
            }
        }
    }
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface fe-1/0/1.0;
            interface lo0.0;
        }
    }
}
```

On Router CE2, configure BGP to connect to the blue VPN in logical system LS1 on Router PE1:

Router CE2

```
[edit]
interfaces {
    fe-1/0/2 {
        vlan-tagging;
        unit 0 {
            description "routing-instance blue CE";
            vlan-id 102;
            family inet {
                address 10.21.1.1/24;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.21.1.100/32;
            }
        }
```

```
    }
  }
  routing-options {
    autonomous-system 200;
  }
  protocols {
    bgp {
      export export_loopback;
      group to_PE {
        type external;
        local-address 10.21.1.1;
        peer-as 100;
        neighbor 10.21.1.2;
      }
    }
  }
  policy-options {
    policy-statement export_loopback {
      from {
        route-filter 10.21.1.100/32 exact;
      }
      then accept;
    }
  }
```

On Router CE3, configure the Fast Ethernet interface in VLAN 600 to connect with the VPLS routing instance in logical system LS2 on Router PE1:

Router CE3

```
[edit]
interfaces {
  fe-1/0/0 {
    vlan-tagging;
    unit 0 {
      description "vpls interface";
      vlan-id 600;
      family inet {
        address 10.1.1.1/24;
      }
    }
  }
}
```

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

Router CE4

```
[edit]
interfaces {
  fe-1/0/3 {
    vlan-tagging;
    unit 0 {
      description "main router interface";
      vlan-id 103;
      family inet {
        address 10.31.1.1/24;
      }
    }
```

```
    }
    lo0 {
      unit 0 {
        family inet {
          address 10.255.41.177/32;
        }
      }
    }
  }
```

On Router PE1, create two VPN routing and forwarding (VRF) routing instances in logical system LS1: red and blue. Configure the CE-facing logical interfaces so that traffic from Router CE1 is placed in the red VPN and traffic from Router CE2 is placed in the blue VPN. Next, create a logical interface at fe-0/0/1.1 to connect to logical system LS1 on Router P0.

Also on Router PE1, create a VPLS routing instance in logical system LS2. Configure a logical interface so that traffic from Router CE3 is sent into the VPLS domain and connects to logical system LS2 on Router P0.

Create a logical system administrator for LS1. The logical system administrator can be responsible for the maintenance of this logical system.

Finally, configure a logical interface to interconnect Router CE4 with the main router portion of Router P0.

Router PE1

```
[edit]
logical-systems {
  ls1 { # The configuration for the first logical system begins here.
    interfaces {
      fe-0/0/1 {
        unit 1 { # This is the core-facing interface for logical system LS1.
          description "ls1 interface";
          vlan-id 101;
          family inet {
            address 10.11.2.1/24;
          }
          family iso;
          family mpls;
        }
      }
      fe-0/1/1 {
        unit 0 { # This logical interface connects to Router CE1.
          description "routing-instance red interface";
          vlan-id 101;
          family inet {
            address 10.11.1.2/24;
          }
        }
      }
      fe-0/1/2 {
        unit 0 { # This logical interface connects to Router CE2.
          description "routing-instance blue interface";
          vlan-id 102;
          family inet {
```

```
            address 10.21.1.2/24;
          }
        }
      }
      lo0 {
        unit 1 {
          description "ls1 loopback";
          family inet {
            address 10.10.10.10/32;
          }
          family iso {
            address 47.1111.1111.1111.1111.00;
          }
        }
      }
}
protocols { # You configure RSVP, MPLS, IS-IS, and BGP for logical system LS1.
    rsvp {
      interface all;
    }
    mpls  {
      label-switched-path to_10.10.10.12 {
        to 10.10.10.12;
      }
      interface all;
    }
    bgp {
      group to_other_PE {
        type internal;
        local-address 10.10.10.10;
        family inet-vpn {
          any;
        }
        neighbor 10.10.10.12;
      }
    }
    isis {
      interface all;
    }
}
policy-options {
    policy-statement from_bgp_to_ospf {
      then accept;
    }
}
routing-instances {
    blue {
      instance-type vrf; # You configure instance blue within logical system LS1.
      interface fe-0/1/2.0;
      route-distinguisher 10.10.10.10:200;
      vrf-target target:20:20;
      protocols {
        bgp {#BGP connects the blue instance with Router CE2.
          group to_CE {
            type external;
            local-address 10.21.1.2;
```

```
                peer-as 200;
                neighbor 10.21.1.1;
              }
            }
          }
        }
        red {
          instance-type vrf; # You configure instance red within logical system LS1.
          interface fe-0/1/1.0;
          route-distinguisher 10.10.10.10:100;
          vrf-target target:10:10;
          protocols {
            ospf {#OSPF connects the red instance with Router CE1.
              export from_bgp_to_ospf;
              area 0.0.0.0 {
                interface all;
              }
            }
          }
        }
      }
      routing-options {
        autonomous-system 100;
      }
    }
    ls2 {# The configuration for the second logical system begins here.
      interfaces {
        fe-0/0/1 {
          unit 0 {# This is the core-facing interface for logical system LS2.
            description "ls2 interface";
            vlan-id 100;
            family inet {
              address 10.1.2.1/24;
            }
            family iso;
            family mpls;
          }
        }
        fe-0/1/0 {
          unit 0 {# This logical interface connects to Router CE3.
            description "vpls interface";
            encapsulation vlan-vpls;
            vlan-id 600;
            family vpls;
          }
        }
        lo0 {
          unit 2 {
            description "ls2 loopback";
            family inet {
              address 10.20.20.20/32;
            }
            family iso {
              address 47.2222.2222.2222.2222.00;
            }
          }
```

```
        }
      }
      protocols { # You configure RSVP, MPLS, IS-IS, and BGP for logical system LS2.
        rsvp {
          interface all;
        }
        mpls {
          label-switched-path to_10.20.20.22 {
            to 10.20.20.22;
          }
          interface all;
        }
        bgp {
          group to_VPLS_PE {
            type internal;
            local-address 10.20.20.20;
            family l2vpn {
              signaling;
            }
            neighbor 10.20.20.22;
          }
        }
        isis {
          interface fe-0/0/1.0;
          interface lo0.2;
        }
      }
      routing-instances {
        new {
          instance-type vpls; # You configure VPLS within logical system LS2.
          interface fe-0/1/0.0;
          route-distinguisher 10.20.20.20:100;
          vrf-target target:30:30;
          protocols {
            vpls {
              site-range 10;
              site newPE {
                site-identifier 1;
              }
            }
          }
        }
      }
      routing-options {
        autonomous-system 400;
      }
    }
}
interfaces {
  fe-0/0/1 {
    vlan-tagging;
    unit 3 {# This is the core-facing interface for the main router of PE1.
      description "main router to P0";
      vlan-id 103;
      family inet {
        address 10.31.2.1/24;
```

```
        }
        family iso;
        family mpls;
      }
    }
    fe-0/1/3 {
      vlan-tagging;
      unit 0 {# This logical interface in the main router of PE1 connects to CE4.
        description "main router to CE4";
        vlan-id 103;
        family inet {
          address 10.31.1.2/24;
        }
      }
    }
    fe-0/1/0 {# You must always configure physical interface statements for
      vlan-tagging; # logical system interfaces at the [edit interfaces] hierarchy level.
      encapsulation vlan-vpls;
    }
    fe-0/1/1 {
      vlan-tagging;
    }
    fe-0/1/2 {
      vlan-tagging;
    }
    lo0 {
      unit 0 {
        description "main router loopback";
        family inet {
          address 10.255.41.173/32;
        }
      }
    }
}
routing-options {
    static {
      route 10.255.41.177/32 next-hop 10.31.1.1;
    }
    autonomous-system 500;
}
protocols {
    bgp {# The main router uses BGP as the exterior gateway protocol.
      group to_main_ls {
        type internal;
        local-address 10.255.41.173;
        export export_address;
        neighbor 10.255.41.179;
        neighbor 10.255.41.175;
      }
    }
    ospf {# The main router uses OSPF as the interior gateway protocol.
      area 0.0.0.0 {
        interface lo0.0;
        interface fe-0/0/1.3;
      }
    }
```

```
}
policy-options {
   policy-statement export_address {
      from {
         route-filter 10.255.41.177/32 exact;
      }
      then accept;
   }
}
system {
   login {
      class ls1–admin {
         permissions all;
         logical-system ls1;
      }
      user ls1–admin {
         class ls1–admin;
         authentication plain-text password;
         New password: password
         Retype new password: password
      }
   }
}
```

On Router P0, configure logical systems LS1, LS2, and the main router. For the logical system, you must configure physical interface properties at the main router [edit interfaces] hierarchy level and assign the logical interfaces to the logical systems. Next, you must configure protocols (such as RSVP, MPLS, BGP, and IS-IS), routing options, and policy options for the logical systems. Last, configure the same logical system administrator for LS1 that you configured on Router PE1. Configure this same logical system administrator for LS2 to have permission to view the LS2 configuration, but not change the configuration for LS2.

In this example, logical system LS1 transports traffic for the red VPN that exists between routers CE1 and CE5. LS1 also connects the blue VPN that exists between routers CE2 and CE6. Logical system LS2 transports VPLS traffic between routers CE3 and CE7.

For the main router on Router P0, you can configure the router as usual. In this example, the main router transports traffic between routers CE4 and CE8. As a result, configure the interfaces and routing protocols (OSPF, BGP) to connect to the main router portion of routers PE1 and PE2.

Router P0

```
[edit]
logical-systems {
   ls1 { # The configuration for the first logical system begins here.
      interfaces {
         fe-1/1/3 {
            unit 1 { # This logical interface connects to LS1 on Router PE1.
               description "ls1 interface";
               vlan-id 101;
               family inet {
                  address 10.11.2.2/24;
               }
               family iso;
```

```
            family mpls;
        }
    }
    so-1/2/0 {
        unit 1 { # This logical interface connects to LS1 on Router PE2.
            description "ls1 interface";
            dlci 101;
            family inet {
                address 10.11.3.1/24;
            }
            family iso;
            family mpls;
        }
    }
    lo0 {
        unit 1 {
            description "ls1 loopback";
            family inet {
                address 10.10.10.11/32;
            }
            family iso {
                address 47.1111.1111.1111.1112.00;
            }
        }
    }
}
protocols { # You configure RSVP, MPLS, and IS-IS for logical system LS1.
    rsvp {
        interface all;
    }
    mpls {
        interface all;
    }
    isis {
        interface all;
    }
}
}
ls2 {# The configuration for the second logical system begins here.
    interfaces {
        fe-1/1/3 {
            unit 0 {# This logical interface connects to LS2 on Router PE1.
                description "ls2 interface";
                vlan-id 100;
                family inet {
                    address 10.1.2.2/24;
                }
                family iso;
                family mpls;
            }
        }
        so-1/2/0 {
            unit 0 {# This logical interface connects to LS2 on Router PE2.
                description "ls2 interface";
                dlci 100;
                family inet {
```

```
                address 10.1.3.1/24;
              }
              family iso;
              family mpls;
            }
          }
          lo0 {
            unit 2 {
              description "ls2 loopback";
              family inet {
                address 10.20.20.21/32;
              }
              family iso {
                address 47.2222.2222.2222.2223.00;
              }
            }
          }
        }
        protocols { # You configure RSVP, MPLS, and IS-IS for logical system LS2.
          rsvp  {
            interface all;
          }
          mpls {
            interface all;
          }
          isis {
            interface fe-1/1/3.0;
            interface so-1/2/0.0;
            interface lo0.2;
          }
        }
      }
    }
    interfaces {
      fe-1/1/3 {
        vlan-tagging;
        unit 3 {# This logical interface connects to the main router on Router PE1.
          description "main router interface";
          vlan-id 103;
          family inet {
            address 10.31.2.2/24;
          }
          family iso;
          family mpls;
        }
      }
      so-1/2/0 {
        dce; # You must configure all physical interface statements for logical
        encapsulation frame-relay; # routers at the [edit interfaces] hierarchy level.
        unit 3 { # This logical interface connects to the main router on Router PE2.
          description "main router interface";
          dlci 103;
          family inet {
            address 10.31.3.1/24;
          }
          family iso;
```

```
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            description "main router loopback";
            family inet {
                address 10.255.41.175/32;
            }
        }
    }
}
routing-options {
    autonomous-system 500;
}
protocols { # You configure BGP and OSPF for the main router.
    bgp {
        group to_main_ls {
            type internal;
            local-address 10.255.41.175
            neighbor 10.255.41.179;
            neighbor 10.255.41.173;
        }
    }
    ospf {
        area 0.0.0.0 {
            interface lo0.0;
            interface fe-1/1/3.3;
            interface so-1/2/0.3;
        }
    }
}
system {
    login {
        class ls1–admin {
            permissions all;
            logical-system ls1;
        }
        class ls1–onlooker {
            permissions view;
            logical-system ls2;
        }
        user ls1–admin {
            class ls1–admin;
        }
    }
}
```

On Router PE2, create two VRF routing instances in logical system LS1: red and blue. Configure the CE-facing logical interfaces so that traffic from Router CE5 is placed in the red VPN and traffic from Router CE6 is placed in the blue VPN. Next, create one logical interface on so-1/2/0.1 to connect to logical system LS1 on Router P0.

Also on Router PE2, create a VPLS routing instance in logical system LS2. Configure a logical interface so that traffic from Router CE7 is sent into the VPLS domain and connects to logical system LS2 on Router P0.

Configure a logical interface to interconnect Router CE8 with the main router portion of Router P0.

Finally, you can optionally create a logical system administrator that has configuration privileges for LS1 and viewing privileges for LS2.

Router PE2

```
[edit]
logical-systems {
   ls1 { # The configuration for the first logical system begins here.
      interfaces {
         fe-0/2/0 {
            unit 1 { # This logical interface connects to Router CE5.
               description "routing-instance red interface";
               vlan-id 101;
               family inet {
                  address 10.11.4.1/24;
               }
            }
            unit 2 { # This logical interface connects to Router CE6.
               description "routing-instance blue interface";
               vlan-id 102;
               family inet {
                  address 10.21.4.1/24;
               }
            }
         }
         so-1/2/0 {
            unit 1 {# This is the core-facing interface for logical system LS1.
               description "ls1 interface";
               dlci 101;
               family inet {
                  address 10.11.3.2/24;
               }
               family iso;
               family mpls;
            }
         }
         lo0 {
            unit 1 {
               description "ls1 loopback";
               family inet {
                  address 10.10.10.12/32;
               }
               family iso {
                  address 47.1111.1111.1111.1113.00;
               }
            }
         }
      }
      protocols {
         rsvp {# You configure RSVP, MPLS, IS-IS, and BGP for logical system LS1.
```

```
        interface all;
    }
    mpls {
        label-switched-path to_10.10.10.10 {
            to 10.10.10.10;
        }
        interface all;
    }
    bgp {
        group to_other_PE {
            type internal;
            local-address 10.10.10.12;
            family inet {
                any;
            }
            family inet-vpn {
                any;
            }
            neighbor 10.10.10.10;
        }
    }
    isis {
        interface all;
    }
}
policy-options {
    policy-statement from_bgp_to_ospf {
        then accept;
    }
}
routing-instances {
    blue {
        instance-type vrf; # You configure instance blue within logical system LS1.
        interface fe-0/2/2.0;
        route-distinguisher 10.10.10.12:200;
        vrf-target target:20:20;
        protocols {
            bgp { # BGP connects the blue instance with Router CE6.
                group to_CE {
                    local-address 10.21.4.1;
                    peer-as 300;
                    neighbor 10.21.4.2;
                }
            }
        }
    }
    red {
        instance-type vrf; # You configure instance red within logical system LS1.
        interface fe-0/2/1.0;
        route-distinguisher 10.10.10.12:100;
        vrf-target target:10:10;
        protocols {
            ospf { # OSPF connects the red instance with Router CE5.
                export from_bgp_to_ospf;
                area 0.0.0.0 {
                    interface all;
```

```
            }
          }
        }
      }
    }
    routing-options {
      autonomous-system 100;
    }
}
logical-systems {
    ls2 { # The configuration for the second logical system begins here.
      interfaces {
        fe-0/2/0 {
          unit 0 { # This logical interface connects to Router CE7.
            description "vpls interface";
            encapsulation vlan-vpls;
            vlan-id 600;
            family vpls;
          }
        }
        so-1/2/0 {
          unit 0 { # This is the core-facing interface for logical system LS2.
            description "ls2 interface";
            dlci 100;
            family inet {
              address 10.1.3.2/24;
            }
            family iso;
            family mpls;
          }
        }
        lo0 {
          unit 2 {
            description "ls2 loopback";
            family inet {
              address 10.20.20.22/32;
            }
            family iso {
              address 47.2222.2222.2222.2224.00;
            }
          }
        }
      }
    protocols { # You configure RSVP, MPLS, IS-IS, and BGP for logical system LS2.
      rsvp {
        interface all;
      }
      mpls {
        label-switched-path to_10.20.20.20 {
          to 10.20.20.20;
        }
        interface all;
      }
      bgp {
        group to_VPLS_PE {
          type internal;
```

```
            local-address 10.20.20.22;
            family l2vpn {
               signaling;
            }
            neighbor 10.20.20.20;
         }
      }
      isis {
         interface so-1/2/0.0;
         interface lo0.2;
      }
   }
   routing-instances {
      new {
         instance-type vpls; # You configure VPLS within logical system LS2.
         interface fe-0/2/0.0;
         route-distinguisher 10.20.20.22:100;
         vrf-target target:30:30;
         protocols {
            vpls {
               site-range 10;
               site newPE {
                  site-identifier 2;
               }
            }
         }
      }
   }
   routing-options {
      autonomous-system 400;
   }
}
interfaces {
   fe-0/2/0 {   # You must always configure physical interface statements for the
      vlan-tagging; # logical system interfaces at the [edit interfaces] hierarchy level.
      encapsulation vlan-vpls;
   }
   fe-0/2/1 {
      vlan-tagging;
   }
   fe-0/2/2 {
      vlan-tagging;
   }
   fe-0/2/3 {
      vlan-tagging;
      unit 0 { # This logical interface in the main router of PE2 connects to CE8.
         description "main router to CE8";
         vlan-id 103;
         family inet {
            address 10.31.4.1/24;
         }
      }
   }
   so-1/2/0 {
      encapsulation frame-relay;
      unit 3 { # This is the core-facing interface for the main router of PE2.
```

```
            description "main router to P0";
            dlci 103;
            family inet {
               address 10.31.3.2/24;
            }
            family iso;
            family mpls;
         }
      }
      lo0 {
         unit 0 {
            description "main router loopback";
            family inet {
               address 10.155.41.179/32;
            }
         }
      }
   }
   routing-options {
      static {
         route 10.255.41.180/32 next-hop 10.31.4.2;
      }
      autonomous-system 500;
   }
   protocols {
      bgp {# The main router uses BGP as the exterior gateway protocol.
         group to_main_ls {
            type internal;
            local-address 10.255.41.179;
            export export_address;
            neighbor 10.255.41.173;
            neighbor 10.255.41.175;
         }
      }
      ospf {# The main router uses OSPF as the interior gateway protocol.
         area 0.0.0.0 {
            interface so-1/2/0.3;
            interface fe-0/2/3.0;
            interface lo0.0;
         }
      }
   }
   policy-options {
      policy-statement export_address {
         from {
            route-filter 10.255.41.180/32 exact;
         }
         then accept;
      }
   }
}
system {
   login {
      class ls1–admin {
         permissions all;
         logical-system ls1;
```

```
        }
        class ls1–onlooker {
            permissions view;
            logical-system ls2;
        }
        user ls1–admin {
            class ls1–admin;
        }
    }
}
```

On Router CE5, configure OSPF to connect to the red VPN in logical system LS1 on Router PE2:

Router CE5

```
[edit]
interfaces {
    fe-0/3/1 {
        vlan-tagging;
        unit 0 {
            description "routing-instance red CE";
            vlan-id 101;
            family inet {
                address 10.11.4.2/24;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.11.4.100/32;
            }
        }
    }
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface fe-0/3/1.0;
            interface lo0.0;
        }
    }
}
system {
    login {
        class ls1–admin {
            permissions all;
            logical-system ls1;
        }
        class ls1–onlooker {
            permissions view;
            logical-system ls2;
        }
        user ls1–admin {
            class ls1–admin;
        }
    }
```

```
}
```

On Router CE6, configure BGP to connect to the blue VPN in logical system LS1 on Router PE2:

Router CE6

```
[edit]
interfaces {
    fe-0/3/2 {
        vlan-tagging;
        unit 0 {
            description "routing-instance blue CE";
            vlan-id 102;
            family inet {
                address 10.21.4.2/24;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.21.4.100/32;
            }
        }
    }
}
routing-options {
    autonomous-system 300;
}
protocols {
    bgp {
        export export_loopback;
        group to_PE {
            type external;
            local-address 10.21.4.2;
            peer-as 100;
            neighbor 10.21.4.1;
        }
    }
}
policy-options {
    policy-statement export_loopback {
        from {
            route-filter 10.21.4.100/32 exact;
        }
        then accept;
    }
}
```

On Router CE7, configure the Fast Ethernet interface in VLAN 600 to connect with the VPLS routing instance in logical system LS2 on Router PE2:

Router CE7

```
[edit]
interfaces {
    fe-0/3/0 {
        vlan-tagging;
        unit 0 {
```

```
        description "vpls interface";
        vlan-id 600;
        family inet {
          address 10.1.1.2/24;
        }
      }
    }
  }
```

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

Router CE8

```
[edit]
interfaces {
  fe-0/3/3 {
    vlan-tagging;
    unit 0 {
      description "main router interface";
      vlan-id 103;
      family inet {
        address 10.31.4.2/24;
      }
    }
  }
}
lo0 {
  unit 0 {
    family inet {
      address 10.255.41.180/32;
    }
  }
}
```

Verifying Your Work

To verify the proper operation of logical systems, the master administrator can use the following commands:

- show bgp summary (logical-system *logical-system-name*)
- show isis adjacency (logical-system *logical-system-name*)
- show mpls lsp (logical-system *logical-system-name*)
- show ospf neighbor (logical-system *logical-system-name*)
- show route (logical-system *logical-system-name*)
- show route protocol protocol (logical-system *logical-system-name*)
- show rsvp session (logical-system *logical-system-name*)
- logical-system (*logical-system-name*)

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

Router CE1 Status

```
user@CE1> show route table
inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.11.1.0/24       *[Direct/0] 00:20:20
                    > via fe-1/0/1.0
10.11.1.1/32       *[Local/0] 00:20:24
                      Local via fe-1/0/1.0
10.11.1.100/32     *[Direct/0] 00:21:53
                    > via lo0.0
10.11.4.0/24       *[OSPF/150] 00:18:30, metric 0, tag 3489661028
                    > to 10.11.1.2 via fe-1/0/1.0
10.11.4.100/32     *[OSPF/10] 00:18:30, metric 2
                    > to 10.11.1.2 via fe-1/0/1.0
224.0.0.5/32       *[OSPF/10] 00:21:58, metric 1
                      MultiRecv
```

Router CE2 Status

```
user@CE2> show route table
inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.21.1.0/24       *[Direct/0] 00:20:30
```

```
                    > via fe-1/0/2.0
10.21.1.1/32       *[Local/0] 00:20:34
                      Local via fe-1/0/2.0
10.21.1.100/32     *[Direct/0] 00:22:03
                    > via lo0.0
10.21.4.0/24       *[BGP/170] 00:18:43, localpref 100
                      AS path: 100 I
                    > to 10.21.1.2 via fe-1/0/2.0
10.21.4.100/32     *[BGP/170] 00:18:43, localpref 100
                      AS path: 100 300 I
                    > to 10.21.1.2 via fe-1/0/2.0
```

Router CE3 Status

```
user@CE3> show route table
inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.1.0/24       *[Direct/0] 00:20:13
                    > via fe-1/0/0.0
10.1.1.1/32       *[Local/0] 00:20:17
                       Local via fe-1/0/0.0
```

Router PE1 Status: Main Router

```
user@PE1> show bgp summary

Groups: 1 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 1          0          0          0          0          0
Peer               AS      InPkt     OutPkt    OutQ   Flaps Last Up/DwnState|#Active/Received/Damped...
10.255.41.175     500          5          8       0       0        2:31 0/0/0                0/0/0
10.255.41.179     500          6          9       0       0        2:35 0/1/0                0/0/0
```

```
user@PE1>  show route protocol bgp
inet.0: 20 destinations, 21 routes (20 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.255.41.180/32    [BGP/170] 00:02:48, localpref 100, from 10.255.41.179
                      AS path: I
                    > to 10.31.2.2 via fe-0/0/1.3
iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

user@PE1>  show ospf neighbor
  Address         Interface             State      ID              Pri  Dead
10.31.2.2        fe-0/0/1.3            Full      10.255.41.175    128   32

user@PE1>  show isis adjacency
IS-IS instance is not running
```

Router PE1 Status: LS1

The master administrator can issue the following command to view the output for a specific logical system.

```
user@PE1> show bgp summary logical-system ls1

Groups: 2 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l3vpn.0            4          4          0          0          0          0
bgp.l3vpn.2            0          0          0          0          0          0
Peer               AS      InPkt     OutPkt    OutQ   Flaps Last Up/DwnState|#Active/Received/Damped...
10.10.10.12       100         13         14       0       0        2:50 Establ
  bgp.l3vpn.0: 4/4/0
  bgp.l3vpn.2: 0/0/0
  blue.inet.0: 2/2/0
  red.inet.0: 2/2/0
10.21.1.1         200         13         14       0       0        4:33 Establ
  blue.inet.0: 1/1/0
```

The logical system administrator for LS1 will see the same output when the following command is issued.

```
ls1-admin@PE1:ls1> show bgp summary

Groups: 2 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l3vpn.0            4          4          0          0          0          0
bgp.l3vpn.2            0          0          0          0          0          0
Peer               AS      InPkt     OutPkt    OutQ   Flaps Last Up/DwnState|#Active/Received/Damped...
10.10.10.12       100         13         14       0       0        2:50 Establ
  bgp.l3vpn.0: 4/4/0
  bgp.l3vpn.2: 0/0/0
  blue.inet.0: 2/2/0
  red.inet.0: 2/2/0
10.21.1.1         200         13         14       0       0        4:33 Establ
  blue.inet.0: 1/1/0
```

red VPN The master administrator can issue the following command to view the output for a specific logical system.

```
user@PE1> show route logical-system ls1 table red
red.inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.11.1.0/24       *[Direct/0] 00:04:51
                    > via fe-0/1/1.0
10.11.1.2/32       *[Local/0] 00:05:45
                      Local via fe-0/1/1.0
10.11.1.100/32     *[OSPF/10] 00:04:02, metric 1
                    > to 10.11.1.1 via fe-0/1/1.0
10.11.4.0/24       *[BGP/170] 00:03:05, localpref 100, from 10.10.10.12
                      AS path: I
                    > to 10.11.2.2 via fe-0/0/1.1, label-switched-path
to_10.10.10.12
```

```
10.11.4.100/32     *[BGP/170] 00:03:05, MED 1, localpref 100, from 10.10.10.12
                      AS path: I
                    > to 10.11.2.2 via fe-0/0/1.1, label-switched-path
to_10.10.10.12
224.0.0.5/32       *[OSPF/10] 00:07:02, metric 1
                      MultiRecv
```

The logical system administrator for LS1 will see the same output when the following command is issued.

```
ls1-admin@PE1:ls1> show route table red
red.inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.11.1.0/24       *[Direct/0] 00:04:51
                    > via fe-0/1/1.0
10.11.1.2/32       *[Local/0] 00:05:45
                      Local via fe-0/1/1.0
10.11.1.100/32     *[OSPF/10] 00:04:02, metric 1
                    > to 10.11.1.1 via fe-0/1/1.0
10.11.4.0/24       *[BGP/170] 00:03:05, localpref 100, from 10.10.10.12
                      AS path: I
                    > to 10.11.2.2 via fe-0/0/1.1, label-switched-path
to_10.10.10.12
10.11.4.100/32     *[BGP/170] 00:03:05, MED 1, localpref 100, from 10.10.10.12
                      AS path: I
                    > to 10.11.2.2 via fe-0/0/1.1, label-switched-path
to_10.10.10.12
224.0.0.5/32       *[OSPF/10] 00:07:02, metric 1
                      MultiRecv
```

blue VPN The master administrator can issue the following command to view the output for a specific logical system.

```
user@PE1> show route logical-system ls1 table blue
blue.inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.21.1.0/24       *[Direct/0] 00:05:29
                    > via fe-0/1/2.0
10.21.1.2/32       *[Local/0] 00:06:23
                      Local via fe-0/1/2.0
10.21.1.100/32     *[BGP/170] 00:05:26, localpref 100
                      AS path: 200 I
                    > to 10.21.1.1 via fe-0/1/2.0
10.21.4.0/24       *[BGP/170] 00:03:43, localpref 100, from 10.10.10.12
                      AS path: I
                    > to 10.11.2.2 via fe-0/0/1.1, label-switched-path
to_10.10.10.12
10.21.4.100/32     *[BGP/170] 00:03:43, localpref 100, from 10.10.10.12
                      AS path: 300 I
                    > to 10.11.2.2 via fe-0/0/1.1, label-switched-path
to_10.10.10.12

user@PE1>  show route logical-system ls1 table inet.0
inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.10.10/32     *[Direct/0] 00:08:05
```

```
                    > via lo0.1
10.10.10.11/32      *[IS-IS/15] 00:05:07, metric 10
                    > to 10.11.2.2 via fe-0/0/1.1
10.10.10.12/32      *[IS-IS/15] 00:04:58, metric 20
                    > to 10.11.2.2 via fe-0/0/1.1
10.11.2.0/24        *[Direct/0] 00:05:38
                    > via fe-0/0/1.1
10.11.2.1/32        *[Local/0] 00:06:51
                      Local via fe-0/0/1.1
10.11.3.0/24        *[IS-IS/15] 00:05:07, metric 20
                    > to 10.11.2.2 via fe-0/0/1.1

user@PE1>  ping logical-system ls1 routing-instance red 10.11.4.100
PING 10.11.4.100 (10.11.4.100): 56 data bytes
64 bytes from 10.11.4.100: icmp_seq=0 ttl=251 time=1.055 ms
^C
--- 10.11.4.100 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.055/1.055/1.055/0.000 ms
```

The logical system administrator for LS1 will see the same output when they issue the following command.

```
ls1-admin@PE1:ls1> show route table blue
blue.inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.21.1.0/24        *[Direct/0] 00:05:29
                    > via fe-0/1/2.0
10.21.1.2/32        *[Local/0] 00:06:23
                      Local via fe-0/1/2.0
10.21.1.100/32      *[BGP/170] 00:05:26, localpref 100
                      AS path: 200 I
                    > to 10.21.1.1 via fe-0/1/2.0
10.21.4.0/24        *[BGP/170] 00:03:43, localpref 100, from 10.10.10.12
                      AS path: I
                    > to 10.11.2.2 via fe-0/0/1.1, label-switched-path
to_10.10.10.12
10.21.4.100/32      *[BGP/170] 00:03:43, localpref 100, from 10.10.10.12
                      AS path: 300 I
                    > to 10.11.2.2 via fe-0/0/1.1, label-switched-path
to_10.10.10.12

ls1-admin@PE1:ls1>  show route table inet.0
inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.10.10/32      *[Direct/0] 00:08:05
                    > via lo0.1
10.10.10.11/32      *[IS-IS/15] 00:05:07, metric 10
                    > to 10.11.2.2 via fe-0/0/1.1
10.10.10.12/32      *[IS-IS/15] 00:04:58, metric 20
                    > to 10.11.2.2 via fe-0/0/1.1
10.11.2.0/24        *[Direct/0] 00:05:38
                    > via fe-0/0/1.1
10.11.2.1/32        *[Local/0] 00:06:51
                      Local via fe-0/0/1.1
10.11.3.0/24        *[IS-IS/15] 00:05:07, metric 20
                    > to 10.11.2.2 via fe-0/0/1.1
```

```
ls1-admin@PE1:ls1>  ping routing-instance red 10.11.4.100
PING 10.11.4.100 (10.11.4.100): 56 data bytes
64 bytes from 10.11.4.100: icmp_seq=0 ttl=251 time=1.055 ms
^C
--- 10.11.4.100 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.055/1.055/1.055/0.000 ms
```

Router PE1 Status: LS2

```
user@PE1> show vpls connections logical-system ls2
Layer-2 VPN Connections:

Legend for connection status (St)
OR -- out of range              WE -- intf encaps != instance encaps
EI -- encapsulation invalid     Dn -- down
EM -- encapsulation mismatch    VC-Dn -- Virtual circuit down
CM -- control-word mismatch     -> -- only outbound conn is up
CN -- circuit not provisioned   <- -- only inbound conn is up
OL -- no outgoing label         Up -- operational
NC -- intf encaps not CCC/TCC   XX -- unknown
NP -- intf h/w not present

Legend for interface status
Up -- operational
Dn -- down

Instance: new
Local site: newPE (1)
    connection-site           Type  St     Time last up          # Up trans
    2                         rmt   Up     Jul 16 14:05:25 2003           1
      Local interface: vt-1/2/0.49152, Status: Up, Encapsulation: VPLS
      Remote PE: 10.20.20.22, Negotiated control-word: No
      Incoming label: 800001, Outgoing label: 800000
```

Router P0 Status: Main Router

```
user@P0> show interfaces terse lo0
Interface               Admin Link Proto Local                 Remote
lo0                     up    up
lo0.0                   up    up   inet  10.255.41.175       --> 0/0
                                        127.0.0.1           --> 0/0
                                   iso
47.0005.80ff.f800.0000.0108.0003.0102.5501.4175.00
                                   inet6 fe80::2a0:a5ff:fe12:2b09
                                         feee::10:255:14:175
lo0.1                   up    up   inet  10.10.10.11         --> 0/0
                                   iso   47.1111.1111.1111.1112.00
lo0.2                   up    up   inet  10.20.20.21         --> 0/0
                                   iso   47.2222.2222.2222.2223.00
lo0.16383               up    up   inet
user@P0> show ospf neighbor
  Address         Interface             State      ID              Pri  Dead
10.31.2.1        fe-1/1/3.3            Full      10.255.41.173    128   34
10.31.3.2        so-1/2/0.3            Full      10.255.41.179    128   37
```

Router P0 Status: LS1

```
user@P0> show isis adjacency logical-system ls1
Interface             System         L State       Hold (secs) SNPA
fe-1/1/3.1            PE1         2 Up                    21  0:90:69:9:4:1
fe-1/1/3.1            PE1         1 Up                    24  0:90:69:9:4:1
so-1/2/0.1            PE2          3 Up                    25

user@P0>  show bgp summary logical-system ls1
BGP is not running

user@P0>  show route protocol isis logical-system ls1
inet.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.10.10.10/32     *[IS-IS/15] 00:09:15, metric 10
                    > to 10.11.2.1 via fe-1/1/3.1
10.10.10.12/32     *[IS-IS/15] 00:09:39, metric 10
                    > to 10.11.3.2 via so-1/2/0.1

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

mpls.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
```

Router P0 Status: LS2

```
user@P0> show bgp summary logical-system ls2
BGP is not running

user@P0>  show isis adjacency logical-system ls2
Interface             System         L State       Hold (secs) SNPA
fe-1/1/3.0            PE1            2 Up                    24  0:90:69:9:4:1
fe-1/1/3.0            PE1            1 Up                    23  0:90:69:9:4:1
so-1/2/0.0            PE2            3 Up                    24

user@P0>  show route protocol isis logical-system ls2

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

10.20.20.20/32     *[IS-IS/15] 00:09:44, metric 10
                    > to 10.1.2.1 via fe-1/1/3.0
10.20.20.22/32     *[IS-IS/15] 00:09:45, metric 10
                    > to 10.1.3.2 via so-1/2/0.0

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

mpls.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
```

Router PE2 Status: Main Router

```
user@PE2> show ospf neighbor
```

```
                         Address         Interface             State      ID              Pri  Dead
                       10.31.4.2         fe-0/2/3.0            Full       10.255.41.180   128   38
                       10.31.3.1         so-1/2/0.3            Full       10.255.41.175   128   36

user@PE2>  show interfaces terse lo0
Interface              Admin Link Proto Local                 Remote
lo0                    up    up
lo0.0                  up    up   inet  10.255.41.179       --> 0/0
                                       127.0.0.1           --> 0/0
                                  iso   47.0005.80ff.f800.0000.0108.0003.0102.5501.4179.00
                                  inet6 fe80::2a0:a5ff:fe12:29ff
                                        feee::10:255:14:179
lo0.1                  up    up   inet  10.10.10.12         --> 0/0
                                  iso   47.1111.1111.1111.1113.00
lo0.2                  up    up   inet  10.20.20.22         --> 0/0
                                  iso   47.2222.2222.2222.2224.00
lo0.16383              up    up   inet

user@PE2>  show bgp summary
Groups: 1 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 1          1          0          0          0          0
Peer              AS      InPkt     OutPkt    OutQ   Flaps Last Up/DwnState|#Active/Received/Damped...
10.255.41.175     500         24         27       0       0       11:46 0/0/0                0/0/0
10.255.41.173     500         25         25       0       0       11:11 1/1/0                0/0/0


                       user@PE2>  show route protocol ospf

                       inet.0: 20 destinations, 22 routes (19 active, 0 holddown, 1 hidden)
                       + = Active Route, - = Last Active, * = Both

                       10.255.41.175/32   *[OSPF/10] 00:00:20, metric 1
                                           > via so-1/2/0.3
                       10.255.41.180/32    [OSPF/10] 00:00:20, metric 1
                                           > to 10.31.4.2 via fe-0/2/3.0
                       10.255.41.173/32    *[OSPF/10] 00:00:20, metric 2
                                           > via so-1/2/0.3
                       10.31.2.0/24       *[OSPF/10] 00:00:20, metric 2
                                           > via so-1/2/0.3
                       10.31.3.0/24        [OSPF/10] 00:00:20, metric 1
                                           > via so-1/2/0.3
                       224.0.0.5/32       *[OSPF/10] 00:13:46, metric 1
                                             MultiRecv

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

                       inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

                       user@PE2>  show route protocol bgp

                       inet.0: 20 destinations, 22 routes (19 active, 0 holddown, 1 hidden)
                       + = Active Route, - = Last Active, * = Both

                       10.255.41.177/32   *[BGP/170] 00:11:23, localpref 100, from 10.255.41.173
                                             AS path: I
                                           > via so-1/2/0.3

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

```
inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
```

Router PE2 Status: LS1

```
user@PE2> show bgp summary logical-system ls1

Groups: 2 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 0          0          0          0          0          0
inet.2                 0          0          0          0          0          0
bgp.l3vpn.0            4          4          0          0          0          0
bgp.l3vpn.2            0          0          0          0          0          0
Peer               AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Damped...
10.10.10.10       100         29         31       0       0       11:25 Establ
  bgp.l3vpn.0: 4/4/0
  bgp.l3vpn.2: 0/0/0
  blue.inet.0: 2/2/0
  red.inet.0: 2/2/0
10.21.4.2         300         27         28       0       0       11:40 Establ
  blue.inet.0: 1/1/0
```

red VPN

```
user@PE2> show route logical-system ls1 table red
red.inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.11.1.0/24       *[BGP/170] 00:12:02, localpref 100, from 10.10.10.10
                      AS path: I
                    > via so-1/2/0.1, label-switched-path to_10.10.10.10
10.11.1.100/32     *[BGP/170] 00:12:02, MED 1, localpref 100, from 10.10.10.10
                      AS path: I
                    > via so-1/2/0.1, label-switched-path to_10.10.10.10
10.11.4.0/24       *[Direct/0] 00:13:22
                    > via fe-0/2/1.0
10.11.4.1/32       *[Local/0] 00:13:29
                      Local via fe-0/2/1.0
10.11.4.100/32     *[OSPF/10] 00:12:35, metric 1
                    > to 10.11.4.2 via fe-0/2/1.0
224.0.0.5/32       *[OSPF/10] 00:15:02, metric 1
                      MultiRecv
```

blue VPN

```
user@PE2> show route logical-system ls1 table blue
blue.inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.21.1.0/24       *[BGP/170] 00:13:12, localpref 100, from 10.10.10.10
                      AS path: I
                    > via so-1/2/0.1, label-switched-path to_10.10.10.10
10.21.1.100/32     *[BGP/170] 00:13:12, localpref 100, from 10.10.10.10
                      AS path: 200 I
                    > via so-1/2/0.1, label-switched-path to_10.10.10.10
10.21.4.0/24       *[Direct/0] 00:14:32
                    > via fe-0/2/2.0
10.21.4.1/32       *[Local/0] 00:14:39
                      Local via fe-0/2/2.0
10.21.4.100/32     *[BGP/170] 00:13:27, localpref 100
                      AS path: 300 I
                    > to 10.21.4.2 via fe-0/2/2.0
```

```
user@PE2>  show mpls lsp logical-system ls1
Ingress LSP: 1 sessions
To              From            State Rt ActivePath       P     LSPname
10.10.10.10     10.10.10.12     Up     0                  *     to_10.10.10.10
Total 1 displayed, Up 1, Down 0
Egress LSP: 1 sessions
To              From            State Rt Style Labelin Labelout LSPname
10.10.10.12     10.10.10.10     Up     0 1 FF       3        - to_10.10.10.12
Total 1 displayed, Up 1, Down 0
Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

user@PE2>  show rsvp session logical-system ls1
Ingress RSVP: 1 sessions
To              From            State Rt Style Labelin Labelout LSPname
10.10.10.10     10.10.10.12     Up     0 1 FF       -   100000 to_10.10.10.10
Total 1 displayed, Up 1, Down 0
Egress RSVP: 1 sessions
To              From            State Rt Style Labelin Labelout LSPname
10.10.10.12     10.10.10.10     Up     0 1 FF       3        - to_10.10.10.12
Total 1 displayed, Up 1, Down 0
Transit RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Router PE2 Status: LS2

```
user@PE2> show vpls connections logical-system ls2
Layer-2 VPN Connections:

Legend for connection status (St)
OR -- out of range              WE -- intf encaps != instance encaps
EI -- encapsulation invalid     Dn -- down
EM -- encapsulation mismatch    VC-Dn -- Virtual circuit down
CM -- control-word mismatch     -> -- only outbound conn is up
CN -- circuit not provisioned   <- -- only inbound  conn is up
OL -- no outgoing label         Up -- operational
NC -- intf encaps not CCC/TCC   XX -- unknown
NP -- intf h/w not present

Legend for interface status
Up -- operational
Dn -- down

Instance: new
Local site: newPE (2)
    connection-site           Type  St     Time last up          # Up trans
    1                         rmt   Up     Jul 16 14:05:25 2003           1
      Local interface: vt-1/1/0.40960, Status: Up, Encapsulation: VPLS
      Remote PE: 10.20.20.20, Negotiated control-word: No
      Incoming label: 800000, Outgoing label: 800001
```

```
user@PE2>  show bgp summary logical-system ls2
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l2vpn.0            1          1          0          0          0          0
Peer              AS      InPkt     OutPkt    OutQ   Flaps Last Up/DwnState|#Active/Received/Damped...
10.20.20.20       400         29         31       0       0       13:29 Establ
```

```
bgp.l2vpn.0: 1/1/0
new.l2vpn.0: 1/1/0
```

```
user@PE2>  show mpls lsp logical-system ls2
Ingress LSP: 1 sessions
To              From            State Rt ActivePath       P     LSPname
10.20.20.20     10.20.20.22     Up     0                  *     to_10.20.20.20
Total 1 displayed, Up 1, Down 0

Egress LSP: 1 sessions
To              From            State Rt Style Labelin Labelout LSPname
10.20.20.22     10.20.20.20     Up     0 1 FF       3        - to_10.20.20.22
Total 1 displayed, Up 1, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

user@PE2>  show rsvp session logical-system ls2
Ingress RSVP: 1 sessions
To              From            State Rt Style Labelin Labelout LSPname
10.20.20.20     10.20.20.22     Up     0 1 FF       -   100016 to_10.20.20.20
Total 1 displayed, Up 1, Down 0

Egress RSVP: 1 sessions
To              From            State Rt Style Labelin Labelout LSPname
10.20.20.22     10.20.20.20     Up     0 1 FF       3        - to_10.20.20.22
Total 1 displayed, Up 1, Down 0

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

Router CE5 Status

```
user@CE5> show route table
inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
10.11.1.0/24       *[OSPF/150] 00:19:47, metric 0, tag 3489661028
                    > to 10.11.4.1 via fe-0/3/1.0
10.11.1.100/32     *[OSPF/10] 00:19:47, metric 2
                    > to 10.11.4.1 via fe-0/3/1.0
10.11.4.0/24       *[Direct/0] 00:21:12
                    > via fe-0/3/1.0
10.11.4.2/32       *[Local/0] 00:21:24
                      Local via fe-0/3/1.0
10.11.4.100/32     *[Direct/0] 00:22:37
                    > via lo0.0
224.0.0.5/32       *[OSPF/10] 00:22:44, metric 1
                      MultiRecv
```

Router CE6 Status

```
user@CE6> show route table
inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.21.1.0/24       *[BGP/170] 00:19:53, localpref 100
```

```
                      AS path: 100 I
                    > to 10.21.4.1 via fe-0/3/2.0
10.21.1.100/32     *[BGP/170] 00:19:53, localpref 100
                      AS path: 100 200 I
                    > to 10.21.4.1 via fe-0/3/2.0
10.21.4.0/24       *[Direct/0] 00:21:16
                    > via fe-0/3/2.0
10.21.4.2/32       *[Local/0] 00:21:28
                      Local via fe-0/3/2.0
10.21.4.100/32     *[Direct/0] 00:22:41
                    > via lo0.0
```

Router CE7 Status

```
user@CE7> show route table

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

10.1.1.0/24       *[Direct/0] 00:21:03
                    > via fe-0/3/0.0
10.1.1.2/32       *[Local/0] 00:21:15
                      Local via fe-0/3/0.0
```

Logical System Administrator Verification Output

Because logical system administrators only have access to the configuration information of the logical system(s) to which they are assigned, the verification output will be limited to these logical systems as well. The following outputs show what the logical system administrator "ls1–admin" in this chapter's example configuration would see.

Verifying Routing Instance Connectivity

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

CE1 to CE5 (red VPN)

```
user@CE1> ping 10.11.4.100
PING 10.11.4.100 (10.11.4.100): 56 data bytes
64 bytes from 10.11.4.100: icmp_seq=0 ttl=252 time=1.216 ms
64 bytes from 10.11.4.100: icmp_seq=1 ttl=252 time=1.052 ms
^C
--- 10.11.4.100 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.052/1.134/1.216/0.082 ms
```

CE2 to CE6 (blue VPN)

```
user@CE2> ping 10.21.4.100
PING 10.21.4.100 (10.21.4.100): 56 data bytes
64 bytes from 10.21.4.100: icmp_seq=0 ttl=252 time=1.205 ms
64 bytes from 10.21.4.100: icmp_seq=1 ttl=252 time=1.021 ms
^C
--- 10.21.4.100 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.021/1.113/1.205/0.092 ms
```

CE3 to CE7 (VPLS)

```
user@CE3> ping 10.1.1.2
PING 10.1.1.2 (10.1.1.2): 56 data bytes
64 bytes from 10.1.1.2: icmp_seq=0 ttl=255 time=1.186 ms
64 bytes from 10.1.1.2: icmp_seq=1 ttl=255 time=1.091 ms
64 bytes from 10.1.1.2: icmp_seq=2 ttl=255 time=1.081 ms
^C
--- 10.1.1.2 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.081/1.119/1.186/0.047 ms
```

For More Information

Because the concepts that constitute logical systems cut across the entire JUNOS software documentation set, you will find the following manuals to be useful references:

- For additional information about logical system administration, see the *JUNOS System Basics Configuration Guide*.
- For additional information about routing protocols, see the *JUNOS Routing Protocols Configuration Guide*.
- For additional information about policies, see the *JUNOS Policy Framework Configuration Guide*.
- For additional information about interface configuration, see the *JUNOS Network Interfaces Configuration Guide*.
- For additional information about MPLS and related protocols, see the *JUNOS MPLS Applications Configuration Guide*.
- For additional information about VPN protocols, see the *JUNOS VPNs Configuration Guide*.
- For additional information about multicast protocols, see the *JUNOS Multicast Protocols Configuration Guide*.
- For additional information about operational mode commands and output, see the *JUNOS Interfaces Command Reference*, the *JUNOS Routing Protocols and Policies Command Reference*, and the *JUNOS System Basics and Services Command Reference*.

Revision History

13 April 2009—9.5R1 Release. Roy Spencer.

15 January 2009—9.4R1 Release. Roy Spencer.

10 October 2008—Added support for logical systems to replace logical routers. Roy Spencer.

10 July 2008—9.2R1 Release. Roy Spencer.

10 April 2008—9.1R1 Release. Roy Spencer.

1 February 2008—9.0R1 Release. Fawn Damitio.

5 October 2007—Added support for logical router administrators and information about enhanced SNMP features. 8.5R1 Release. Fawn Damitio.

29 June 2007—Added support for SNMP usage limits. 8.4R1 Release. Fawn Damitio.

27 March 2007—8.3R1 Release. Fawn Damitio.

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

15 September 2006—8.1R1 Release. Richard Hendricks.

29 June 2006—8.0R1 Release. Richard Hendricks.

27 March 2006—7.6R1 Release. Richard Hendricks.

9 January 2006—Added SNMP support within a logical router, 7.5R1 Release. Richard Hendricks.

14 September 2005—Added multicast protocol RP and source DR functionality within a logical router, 7.4R1 Release. Richard Hendricks.

13 June 2005—7.3R1 Release. Richard Hendricks.

5 April 2005—7.2R1 Release. Richard Hendricks.

2 February 2005—7.1R1 Release. Richard Hendricks.

6 October 2004—Added support for implementing logical tunnel (lt) interfaces on an integrated Adaptive Services Module in an M7i router, 7.0R1 Release. Richard Hendricks.

6 July 2004—6.4R1 Release. Richard Hendricks.

5 April 2004—Revised lt interface families and encapsulation types and added support for graceful Routing Engine switchover, 6.3R1 Release. Richard Hendricks.

22 December 2003—Added new protocol family and encapsulation types for logical tunnel interfaces, 6.2R1 Release. Richard Hendricks.

22 September 2003—Added a new example, the logical tunnel (lt) interface type, and interface hierarchy changes. 6.1R1 Release. Richard Hendricks.

30 June 2003—Initial document written, Release 6.0R1. Richard Hendricks.

Chapter 8

OSPF Version 3 for IPv6

This feature guide covers the following topics:

Overview

OSPF version 2, introduced as RFC 2328 in 1998, has been one of the most widely deployed interior gateway protocols (IGPs) for intradomain routing. The protocol is extended in version 3 (RFC 2740) to support OSPF in IPv6 networks. Most of the functionality of OSPFv2 carries over into OSPFv3, but there are some significant changes to explore.

OSPFv3 adds support for IPv6 in the Open Shortest Path First (OSPF) routing protocol, as detailed in RFC 2740. Most configuration and operational commands function essentially the same as in OSPFv2:

- All OSPFv3 operational and configuration commands include the identifier **ospf3** in place of the familiar **ospf** option. For example, **show ospf database** in OSPFv2 becomes **show ospf3 database** in OSPFv3.
- OSPFv3 Router IDs, Area IDs, and LSA link-state IDs remain at the OSPFv2 IPv4 size of 32 bits.
- All the optional capabilities in OSPFv2 for IPv4, such as not-so-stubby areas (NSSA), are supported in OSPFv3 for IPv6.

However, there are many significant changes to note about OSPFv3 for IPv6:

- Router link-state advertisements (LSAs) and Network LSAs no longer carry prefix information. In OSPFv3, these LSAs only carry topology information.

NOTE: Because addressing information in the LSA header, Router LSA, and Network LSA (Type 2) has been removed, the OSPFv3 protocol is designed to be network protocol independent.

- New and modified LSAs have been created to handle the flow of IPv6 addresses and prefixes in an OSPFv3 network. As a result, some show command output appears in a different format for OSPFv3. The LSAs that have been modified are:
  - Interarea-Prefix LSA—This replaces the Network Summary or Type 3 LSA.
  - Interarea Router LSA—This replaces the Autonomous System Boundary Router (ASBR) Summary or Type 4 LSA.

  New LSAs introduced in OSPFv3 are:
  - Link LSA—This LSA has local scope and does not extend beyond the link it is associated with. The purpose of a link LSA is to provide the router's IPv6 link-local address to neighbors, inform other routers of the associated IPv6 prefixes available on the link, and provide information to the Network LSA. On all OSPF interfaces except virtual links, OSPF packets are sent using the interface's link-local address as the source address.

NOTE: A link-local address is an IPv6 address that starts with the first 10 bits set to 1111111010. This is often displayed in hexadecimal as fe80.

Juniper Networks M-series, MX-series, and T-series routing platforms automatically generate link-local addresses when IPv6 is enabled. The routing platform selects one interface MAC address (derived from the available interfaces) and appends this to the fe80 prefix with some additional bit stuffing. For more information about link-local addresses, see RFC 2373.

  - Intra-Area-Prefix LSA—This carries all IPv6 prefix information to all OSPFv3 routers within an area (this information in IPv4 is carried by the Router and Network LSAs).
- OSPFv3 now runs on a per-link basis, instead of on a per-IP-subnet basis.
- IPv6 link-local addresses are used for OSPFv3 neighbor exchanges (except over virtual links).
- The flooding scope for LSAs has been generalized into three categories for OSPFv3:
  - Link-local scope—The OSPFv3 packet is flooded to the members of a link.
  - Area scope—The OSPFv3 packet is flooded to all members of an OSPFv3 area.

    - AS scope—The OSPFv3 packet is flooded to all members of an AS.
- Authentication has been removed from the OSPFv3 protocol itself and relies on the authentication header (AH) and Encapsulating Security Payload (ESP) portions of the IP Security (IPSec) protocol for all authentication tasks in IPv6. For more information about configuring IPSec, see "Configuring IPSec" on page 410.
- Label-switched paths (LSPs) and traffic engineering are not supported in OSPFv3.
- Neighboring routers are always identified by the 32-bit router ID in OSPFv3.

System Requirements

To implement OSPFv3 for IPv6, your system must meet these minimum requirements:

- JUNOS Release 8.2 or later for MX-series routing platforms
- JUNOS Release 7.2 or later for J-series Services Routers
- JUNOS Release 5.5 or later for M-series and T-series routing platforms
- Two Juniper Networks J-series, M-series, MX-series, or T-series routing platforms

Terms and Acronyms

L

link-state advertisement (LSA) A multi-tiered message format for OSPFv2 and OSPFv3 that carries information about the OSPF network to OSPF-enabled routers. The collection of LSAs forms the link-state database used by the routers to select optimum paths. Different LSA levels limit the scope of OSPF protocol message delivery to links, areas, or autonomous systems (ASs).

O

Open Shortest Path First (OSPF) A link-state IGP that makes routing decisions based on the shortest-path-first (SPF) algorithm (also referred to as the *Dijkstra algorithm*).

OSPFv3 The IPv6-enabled version of the OSPF protocol.

Configuring OSPFv3 for IPv6

To implement OSPFv3 for IPv6, you must configure the following:

- Configuring OSPFv3 as the Routing Protocol on page 238
- Configuring Interfaces in OSPFv3 Areas on page 238
- Configuring Virtual Links for OSPFv3 on page 238

Configuring OSPFv3 as the Routing Protocol

You enable OSPFv3 almost the same way you enable OSPFv2. The only difference is that you use the ospf3 statement in place of ospf at the [edit protocols] hierarchy level.

```
[edit]
protocols {
   ospf3 {
      ...
   }
}
```

Configuring Interfaces in OSPFv3 Areas

To place selected interfaces in an OSPFv3 area, use the interface statement at the [edit protocols ospf3 area *area-number*] hierarchy level.

```
[edit]
protocols {
   ospf3 {
      area 0 {
         interface at-0/0/0.0;
         interface fe-1/1/1;
      }
   }
}
```

Configuring Virtual Links for OSPFv3

Virtual links can connect discontiguous sections of the OSPF backbone Area 0 or extend backbone access to areas not directly adjacent to Area 0 (a requirement of the OSPF protocol). To configure a virtual link, configure the virtual-link statement at the [edit protocols ospf3 area 0] hierarchy level. In the statement, specify the router ID of your neighbor (often the loopback interface IP address) and the OSPFv3 area that the virtual link travels across to reach Area 0.

```
[edit]
protocols {
   ospf3 {
      area 0.0.0.0 {
         virtual-link neighbor-id neighbor-router-id transit-area area;
      }
   }
}
```

Example: Configuring OSPFv3 for IPv6

Figure 18: OSPFv3 for IPv6 Topology Diagram

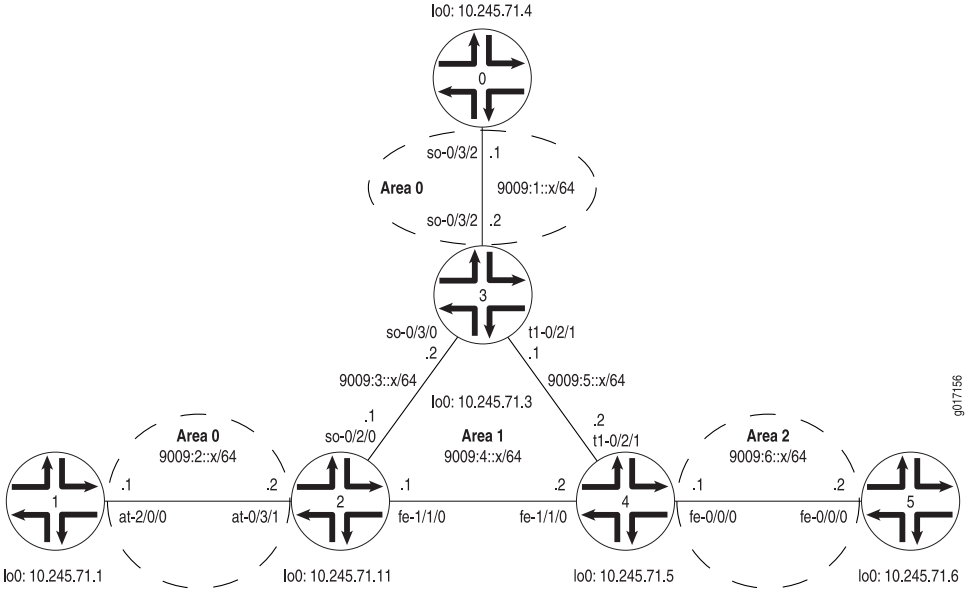


Figure 18 on page 239 shows an OSPFv3 topology. Routers 0, 1, 2, and 3 are connected to the OSPFv3 backbone Area 0; Routers 2, 3, and 4 connect to each other across Area 1; and Area 2 is located between Routers 4 and 5. Because Router 5 does not have a direct adjacency to Area 0, a virtual link is required across Area 1 between Routers 3 and 4. Similarly, because Routers 0 and 1 have two separate Area 0 backbone sections, you need to configure a second virtual link across Area 1 between Routers 2 and 3.

On Router 0, add the so-0/3/2 interface into Area 0 of the OSPFv3 process.

Router 0

```
[edit]
interfaces {
   so-0/3/2 {
      unit 0 {
         family inet {
            address 10.19.1.1/24;
         }
         family inet6 {
            address 9009:1::1/64;
         }
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.245.71.4/32;
         }
         family inet6 {
```

```
            address feee::10:255:71:4/128;
          }
        }
      }
    }
    protocols {
      ospf3 {
        area 0.0.0.0 {
          interface so-0/3/2.0;
          interface lo0.0 {
            passive;
          }
        }
      }
    }
```

On Router 1, add the at-2/0/0 interface into Area 0 of the OSPFv3 process:

Router 1

```
[edit]
interfaces {
  at-2/0/0 {
    atm-options {
      vpi 0;
    }
    unit 0 {
      vci 0.77;
      family inet {
        address 10.19.2.1/24;
      }
      family inet6 {
        address 9009:2::1/64;
      }
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.245.71.1/32;
      }
      family inet6 {
        address feee::10:255:71:1/128;
      }
    }
  }
}
protocols {
  ospf3 {
    area 0.0.0.0 {
      interface at-2/0/0.0;
      interface lo0.0 {
        passive;
      }
    }
  }
}
```

On Router 2, add the interfaces connected to Routers 1, 3, and 4 into the OSPFv3 process. You must also complete the virtual link to Router 3 through Area 1 so that Router 1 can access the discontiguous portion of the OSPF backbone found on Router 0.

Router 2

```
[edit]
interfaces {
   so-0/2/0 {
      unit 0 {
         family inet {
            address 10.19.3.1/24;
         }
         family inet6 {
            address 9009:3::1/64;
         }
      }
   }
   at-0/3/1 {
      atm-options {
         vpi 0 {
            maximum-vcs 1200;
         }
      }
      unit 0 {
         vci 0.77;
         family inet {
            address 10.19.2.2/24;
         }
         family inet6 {
            address 9009:2::2/64;
         }
      }
   }
   fe-1/1/0 {
      unit 0 {
         family inet {
            address 10.19.4.1/24;
         }
         family inet6 {
            address 9009:4::1/64;
         }
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.245.71.11/32;
         }
         family inet6 {
            address feee::10:255:71:11/128;
         }
      }
   }
}
protocols {
   ospf3 {
```

```
    area 0.0.0.0 {
      virtual-link neighbor-id 10.245.71.3 transit-area 0.0.0.1;
      interface at-0/3/1.0;
    }
    area 0.0.0.1 {
      interface so-0/2/0.0 {
        metric 1;
      }
      interface fe-1/1/0.0 {
        metric 10;
      }
      interface lo0.0 {
        passive;
      }
    }
  }
}
```

For the OSPFv3 process on Router 3, configure the interfaces connected to Routers 2 and 4 into Area 1 and the interface connected to Router 0 into Area 0. You must also configure two virtual links through Area 1—one connecting to Router 2 and the second connecting to Router 4. The virtual links allow Router 5 to access the OSPF backbone, and connect the discontiguous sections of Area 0 located at Routers 0 and 1.

Router 3

```
[edit]
interfaces {
  t1-0/2/1 {
    unit 0 {
      family inet {
        address 10.19.5.1/24;
      }
      family inet6 {
        address 9009:5::1/64;
      }
    }
  }
  so-0/3/0 {
    unit 0 {
      family inet {
        address 10.19.3.2/24;
      }
      family inet6 {
        address 9009:3::2/64;
      }
    }
  }
  so-0/3/2 {
    unit 0 {
      family inet {
        address 10.19.1.2/24;
      }
      family inet6 {
        address 9009:1::2/64;
      }
    }
```

```
    }
    lo0 {
      unit 0 {
        family inet {
          address 10.245.71.3/32;
        }
        family inet6 {
          address feee::10:255:71:3/128;
        }
      }
    }
  }
  protocols {
    ospf3 {
      area 0.0.0.0 {
        virtual-link neighbor-id 10.245.71.11 transit-area 0.0.0.1;
        virtual-link neighbor-id 10.245.71.5 transit-area 0.0.0.1;
        interface so-0/3/2.0;
      }
      area 0.0.0.1 {
        interface so-0/3/0.0 {
          metric 1;
        }
        interface t1-0/2/1.0 {
          metric 1;
        }
        interface lo0.0 {
          passive;
        }
      }
    }
  }
```

On Router 4, add the connected interfaces into the OSPFv3 process. You must also complete the virtual link to Router 3 through Area 1 so that Router 5 can access the OSPF backbone.

Router 4

```
[edit]
interfaces {
  fe-0/0/0 {
    unit 0 {
      family inet {
        address 10.19.6.1/24;
      }
      family inet6 {
        address 9009:6::1/64;
      }
    }
  }
  t1-0/2/1 {
    unit 0 {
      family inet {
        address 10.19.5.2/24;
      }
      family inet6 {
        address 9009:5::2/64;
```

```
            }
          }
        }
        fe-1/1/0 {
          unit 0 {
            family inet {
              address 10.19.4.2/24;
            }
            family inet6 {
              address 9009:4::2/64;
            }
          }
        }
        lo0 {
          unit 0 {
            family inet {
              address 10.245.71.5/32;
            }
            family inet6 {
              address feee::10:255:71:5/128;
            }
          }
        }
      }
      protocols {
        ospf3 {
          area 0.0.0.1 {
            interface fe-1/1/0.0 {
              metric 10;
            }
            interface t1-0/2/1.0 {
              metric 1;
            }
            interface lo0.0 {
              passive;
            }
          }
          area 0.0.0.0 {
            virtual-link neighbor-id 10.245.71.3 transit-area 0.0.0.1;
          }
          area 0.0.0.2 {
            interface fe-0/0/0.0;
          }
        }
      }
```

On Router 5, add the **fe-0/0/0** interface into the OSPFv3 process to complete this example:

Router 5

```
[edit]
interfaces {
  fe-0/0/0 {
    unit 0 {
      family inet {
        address 10.19.6.2/24;
      }
```

```
            family inet6 {
               address 9009:6::2/64;
            }
         }
      }
      lo0 {
         unit 0 {
            family inet {
               address 10.245.71.6/32;
            }
            family inet6 {
               address feee::10:255:71:6/128;
            }
         }
      }
   }
   protocols {
      ospf3 {
         area 0.0.0.2 {
            interface fe-0/0/0.0;
            interface lo0.0 {
               passive;
            }
         }
      }
   }
```

Verifying Your Work

To verify proper operation of OSPFv3 for IPv6, use the following commands:

- show ospf3 interface
- show ospf3 neighbor
- show ospf3 database
- show ospf3 route
- show interfaces terse (to see the IPv6 link local address assigned to the lo0 interface)

NOTE: To view prefix information, you must use the extensive option with the show ospf3 database command.

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

NOTE: In the below sample output, the stars indicate the "best" routes. These routes are the routes that are installed in the routing table.

Router 0 Status

```
user@router0> show ospf3 database
    OSPF link state database, area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Router      0.0.0.1          10.245.71.1       0x80000005   764  0x89ce  40
Router      0.0.0.1          10.245.71.3       0x80000006  1360  0x2357  72
Router     *0.0.0.1          10.245.71.4       0x80000004   758  0xc09c  40
Router      0.0.0.1          10.245.71.5       0x80000003  1891  0xf774  40
Router      0.0.0.1          10.245.71.11      0x80000005  1393  0x7f6b  56
InterArPfx  0.0.0.1          10.245.71.3       0x80000003   758  0x9f52  36
InterArPfx  0.0.0.2          10.245.71.3       0x80000003   616  0xb13d  36
InterArPfx  0.0.0.3          10.245.71.3       0x80000003   473  0x1da2  36
InterArPfx  0.0.0.4          10.245.71.3       0x80000003   458  0x99f0  44
InterArPfx  0.0.0.5          10.245.71.3       0x80000004  1058  0xbf22  36
InterArPfx  0.0.0.6          10.245.71.3       0x80000002  1958  0x5c67  36
InterArPfx  0.0.0.7          10.245.71.3       0x80000002  1816  0xf088  44
InterArPfx  0.0.0.8          10.245.71.3       0x80000002  1673  0xd3d6  36
InterArPfx  0.0.0.9          10.245.71.3       0x80000002  1658  0xa3df  44
InterArPfx  0.0.0.1          10.245.71.5       0x80000004   479  0xd50f  36
InterArPfx  0.0.0.2          10.245.71.5       0x80000003   310  0xa547  36
InterArPfx  0.0.0.3          10.245.71.5       0x80000003   913  0x1cbb  36
InterArPfx  0.0.0.5          10.245.71.5       0x80000003   163  0xddcd  36
InterArPfx  0.0.0.6          10.245.71.5       0x80000003    13  0xadd6  44
InterArPfx  0.0.0.7          10.245.71.5       0x80000002  2633  0x5f8a  36
InterArPfx  0.0.0.8          10.245.71.5       0x80000002  2488  0x427c  36
InterArPfx  0.0.0.9          10.245.71.5       0x80000002  2338  0xdcda  36
InterArPfx  0.0.0.10         10.245.71.5       0x80000002  2188  0x5929  44
InterArPfx  0.0.0.11         10.245.71.5       0x80000002  2038  0xc2af  44
InterArPfx  0.0.0.12         10.245.71.5       0x80000002   763  0x664   44
InterArPfx  0.0.0.1          10.245.71.11      0x80000003   463  0x6f7a  36
InterArPfx  0.0.0.2          10.245.71.11      0x80000003   328  0xa935  36
InterArPfx  0.0.0.3          10.245.71.11      0x80000003   193  0x427c  36
InterArPfx  0.0.0.4          10.245.71.11      0x80000003   163  0xd69d  44
InterArPfx  0.0.0.5          10.245.71.11      0x80000002  1993  0x6b78  36
InterArPfx  0.0.0.6          10.245.71.11      0x80000002  1963  0xd6dd  36
InterArPfx  0.0.0.7          10.245.71.11      0x80000002  1828  0x532c  44
InterArPfx  0.0.0.8          10.245.71.11      0x80000002  1663  0xa9f7  36
InterArPfx  0.0.0.9          10.245.71.11      0x80000002  1528  0x7901  44
InterArRtr  0.0.0.1          10.245.71.5       0x80000002   620  0xc69c  32
IntraArPfx  0.0.0.1          10.245.71.1       0x80000005   464  0x3f8   76
IntraArPfx  0.0.0.1          10.245.71.3       0x80000005  1509  0x5cc1  64
IntraArPfx *0.0.0.1          10.245.71.4       0x80000004   458  0xba44  64
```

```
IntraArPfx  0.0.0.1          10.245.71.11     0x80000003  1693  0xd835  64
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Extern     *0.0.0.1          10.245.71.4      0x80000003  1058  0x8449  36
Extern      0.0.0.1          10.245.71.6      0x80000003  1064  0xdc9e  36
    OSPF Link-Local link state database, interface so-0/3/2.0
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Link        0.0.0.6          10.245.71.3      0x80000004   158  0xae30  56
Link       *0.0.0.2          10.245.71.4      0x80000004   158  0x9e80  56

user@router0>  show ospf3 interface
Interface             State     Area            DR-ID           BDR-ID
Nbrs
lo0.0                 DRother  0.0.0.0         0.0.0.0         0.0.0.0
0
so-0/3/2.0            PtToPt   0.0.0.0         0.0.0.0         0.0.0.0
1

user@router0>  show ospf3 neighbor
ID               Interface              State     Pri   Dead
10.245.71.3      so-0/3/2.0             Full      128   34
  Neighbor-address fe80::201:afff:fe00:86ca

user@router0>  show ospf3 route
Prefix                                  Path   Route       NH   Metric
                                        type   type        type
10.245.71.1                             Intra  Router      IP   25
  NH-interface so-0/3/2.0
10.245.71.3                             Intra  Area BR     IP   12
  NH-interface so-0/3/2.0
10.245.71.5                             Intra  Area BR     IP   13
  NH-interface so-0/3/2.0
10.245.71.6                             Inter  AS BR       IP   33
  NH-interface so-0/3/2.0
10.245.71.11                            Intra  Area BR     IP   13
  NH-interface so-0/3/2.0
9009:1::/64                             Intra  Network     IP   12
  NH-interface so-0/3/2.0
9009:1::2/128                           Intra  Network     IP   12
  NH-interface so-0/3/2.0
9009:2::/64                             Intra  Network     IP   25
  NH-interface so-0/3/2.0
9009:2::2/128                           Intra  Network     IP   13
  NH-interface so-0/3/2.0
9009:3::/64                             Inter  Network     IP   13
  NH-interface so-0/3/2.0
9009:4::/64                             Inter  Network     IP   23
  NH-interface so-0/3/2.0
9009:5::/64                             Inter  Network     IP   13
  NH-interface so-0/3/2.0
9009:6::/64                             Inter  Network     IP   33
  NH-interface so-0/3/2.0
9009:110::/64                           Intra  Network     IP   27
  NH-interface so-0/3/2.0
9009:120::/64                           Inter  Network     IP   25
  NH-interface so-0/3/2.0
9009:130::/64                           Inter  Network     IP   15
  NH-interface so-0/3/2.0
9009:140::/64                           Inter  Network     IP   16
  NH-interface so-0/3/2.0
9009:150::/64                           Ext2   Network     IP   0
```

```
  NH-interface so-0/3/2.0
feee::10:255:71:1/128                       Intra  Network     IP   25
  NH-interface so-0/3/2.0
feee::10:255:71:3/128                       Inter  Network     IP   12
  NH-interface so-0/3/2.0
feee::10:255:71:4/128                       Intra  Network     IP   0
  NH-interface lo0.0
feee::10:255:71:5/128                       Inter  Network     IP   13
  NH-interface so-0/3/2.0
feee::10:255:71:6/128                       Inter  Network     IP   33
  NH-interface so-0/3/2.0
feee::10:255:71:11/128                      Inter  Network     IP   13
  NH-interface so-0/3/2.0

user@router0>  show interfaces terse
Interface       Admin Link Proto Local                 Remote
...
so-0/3/2        up    up
so-0/3/2.0      up    up   inet  10.19.1.1/24
                           inet6 9009:1::1/64
                                 fe80::201:afff:fe03:6fa1/64
...
lo0             up    up
lo0.0           up    up   inet  10.245.71.4         --> 0/0
                                 127.0.0.1           --> 0/0
                            inet6 fe80::201:afff:fe03:6fa1
                                 feee::10:255:71:4
...
```

To provide a comparison between OSPFv3 show commands and legacy OSPFv2 show commands, the following is some sample output of the OSPFv2 connection between Routers 0 and 3:

```
user@router0> show ospf interface
Interface             State     Area            DR ID           BDR ID
Nbrs
lo0.0                 DRother   0.0.0.0         0.0.0.0         0.0.0.0
0
lo0.0                 DRother   0.0.0.0         0.0.0.0         0.0.0.0
0
so-0/3/2.0            PtToPt    0.0.0.0         0.0.0.0         0.0.0.0
1

user@router0>  show ospf neighbor
  Address         Interface             State      ID              Pri  Dead
10.19.1.2        so-0/3/2.0            Full      10.245.71.3      128   35

user@router0>  show ospf database
    OSPF link state database, area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router   10.245.71.3      10.245.71.3      0x80000002   636  0x2  0x5c45  60
Router  *10.245.71.4      10.245.71.4      0x80000002   640  0x2  0x267a  60

user@router0>  show ospf route
Prefix              Path   Route       NH   Metric  NextHop       Nexthop

                    Type   Type        Type         Interface     addr/label
10.245.71.3         Intra  Router      IP   1       so-0/3/2.0
10.19.1.0/24        Intra  Network     IP   1       so-0/3/2.0
```

```
10.245.71.3/32       Intra  Network     IP   1       so-0/3/2.0
10.245.71.4/32       Intra  Network     IP   0       lo0.0
```

Router 1 Status

```
user@router1> show ospf3 interface
Interface             State     Area            DR-ID           BDR-ID
Nbrs
at-2/0/0.0            PtToPt    0.0.0.0         0.0.0.0         0.0.0.0
1
ge-1/1/0.0            DRother   0.0.0.0         0.0.0.0         0.0.0.0
0
lo0.0                 DRother   0.0.0.0         0.0.0.0         0.0.0.0
0

user@router1>  show ospf3 neighbor
ID               Interface              State     Pri   Dead
10.245.71.11     at-2/0/0.0             Full      128   36
  Neighbor-address fe80::2a0:a5ff:fe3d:56

user@router1>  show ospf3 database
    OSPF link state database, area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Router     *0.0.0.1          10.245.71.1       0x80000005   574  0x89ce  40
Router      0.0.0.1          10.245.71.3       0x80000006  1174  0x2357  72
Router      0.0.0.1          10.245.71.4       0x80000004   574  0xc09c  40
Router      0.0.0.1          10.245.71.5       0x80000003  1706  0xf774  40
Router      0.0.0.1          10.245.71.11      0x80000005  1205  0x7f6b  56
InterArPfx  0.0.0.1          10.245.71.3       0x80000003   572  0x9f52  36
InterArPfx  0.0.0.2          10.245.71.3       0x80000003   430  0xb13d  36
InterArPfx  0.0.0.3          10.245.71.3       0x80000003   288  0x1da2  36
InterArPfx  0.0.0.4          10.245.71.3       0x80000003   273  0x99f0  44
InterArPfx  0.0.0.5          10.245.71.3       0x80000004   873  0xbf22  36
InterArPfx  0.0.0.6          10.245.71.3       0x80000002  1773  0x5c67  36
InterArPfx  0.0.0.7          10.245.71.3       0x80000002  1630  0xf088  44
InterArPfx  0.0.0.8          10.245.71.3       0x80000002  1488  0xd3d6  36
InterArPfx  0.0.0.9          10.245.71.3       0x80000002  1473  0xa3df  44
InterArPfx  0.0.0.1          10.245.71.5       0x80000004   293  0xd50f  36
InterArPfx  0.0.0.2          10.245.71.5       0x80000003   124  0xa547  36
InterArPfx  0.0.0.3          10.245.71.5       0x80000003   727  0x1cbb  36
InterArPfx  0.0.0.5          10.245.71.5       0x80000002  2695  0xdfcc  36
InterArPfx  0.0.0.6          10.245.71.5       0x80000002  2601  0xafd5  44
InterArPfx  0.0.0.7          10.245.71.5       0x80000002  2448  0x5f8a  36
InterArPfx  0.0.0.8          10.245.71.5       0x80000002  2302  0x427c  36
InterArPfx  0.0.0.9          10.245.71.5       0x80000002  2152  0xdcda  36
InterArPfx  0.0.0.10         10.245.71.5       0x80000002  2002  0x5929  44
InterArPfx  0.0.0.11         10.245.71.5       0x80000002  1852  0xc2af  44
InterArPfx  0.0.0.12         10.245.71.5       0x80000002   577  0x664   44
InterArPfx  0.0.0.1          10.245.71.11      0x80000003   275  0x6f7a  36
InterArPfx  0.0.0.2          10.245.71.11      0x80000003   140  0xa935  36
InterArPfx  0.0.0.3          10.245.71.11      0x80000003     5  0x427c  36
InterArPfx  0.0.0.4          10.245.71.11      0x80000002  2105  0xd89c  44
InterArPfx  0.0.0.5          10.245.71.11      0x80000002  1805  0x6b78  36
InterArPfx  0.0.0.6          10.245.71.11      0x80000002  1775  0xd6dd  36
InterArPfx  0.0.0.7          10.245.71.11      0x80000002  1640  0x532c  44
InterArPfx  0.0.0.8          10.245.71.11      0x80000002  1475  0xa9f7  36
InterArPfx  0.0.0.9          10.245.71.11      0x80000002  1340  0x7901  44
InterArRtr  0.0.0.1          10.245.71.5       0x80000002   434  0xc69c  32
IntraArPfx *0.0.0.1          10.245.71.1       0x80000005   274  0x3f8   76
```

```
IntraArPfx  0.0.0.1          10.245.71.3      0x80000005  1323  0x5cc1  64
IntraArPfx  0.0.0.1          10.245.71.4      0x80000004   275  0xba44  64
IntraArPfx  0.0.0.1          10.245.71.11     0x80000003  1505  0xd835  64
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Extern      0.0.0.1          10.245.71.4      0x80000003   874  0x8449  36
Extern      0.0.0.1          10.245.71.6      0x80000003   877  0xdc9e  36
    OSPF Link-Local link state database, interface at-2/0/0.0
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Link       *0.0.0.3          10.245.71.1      0x80000004   874  0x296b  56
Link        0.0.0.6          10.245.71.11     0x80000003   605  0xaf4f  56

user@router1>  show ospf3 route
Prefix                                  Path   Route       NH   Metric
                                        type   type        type
10.245.71.3                             Intra  Area BR     IP   13
  NH-interface at-2/0/0.0
10.245.71.4                             Intra  AS BR       IP   25
  NH-interface at-2/0/0.0
10.245.71.5                             Intra  Area BR     IP   14
  NH-interface at-2/0/0.0
10.245.71.6                             Inter  AS BR       IP   34
  NH-interface at-2/0/0.0
10.245.71.11                            Intra  Area BR     IP   12
  NH-interface at-2/0/0.0
9009:1::/64                             Intra  Network     IP   25
  NH-interface at-2/0/0.0
9009:1::2/128                           Intra  Network     IP   13
  NH-interface at-2/0/0.0
9009:2::/64                             Intra  Network     IP   12
  NH-interface at-2/0/0.0
9009:2::2/128                           Intra  Network     IP   12
  NH-interface at-2/0/0.0
9009:3::/64                             Inter  Network     IP   13
  NH-interface at-2/0/0.0
9009:4::/64                             Inter  Network     IP   22
  NH-interface at-2/0/0.0
9009:5::/64                             Inter  Network     IP   14
  NH-interface at-2/0/0.0
9009:6::/64                             Inter  Network     IP   34
  NH-interface at-2/0/0.0
9009:100::/64                           Ext2   Network     IP   0
  NH-interface at-2/0/0.0
9009:110::/64                           Intra  Network     IP   2
  NH-interface ge-1/1/0.0
9009:120::/64                           Inter  Network     IP   24
  NH-interface at-2/0/0.0
9009:130::/64                           Inter  Network     IP   16
  NH-interface at-2/0/0.0
9009:140::/64                           Inter  Network     IP   17
  NH-interface at-2/0/0.0
9009:150::/64                           Ext2   Network     IP   0
  NH-interface at-2/0/0.0
feee::10:255:71:1/128                   Intra  Network     IP   0
  NH-interface lo0.0
feee::10:255:71:3/128                   Inter  Network     IP   13
  NH-interface at-2/0/0.0
feee::10:255:71:4/128                   Intra  Network     IP   25
  NH-interface at-2/0/0.0
feee::10:255:71:5/128                   Inter  Network     IP   14
  NH-interface at-2/0/0.0
```

```
feee::10:255:71:6/128                       Inter  Network     IP   34
  NH-interface at-2/0/0.0
feee::10:255:71:11/128                      Inter  Network     IP   12
  NH-interface at-2/0/0.0

user@router1>  show interfaces terse
Interface       Admin Link Proto Local                 Remote
...
at-2/0/0        up    up
at-2/0/0.0      up    up   inet  10.19.2.1/24
                           inet6 9009:2::1/64
                                 fe80::2a0:a5ff:fe3d:dbf/64
...
lo0             up    up
lo0.0           up    up   inet  10.245.71.1         --> 0/0
                                 127.0.0.1           --> 0/0
                            inet6 fe80::2a0:a5ff:fe3d:dbf
                                 feee::10:255:71:1
...
```

Router 2 Status

```
user@router2> show ospf3 interface
Interface             State     Area            DR-ID           BDR-ID
Nbrs
at-0/3/1.0            PtToPt   0.0.0.0         0.0.0.0         0.0.0.0
1
vl -10.245.71.3        PtToPt   0.0.0.0         0.0.0.0         0.0.0.0
 1
at-0/3/0.0            PtToPt   0.0.0.1         0.0.0.0         0.0.0.0
0
fe-1/1/0.0            DR       0.0.0.1         10.245.71.11    10.245.71.5
1
lo0.0                 DRother  0.0.0.1         0.0.0.0         0.0.0.0
0
so-0/2/0.0            PtToPt   0.0.0.1         0.0.0.0         0.0.0.0
1

user@router2>  show ospf3 neighbor
ID              Interface             State     Pri   Dead
10.245.71.1     at-0/3/1.0            Full      128   36
  Neighbor-address fe80::2a0:a5ff:fe3d:dbf
10.245.71.3       vl -10.245.71.3         Full        0   33
  Neighbor-address 9009:3::2
10.245.71.5     fe-1/1/0.0            Full      128   36
  Neighbor-address fe80::290:69ff:fe98:909d
10.245.71.3     so-0/2/0.0            Full      128   33
  Neighbor-address fe80::201:afff:fe00:86ca

user@router2>  show ospf3 database
    OSPF link state database, area 0.0.0.0
 Type       ID              Adv Rtr           Seq      Age  Cksum  Len
Router      0.0.0.1         10.245.71.1      0x80000005   277  0x89ce  40
Router      0.0.0.1         10.245.71.3      0x80000006   875  0x2357  72
Router      0.0.0.1         10.245.71.4      0x80000004   275  0xc09c  40
Router      0.0.0.1         10.245.71.5      0x80000003  1407  0xf774  40
Router     *0.0.0.1         10.245.71.11     0x80000005   906  0x7f6b  56
InterArPfx  0.0.0.1         10.245.71.3      0x80000003   273  0x9f52  36
InterArPfx  0.0.0.2         10.245.71.3      0x80000003   131  0xb13d  36
```

```
InterArPfx  0.0.0.3          10.245.71.3      0x80000002  2225  0x1fa1  36
InterArPfx  0.0.0.4          10.245.71.3      0x80000002  2076  0x9bef  44
InterArPfx  0.0.0.5          10.245.71.3      0x80000004   574  0xbf22  36
InterArPfx  0.0.0.6          10.245.71.3      0x80000002  1474  0x5c67  36
InterArPfx  0.0.0.7          10.245.71.3      0x80000002  1331  0xf088  44
InterArPfx  0.0.0.8          10.245.71.3      0x80000002  1189  0xd3d6  36
InterArPfx  0.0.0.9          10.245.71.3      0x80000002  1174  0xa3df  44
InterArPfx  0.0.0.1          10.245.71.5      0x80000003  2923  0xd70e  36
InterArPfx  0.0.0.2          10.245.71.5      0x80000002  2537  0xa746  36
InterArPfx  0.0.0.3          10.245.71.5      0x80000003   428  0x1cbb  36
InterArPfx  0.0.0.5          10.245.71.5      0x80000002  2396  0xdfcc  36
InterArPfx  0.0.0.6          10.245.71.5      0x80000002  2302  0xafd5  44
InterArPfx  0.0.0.7          10.245.71.5      0x80000002  2149  0x5f8a  36
InterArPfx  0.0.0.8          10.245.71.5      0x80000002  2003  0x427c  36
InterArPfx  0.0.0.9          10.245.71.5      0x80000002  1853  0xdcda  36
InterArPfx  0.0.0.10         10.245.71.5      0x80000002  1703  0x5929  44
InterArPfx  0.0.0.11         10.245.71.5      0x80000002  1553  0xc2af  44
InterArPfx  0.0.0.12         10.245.71.5      0x80000002   278  0x664   44
InterArPfx *0.0.0.1          10.245.71.11     0x80000002  2108  0x7179  36
InterArPfx *0.0.0.2          10.245.71.11     0x80000002  2076  0xab34  36
InterArPfx *0.0.0.3          10.245.71.11     0x80000002  1941  0x447b  36
InterArPfx *0.0.0.4          10.245.71.11     0x80000002  1806  0xd89c  44
InterArPfx *0.0.0.5          10.245.71.11     0x80000002  1506  0x6b78  36
InterArPfx *0.0.0.6          10.245.71.11     0x80000002  1476  0xd6dd  36
InterArPfx *0.0.0.7          10.245.71.11     0x80000002  1341  0x532c  44
InterArPfx *0.0.0.8          10.245.71.11     0x80000002  1176  0xa9f7  36
InterArPfx *0.0.0.9          10.245.71.11     0x80000002  1041  0x7901  44
InterArRtr  0.0.0.1          10.245.71.5      0x80000002   135  0xc69c  32
IntraArPfx  0.0.0.1          10.245.71.1      0x80000004   877  0x5f7   76
IntraArPfx  0.0.0.1          10.245.71.3      0x80000005  1024  0x5cc1  64
IntraArPfx  0.0.0.1          10.245.71.4      0x80000003  1176  0xbc43  64
IntraArPfx *0.0.0.1          10.245.71.11     0x80000003  1206  0xd835  64
    OSPF link state database, area 0.0.0.1
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Router      0.0.0.1          10.245.71.3      0x80000006   574  0xad3f  56
Router      0.0.0.1          10.245.71.5      0x80000006   577  0xde02  56
Router     *0.0.0.1          10.245.71.11     0x80000007   576  0x8853  56
Network    *0.0.0.4          10.245.71.11     0x80000003   606  0xfd16  32
InterArPfx  0.0.0.1          10.245.71.3      0x80000002  1774  0xc722  36
InterArPfx  0.0.0.2          10.245.71.3      0x80000002  1624  0x7b2f  44
InterArPfx  0.0.0.3          10.245.71.3      0x80000002   874  0x877   44
InterArPfx  0.0.0.1          10.245.71.5      0x80000003   352  0x30a9  36
InterArPfx  0.0.0.3          10.245.71.5      0x80000002   205  0x6013  44
InterArPfx *0.0.0.1          10.245.71.11     0x80000003   141  0xa33c  36
InterArPfx *0.0.0.2          10.245.71.11     0x80000003     6  0x5749  44
InterArPfx *0.0.0.3          10.245.71.11     0x80000002  1776  0x6f5e  36
InterArPfx *0.0.0.4          10.245.71.11     0x80000002  1641  0x7ff9  44
InterArRtr  0.0.0.1          10.245.71.3      0x80000002   724  0x6609  32
InterArRtr  0.0.0.1          10.245.71.5      0x80000002    64  0xc69c  32
IntraArPfx  0.0.0.1          10.245.71.3      0x80000004   424  0x4a98  88
IntraArPfx  0.0.0.1          10.245.71.5      0x80000004   502  0x3691  76
IntraArPfx *0.0.0.1          10.245.71.11     0x80000005   441  0x2c5   76
IntraArPfx *0.0.0.5          10.245.71.11     0x80000003   741  0xfa59  44
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Extern      0.0.0.1          10.245.71.4      0x80000003   575  0x8449  36
Extern      0.0.0.1          10.245.71.6      0x80000003   578  0xdc9e  36
    OSPF Link-Local link state database, interface at-0/3/1.0
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Link        0.0.0.3          10.245.71.1      0x80000004   577  0x296b  56
Link       *0.0.0.6          10.245.71.11     0x80000003   306  0xaf4f  56
```

```
    OSPF Link-Local link state database, interface fe-1/1/0.0
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Link        0.0.0.5          10.245.71.5      0x80000003   727  0x40dc  56
Link       *0.0.0.4          10.245.71.11     0x80000004   876  0x73ab  56
    OSPF Link-Local link state database, interface so-0/2/0.0
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Link        0.0.0.4          10.245.71.3      0x80000003  2074  0x9d6   56
Link       *0.0.0.3          10.245.71.11     0x80000004   276  0xed12  56

user@router2>  show ospf3 route
Prefix                                     Path   Route       NH   Metric
                                           type   type        type
10.245.71.1                                Intra  Router      IP   12
  NH-interface at-0/3/1.0
10.245.71.3                                Intra  Area BR     IP   1
  NH-interface so-0/2/0.0
10.245.71.4                                Intra  AS BR       IP   13
  NH-interface so-0/2/0.0
10.245.71.5                                Intra  Area BR     IP   2
  NH-interface so-0/2/0.0
10.245.71.6                                Inter  AS BR       IP   22
  NH-interface so-0/2/0.0
10.245.71.11;0.0.0.4                       Intra  Transit     IP   10
  NH-interface fe-1/1/0.0
9009:1::/64                                Intra  Network     IP   13
  NH-interface so-0/2/0.0
9009:1::2/128                              Intra  Network     IP   1
  NH-interface so-0/2/0.0
9009:2::/64                                Intra  Network     IP   12
  NH-interface at-0/3/1.0
9009:2::2/128                              Intra  Network     IP   0
  NH-interface at-0/3/1.0
9009:3::/64                                Intra  Network     IP   1
  NH-interface so-0/2/0.0
9009:4::/64                                Intra  Network     IP   10
  NH-interface fe-1/1/0.0
9009:5::/64                                Intra  Network     IP   2
  NH-interface so-0/2/0.0
9009:6::/64                                Inter  Network     IP   22
  NH-interface so-0/2/0.0
9009:100::/64                              Ext2   Network     IP   0
  NH-interface so-0/2/0.0
9009:110::/64                              Intra  Network     IP   14
  NH-interface at-0/3/1.0
9009:120::/64                              Intra  Network     IP   12
  NH-interface at-0/3/0.0
9009:130::/64                              Intra  Network     IP   4
  NH-interface so-0/2/0.0
9009:140::/64                              Intra  Network     IP   5
  NH-interface so-0/2/0.0
9009:150::/64                              Ext2   Network     IP   0
  NH-interface so-0/2/0.0
feee::10:255:71:1/128                      Intra  Network     IP   12
  NH-interface at-0/3/1.0
feee::10:255:71:3/128                      Intra  Network     IP   1
  NH-interface so-0/2/0.0
feee::10:255:71:4/128                      Intra  Network     IP   13
  NH-interface so-0/2/0.0
feee::10:255:71:5/128                      Intra  Network     IP   2
  NH-interface so-0/2/0.0
feee::10:255:71:6/128                      Inter  Network     IP   22
```

```
  NH-interface so-0/2/0.0
feee::10:255:71:11/128                      Intra  Network     IP   0
  NH-interface lo0.0

user@router2>  show interfaces terse
Interface       Admin Link Proto Local                 Remote
...
so-0/2/0        up    up
so-0/2/0.0      up    up   inet  10.19.3.1/24
                           inet6 9009:3::1/64
                                 fe80::2a0:a5ff:fe3d:56/64
...
at-0/3/1        up    up
at-0/3/1.0      up    up   inet  10.19.2.2/24
                           inet6 9009:2::2/64
                                 fe80::2a0:a5ff:fe3d:56/64
...
fe-1/1/0        up    up
fe-1/1/0.0      up    up   inet  10.19.4.1/24
                           inet6 9009:4::1/64
                                 fe80::290:69ff:fea0:809d/64
...
lo0             up    up
lo0.0           up    up   inet  10.245.71.11        --> 0/0
                                 127.0.0.1           --> 0/0
                           inet6 fe80::2a0:a5ff:fe3d:56
                                 feee::10:255:71:11
...
```

Router 3 Status

```
user@router3> show ospf3 interface
Interface             State     Area            DR-ID           BDR-ID
Nbrs
so-0/3/2.0            PtToPt    0.0.0.0         0.0.0.0         0.0.0.0
1
vl-10.245.71.11       PtToPt    0.0.0.0         0.0.0.0         0.0.0.0
1
vl-10.245.71.5        PtToPt    0.0.0.0         0.0.0.0         0.0.0.0
1
at-1/2/0.0            PtToPt    0.0.0.1         0.0.0.0         0.0.0.0
0
lo0.0                 DRother   0.0.0.1         0.0.0.0         0.0.0.0
0
so-0/3/0.0            PtToPt    0.0.0.1         0.0.0.0         0.0.0.0
1
t1-0/2/1.0            PtToPt    0.0.0.1         0.0.0.0         0.0.0.0
1

user@router3> show ospf3 neighbor
ID              Interface             State     Pri   Dead
10.245.71.4     so-0/3/2.0            Full      128   38
  Neighbor-address fe80::201:afff:fe03:6fa1
10.245.71.11    vl-10.245.71.11       Full        0   36
  Neighbor-address 9009:3::1
10.245.71.5     vl-10.245.71.5        Full        0   35
  Neighbor-address 9009:5::2
10.245.71.11    so-0/3/0.0            Full      128   37
  Neighbor-address fe80::2a0:a5ff:fe3d:56
```

```
10.245.71.5      t1-0/2/1.0             Full      128   39
  Neighbor-address fe80::2a0:a5ff:fe3d:b63

user@router3>  show ospf3 database
    OSPF link state database, area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Router      0.0.0.1          10.245.71.1       0x80000005    94  0x89ce  40
Router     *0.0.0.1          10.245.71.3       0x80000006   690  0x2357  72
Router      0.0.0.1          10.245.71.4       0x80000004    90  0xc09c  40
Router      0.0.0.1          10.245.71.5       0x80000003  1222  0xf774  40
Router      0.0.0.1          10.245.71.11      0x80000005   723  0x7f6b  56
InterArPfx *0.0.0.1          10.245.71.3       0x80000003    88  0x9f52  36
InterArPfx *0.0.0.2          10.245.71.3       0x80000002  2188  0xb33c  36
InterArPfx *0.0.0.3          10.245.71.3       0x80000002  2040  0x1fa1  36
InterArPfx *0.0.0.4          10.245.71.3       0x80000002  1891  0x9bef  44
InterArPfx *0.0.0.5          10.245.71.3       0x80000004   388  0xbf22  36
InterArPfx *0.0.0.6          10.245.71.3       0x80000002  1288  0x5c67  36
InterArPfx *0.0.0.7          10.245.71.3       0x80000002  1146  0xf088  44
InterArPfx *0.0.0.8          10.245.71.3       0x80000002  1003  0xd3d6  36
InterArPfx *0.0.0.9          10.245.71.3       0x80000002   988  0xa3df  44
InterArPfx  0.0.0.1          10.245.71.5       0x80000003  2738  0xd70e  36
InterArPfx  0.0.0.2          10.245.71.5       0x80000002  2352  0xa746  36
InterArPfx  0.0.0.3          10.245.71.5       0x80000003   243  0x1cbb  36
InterArPfx  0.0.0.5          10.245.71.5       0x80000002  2211  0xdfcc  36
InterArPfx  0.0.0.6          10.245.71.5       0x80000002  2117  0xafd5  44
InterArPfx  0.0.0.7          10.245.71.5       0x80000002  1964  0x5f8a  36
InterArPfx  0.0.0.8          10.245.71.5       0x80000002  1818  0x427c  36
InterArPfx  0.0.0.9          10.245.71.5       0x80000002  1668  0xdcda  36
InterArPfx  0.0.0.10         10.245.71.5       0x80000002  1518  0x5929  44
InterArPfx  0.0.0.11         10.245.71.5       0x80000002  1368  0xc2af  44
InterArPfx  0.0.0.12         10.245.71.5       0x80000002    93  0x664   44
InterArPfx  0.0.0.1          10.245.71.11      0x80000002  1925  0x7179  36
InterArPfx  0.0.0.2          10.245.71.11      0x80000002  1893  0xab34  36
InterArPfx  0.0.0.3          10.245.71.11      0x80000002  1758  0x447b  36
InterArPfx  0.0.0.4          10.245.71.11      0x80000002  1623  0xd89c  44
InterArPfx  0.0.0.5          10.245.71.11      0x80000002  1323  0x6b78  36
InterArPfx  0.0.0.6          10.245.71.11      0x80000002  1293  0xd6dd  36
InterArPfx  0.0.0.7          10.245.71.11      0x80000002  1158  0x532c  44
InterArPfx  0.0.0.8          10.245.71.11      0x80000002   993  0xa9f7  36
InterArPfx  0.0.0.9          10.245.71.11      0x80000002   858  0x7901  44
InterArRtr  0.0.0.1          10.245.71.5       0x80000001  2743  0xc89b  32
IntraArPfx  0.0.0.1          10.245.71.1       0x80000004   694  0x5f7   76
IntraArPfx *0.0.0.1          10.245.71.3       0x80000005   839  0x5cc1  64
IntraArPfx  0.0.0.1          10.245.71.4       0x80000003   990  0xbc43  64
IntraArPfx  0.0.0.1          10.245.71.11      0x80000003  1023  0xd835  64
    OSPF link state database, area 0.0.0.1
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Router     *0.0.0.1          10.245.71.3       0x80000006   389  0xad3f  56
Router      0.0.0.1          10.245.71.5       0x80000006   393  0xde02  56
Router      0.0.0.1          10.245.71.11      0x80000007   393  0x8853  56
Network     0.0.0.4          10.245.71.11      0x80000003   423  0xfd16  32
InterArPfx *0.0.0.1          10.245.71.3       0x80000002  1588  0xc722  36
InterArPfx *0.0.0.2          10.245.71.3       0x80000002  1438  0x7b2f  44
InterArPfx *0.0.0.3          10.245.71.3       0x80000002   688  0x877   44
InterArPfx  0.0.0.1          10.245.71.5       0x80000003   168  0x30a9  36
InterArPfx  0.0.0.3          10.245.71.5       0x80000002    21  0x6013  44
InterArPfx  0.0.0.1          10.245.71.11      0x80000002  2193  0xa53b  36
InterArPfx  0.0.0.2          10.245.71.11      0x80000002  2059  0x5948  44
InterArPfx  0.0.0.3          10.245.71.11      0x80000002  1593  0x6f5e  36
InterArPfx  0.0.0.4          10.245.71.11      0x80000002  1458  0x7ff9  44
InterArRtr *0.0.0.1          10.245.71.3       0x80000002   538  0x6609  32
```

```
InterArRtr  0.0.0.1          10.245.71.5      0x80000001  2743  0xc89b  32
IntraArPfx *0.0.0.1          10.245.71.3      0x80000004   238  0x4a98  88
IntraArPfx  0.0.0.1          10.245.71.5      0x80000004   318  0x3691  76
IntraArPfx  0.0.0.1          10.245.71.11     0x80000005   258  0x2c5   76
IntraArPfx  0.0.0.5          10.245.71.11     0x80000003   558  0xfa59  44
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Extern      0.0.0.1          10.245.71.4      0x80000003   390  0x8449  36
Extern      0.0.0.1          10.245.71.6      0x80000003   394  0xdc9e  36
    OSPF Link-Local link state database, interface so-0/3/0.0
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Link       *0.0.0.4          10.245.71.3      0x80000003  1888  0x9d6   56
Link        0.0.0.3          10.245.71.11     0x80000004    93  0xed12  56
    OSPF Link-Local link state database, interface so-0/3/2.0
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Link       *0.0.0.6          10.245.71.3      0x80000003  1589  0xb02f  56
Link        0.0.0.2          10.245.71.4      0x80000003   690  0xa07f  56
    OSPF Link-Local link state database, interface t1-0/2/1.0
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Link       *0.0.0.5          10.245.71.3      0x80000003  1738  0x4399  56
Link        0.0.0.3          10.245.71.5      0x80000002  2423  0x618c  56

user@router3>  show ospf3 route
Prefix                                  Path   Route       NH   Metric
                                        type   type        type
10.245.71.1                             Intra  Router      IP   13
  NH-interface (null), NH-addr feee::10:255:71:11
10.245.71.4                             Intra  AS BR       IP   12
  NH-interface so-0/3/2.0
10.245.71.5                             Intra  Area BR     IP   1
  NH-interface t1-0/2/1.0
10.245.71.6                             Inter  AS BR       IP   21
  NH-interface t1-0/2/1.0
10.245.71.11                            Intra  Area BR     IP   1
  NH-interface so-0/3/0.0
10.245.71.11;0.0.0.4                    Intra  Transit     IP   11
  NH-interface so-0/3/0.0
  NH-interface t1-0/2/1.0
9009:1::/64                             Intra  Network     IP   12
  NH-interface so-0/3/2.0
9009:1::2/128                           Intra  Network     IP   0
  NH-interface so-0/3/2.0
9009:2::/64                             Intra  Network     IP   13
  NH-interface so-0/3/0.0
9009:2::2/128                           Intra  Network     IP   1
  NH-interface so-0/3/0.0
9009:3::/64                             Intra  Network     IP   1
  NH-interface so-0/3/0.0
9009:4::/64                             Intra  Network     IP   11
  NH-interface so-0/3/0.0
  NH-interface t1-0/2/1.0
9009:5::/64                             Intra  Network     IP   1
  NH-interface t1-0/2/1.0
9009:6::/64                             Inter  Network     IP   21
  NH-interface t1-0/2/1.0
9009:100::/64                           Ext2   Network     IP   0
  NH-interface so-0/3/2.0
9009:110::/64                           Intra  Network     IP   15
  NH-interface so-0/3/0.0
9009:120::/64                           Intra  Network     IP   13
  NH-interface so-0/3/0.0
```

```
9009:130::/64                              Intra  Network     IP   3
  NH-interface at-1/2/0.0
9009:140::/64                              Intra  Network     IP   4
  NH-interface t1-0/2/1.0
9009:150::/64                              Ext2   Network     IP   0
  NH-interface t1-0/2/1.0
feee::10:255:71:1/128                      Intra  Network     IP   13
  NH-interface so-0/3/0.0
feee::10:255:71:3/128                      Intra  Network     IP   0
  NH-interface lo0.0
feee::10:255:71:4/128                      Intra  Network     IP   12
  NH-interface so-0/3/2.0
feee::10:255:71:5/128                      Intra  Network     IP   1
  NH-interface t1-0/2/1.0
feee::10:255:71:6/128                      Inter  Network     IP   21
  NH-interface t1-0/2/1.0
feee::10:255:71:11/128                     Intra  Network     IP   1
  NH-interface so-0/3/0.0

user@router3>  show interfaces terse
Interface       Admin Link Proto Local                 Remote
...
t1-0/2/1.0      up    up   inet  10.19.5.1/24
                           inet6 9009:5::1/64
                                 fe80::201:afff:fe00:86ca/64
...
so-0/3/0        up    up
so-0/3/0.0      up    up   inet  10.19.3.2/24
                           inet6 9009:3::2/64
                                 fe80::201:afff:fe00:86ca/64
so-0/3/1        up    up
so-0/3/2        up    up
so-0/3/2.0      up    up   inet  10.19.1.2/24
                           inet6 9009:1::2/64
                                 fe80::201:afff:fe00:86ca/64
...
lo0             up    up
lo0.0           up    up   inet  10.245.71.3         --> 0/0
                                 127.0.0.1           --> 0/0
                           inet6 fe80::201:afff:fe00:86ca
                                 feee::10:255:71:3
...
```

To provide a comparison between OSPFv3 show commands and legacy OSPFv2 show commands, the following is some sample output of the OSPFv2 connection between Routers 0 and 3:

```
user@router3> show ospf interface
Interface             State     Area            DR ID           BDR ID
Nbrs
lo0.0                 DRother  0.0.0.0         0.0.0.0         0.0.0.0
0
lo0.0                 DRother  0.0.0.0         0.0.0.0         0.0.0.0
0
so-0/3/2.0            PtToPt   0.0.0.0         0.0.0.0         0.0.0.0
1

user@router3> show ospf neighbor
  Address         Interface             State      ID              Pri  Dead
10.19.1.1        so-0/3/2.0            Full      10.245.71.4      128   38
```

```
user@router3> show ospf database
    OSPF link state database, area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router  *10.245.71.3      10.245.71.3      0x80000002    67  0x2  0x5c45  60
Router   10.245.71.4      10.245.71.4      0x80000002    74  0x2  0x267a  60

user@router3> show ospf route
Prefix               Path   Route       NH   Metric  NextHop       Nexthop

                     Type   Type        Type         Interface     addr/label
10.245.71.4          Intra  Router      IP   1       so-0/3/2.0
10.19.1.0/24         Intra  Network     IP   1       so-0/3/2.0
10.245.71.3/32       Intra  Network     IP   0       lo0.0
10.245.71.4/32       Intra  Network     IP   1       so-0/3/2.0
```

Router 4 Status

```
user@router4> show ospf3 interface
Interface              State     Area            DR-ID           BDR-ID
Nbrs
vl- 10.245.71.3         PtToPt   0.0.0.0         0.0.0.0         0.0.0.0
 1
at-0/3/0.0             PtToPt   0.0.0.1         0.0.0.0         0.0.0.0
0
fe-1/1/0.0             BDR      0.0.0.1         10.245.71.11    10.245.71.5
1
lo0.0                  DRother  0.0.0.1         0.0.0.0         0.0.0.0
0
t1-0/2/1.0             PtToPt   0.0.0.1         0.0.0.0         0.0.0.0
1
fe-0/0/0.0             BDR      0.0.0.2         10.245.71.6     10.245.71.5
1

user@router4>  show ospf3 neighbor
ID               Interface              State     Pri   Dead
10.245.71.3       vl -10.245.71.3         Full        0   32
  Neighbor-address 9009:5::1
10.245.71.11     fe-1/1/0.0             Full      128   37
  Neighbor-address fe80::290:69ff:fea0:809d
10.245.71.3      t1-0/2/1.0             Full      128   32
  Neighbor-address fe80::201:afff:fe00:86ca
10.245.71.6      fe-0/0/0.0             Full      128   35
  Neighbor-address fe80::290:69ff:fe94:c400

user@router4>  show ospf3 database
    OSPF link state database, area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Router      0.0.0.1          10.245.71.1      0x80000004   894  0x8bcd  40
Router      0.0.0.1          10.245.71.3      0x80000006   590  0x2357  72
Router      0.0.0.1          10.245.71.4      0x80000003  1190  0xc29b  40
Router     *0.0.0.1          10.245.71.5      0x80000003  1120  0xf774  40
Router      0.0.0.1          10.245.71.11     0x80000005   623  0x7f6b  56
InterArPfx  0.0.0.1          10.245.71.3      0x80000002  2114  0xa151  36
InterArPfx  0.0.0.2          10.245.71.3      0x80000002  2089  0xb33c  36
InterArPfx  0.0.0.3          10.245.71.3      0x80000002  1940  0x1fa1  36
InterArPfx  0.0.0.4          10.245.71.3      0x80000002  1791  0x9bef  44
InterArPfx  0.0.0.5          10.245.71.3      0x80000004   289  0xbf22  36
InterArPfx  0.0.0.6          10.245.71.3      0x80000002  1188  0x5c67  36
```

```
InterArPfx  0.0.0.7          10.245.71.3      0x80000002  1046  0xf088  44
InterArPfx  0.0.0.8          10.245.71.3      0x80000002   904  0xd3d6  36
InterArPfx  0.0.0.9          10.245.71.3      0x80000002   888  0xa3df  44
InterArPfx *0.0.0.1          10.245.71.5      0x80000003  2636  0xd70e  36
InterArPfx *0.0.0.2          10.245.71.5      0x80000002  2250  0xa746  36
InterArPfx *0.0.0.3          10.245.71.5      0x80000003   141  0x1cbb  36
InterArPfx *0.0.0.5          10.245.71.5      0x80000002  2109  0xdfcc  36
InterArPfx *0.0.0.6          10.245.71.5      0x80000002  2015  0xafd5  44
InterArPfx *0.0.0.7          10.245.71.5      0x80000002  1862  0x5f8a  36
InterArPfx *0.0.0.8          10.245.71.5      0x80000002  1716  0x427c  36
InterArPfx *0.0.0.9          10.245.71.5      0x80000002  1566  0xdcda  36
InterArPfx *0.0.0.10         10.245.71.5      0x80000002  1416  0x5929  44
InterArPfx *0.0.0.11         10.245.71.5      0x80000002  1266  0xc2af  44
InterArPfx *0.0.0.12         10.245.71.5      0x80000001  2641  0x863   44
InterArPfx  0.0.0.1          10.245.71.11     0x80000002  1825  0x7179  36
InterArPfx  0.0.0.2          10.245.71.11     0x80000002  1793  0xab34  36
InterArPfx  0.0.0.3          10.245.71.11     0x80000002  1658  0x447b  36
InterArPfx  0.0.0.4          10.245.71.11     0x80000002  1523  0xd89c  44
InterArPfx  0.0.0.5          10.245.71.11     0x80000002  1223  0x6b78  36
InterArPfx  0.0.0.6          10.245.71.11     0x80000002  1193  0xd6dd  36
InterArPfx  0.0.0.7          10.245.71.11     0x80000002  1058  0x532c  44
InterArPfx  0.0.0.8          10.245.71.11     0x80000002   893  0xa9f7  36
InterArPfx  0.0.0.9          10.245.71.11     0x80000002   758  0x7901  44
InterArRtr *0.0.0.1          10.245.71.5      0x80000001  2641  0xc89b  32
IntraArPfx  0.0.0.1          10.245.71.1      0x80000004   594  0x5f7   76
IntraArPfx  0.0.0.1          10.245.71.3      0x80000005   739  0x5cc1  64
IntraArPfx  0.0.0.1          10.245.71.4      0x80000003   890  0xbc43  64
IntraArPfx  0.0.0.1          10.245.71.11     0x80000003   923  0xd835  64
    OSPF link state database, area 0.0.0.1
 Type       ID               Adv Rtr          Seq       Age  Cksum  Len
Router      0.0.0.1          10.245.71.3      0x80000006   289  0xad3f  56
Router     *0.0.0.1          10.245.71.5      0x80000006   291  0xde02  56
Router      0.0.0.1          10.245.71.11     0x80000007   292  0x8853  56
Network     0.0.0.4          10.245.71.11     0x80000003   322  0xfd16  32
InterArPfx  0.0.0.1          10.245.71.3      0x80000002  1488  0xc722  36
InterArPfx  0.0.0.2          10.245.71.3      0x80000002  1339  0x7b2f  44
InterArPfx  0.0.0.3          10.245.71.3      0x80000002   589  0x877   44
InterArPfx *0.0.0.1          10.245.71.5      0x80000003    66  0x30a9  36
InterArPfx *0.0.0.3          10.245.71.5      0x80000001  2641  0x6212  44
InterArPfx  0.0.0.1          10.245.71.11     0x80000002  2092  0xa53b  36
InterArPfx  0.0.0.2          10.245.71.11     0x80000002  1958  0x5948  44
InterArPfx  0.0.0.3          10.245.71.11     0x80000002  1492  0x6f5e  36
InterArPfx  0.0.0.4          10.245.71.11     0x80000002  1357  0x7ff9  44
InterArRtr  0.0.0.1          10.245.71.3      0x80000002   439  0x6609  32
InterArRtr *0.0.0.1          10.245.71.5      0x80000001  2641  0xc89b  32
IntraArPfx  0.0.0.1          10.245.71.3      0x80000004   139  0x4a98  88
IntraArPfx *0.0.0.1          10.245.71.5      0x80000004   216  0x3691  76
IntraArPfx  0.0.0.1          10.245.71.11     0x80000005   157  0x2c5   76
IntraArPfx  0.0.0.5          10.245.71.11     0x80000003   457  0xfa59  44
    OSPF link state database, area 0.0.0.2
 Type       ID               Adv Rtr          Seq       Age  Cksum  Len
Router     *0.0.0.1          10.245.71.5      0x80000004   366  0x252e  40
Router      0.0.0.1          10.245.71.6      0x80000004  1492  0x64d   40
Network     0.0.0.2          10.245.71.6      0x80000003   892  0xfd22  32
InterArPfx *0.0.0.1          10.245.71.5      0x80000003  2636  0xd70e  36
InterArPfx *0.0.0.2          10.245.71.5      0x80000002  2179  0xa746  36
InterArPfx *0.0.0.3          10.245.71.5      0x80000002  2091  0xf3ba  36
InterArPfx *0.0.0.4          10.245.71.5      0x80000002  1938  0xc3c3  44
InterArPfx *0.0.0.5          10.245.71.5      0x80000002  1791  0x7378  36
InterArPfx *0.0.0.6          10.245.71.5      0x80000002  1641  0x566a  36
InterArPfx *0.0.0.7          10.245.71.5      0x80000002  1491  0xf0c8  36
```

```
InterArPfx *0.0.0.8          10.245.71.5      0x80000002  1341  0x6d17  44
InterArPfx *0.0.0.9          10.245.71.5      0x80000002  1191  0xd69d  44
InterArPfx *0.0.0.10         10.245.71.5      0x80000002  1049  0x6776  36
InterArPfx *0.0.0.11         10.245.71.5      0x80000002   979  0x1b83  44
InterArPfx *0.0.0.12         10.245.71.5      0x80000002   908  0x6772  36
InterArPfx *0.0.0.13         10.245.71.5      0x80000002   891  0x1b7f  44
InterArPfx *0.0.0.14         10.245.71.5      0x80000002   815  0x3195  36
InterArPfx *0.0.0.15         10.245.71.5      0x80000002   738  0x4131  44
InterArPfx *0.0.0.16         10.245.71.5      0x80000002   662  0x7fef  44
InterArRtr *0.0.0.1          10.245.71.5      0x80000002   591  0x6408  32
IntraArPfx  0.0.0.1          10.245.71.6      0x80000005  1192  0x42b9  52
IntraArPfx  0.0.0.3          10.245.71.6      0x80000003   592  0xfe61  44
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Extern      0.0.0.1          10.245.71.4      0x80000003   290  0x8449  36
Extern      0.0.0.1          10.245.71.6      0x80000003   292  0xdc9e  36
    OSPF Link-Local link state database, interface fe-0/0/0.0
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Link       *0.0.0.4          10.245.71.5      0x80000003   516  0x3b6   56
Link        0.0.0.2          10.245.71.6      0x80000004  1792  0x782   56
    OSPF Link-Local link state database, interface fe-1/1/0.0
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Link       *0.0.0.5          10.245.71.5      0x80000003   441  0x40dc  56
Link        0.0.0.4          10.245.71.11     0x80000004   592  0x73ab  56
    OSPF Link-Local link state database, interface t1-0/2/1.0
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Link        0.0.0.5          10.245.71.3      0x80000003  1639  0x4399  56
Link       *0.0.0.3          10.245.71.5      0x80000002  2321  0x618c  56

user@router4>  show ospf3 route
Prefix                                     Path   Route       NH   Metric
                                           type   type        type
10.245.71.1                                Intra  Router      IP   14
  NH-interface (null), NH-addr feee::10:255:71:3
10.245.71.3                                Intra  Area BR     IP   1
  NH-interface t1-0/2/1.0
10.245.71.4                                Intra  AS BR       IP   13
  NH-interface t1-0/2/1.0
10.245.71.6                                Intra  AS BR       IP   20
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe94:c400
10.245.71.6;0.0.0.2                        Intra  Transit     IP   20
  NH-interface fe-0/0/0.0
10.245.71.11                               Intra  Area BR     IP   2
  NH-interface t1-0/2/1.0
10.245.71.11;0.0.0.4                       Intra  Transit     IP   10
  NH-interface fe-1/1/0.0
9009:1::/64                                Intra  Network     IP   13
  NH-interface t1-0/2/1.0
9009:1::2/128                              Intra  Network     IP   1
  NH-interface t1-0/2/1.0
9009:2::/64                                Intra  Network     IP   14
  NH-interface t1-0/2/1.0
9009:2::2/128                              Intra  Network     IP   2
  NH-interface t1-0/2/1.0
9009:3::/64                                Intra  Network     IP   2
  NH-interface t1-0/2/1.0
9009:4::/64                                Intra  Network     IP   10
  NH-interface fe-1/1/0.0
9009:5::/64                                Intra  Network     IP   1
  NH-interface t1-0/2/1.0
9009:6::/64                                Intra  Network     IP   20
```

```
  NH-interface fe-0/0/0.0
9009:100::/64                                Ext2   Network     IP   0
  NH-interface t1-0/2/1.0
9009:110::/64                                Intra  Network     IP   16
  NH-interface t1-0/2/1.0
9009:120::/64                                Intra  Network     IP   14
  NH-interface t1-0/2/1.0
9009:130::/64                                Intra  Network     IP   4
  NH-interface t1-0/2/1.0
9009:140::/64                                Intra  Network     IP   3
  NH-interface at-0/3/0.0
9009:150::/64                                Ext2   Network     IP   0
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe94:c400
feee::10:255:71:1/128                        Intra  Network     IP   14
  NH-interface t1-0/2/1.0
feee::10:255:71:3/128                        Intra  Network     IP   1
  NH-interface t1-0/2/1.0
feee::10:255:71:4/128                        Intra  Network     IP   13
  NH-interface t1-0/2/1.0
feee::10:255:71:5/128                        Intra  Network     IP   0
  NH-interface lo0.0
feee::10:255:71:6/128                        Intra  Network     IP   20
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe94:c400
feee::10:255:71:11/128                       Intra  Network     IP   2
  NH-interface t1-0/2/1.0

user@router4>  show interfaces terse
Interface       Admin Link Proto Local                 Remote
fe-0/0/0        up    up
fe-0/0/0.0      up    up   inet  10.19.6.1/24
                           inet6 9009:6::1/64
                                 fe80::290:69ff:fe98:9000/64
...
t1-0/2/1        up    up
t1-0/2/1.0      up    up   inet  10.19.5.2/24
                           inet6 9009:5::2/64
                                 fe80::2a0:a5ff:fe3d:b63/64
...
fe-1/1/0        up    up
fe-1/1/0.0      up    up   inet  10.19.4.2/24
                           inet6 9009:4::2/64
                                 fe80::290:69ff:fe98:909d/64
...
lo0             up    up
lo0.0           up    up   inet  10.245.71.5         --> 0/0
                                 127.0.0.1           --> 0/0
                           inet6 fe80::2a0:a5ff:fe3d:b63
                                 feee::10:255:71:5
...
```

Router 5 Status

```
user@router5> show ospf3 interface
Interface             State     Area            DR-ID           BDR-ID
Nbrs
fe-0/0/0.0            DR       0.0.0.2         10.245.71.6     10.245.71.5
1
lo0.0                 DRother  0.0.0.2         0.0.0.0         0.0.0.0
0
```

```
user@router5> show ospf3 neighbor
ID              Interface             State     Pri   Dead
10.245.71.5     fe-0/0/0.0            Full      128   33
  Neighbor-address fe80::290:69ff:fe98:9000

user@router5>  show ospf3 database
    OSPF link state database, area 0.0.0.2
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Router      0.0.0.1          10.245.71.5      0x80000003  2237  0x272d  40
Router     *0.0.0.1          10.245.71.6      0x80000004  1082  0x64d   40
Network    *0.0.0.2          10.245.71.6      0x80000003   482  0xfd22  32
InterArPfx  0.0.0.1          10.245.71.5      0x80000003  2228  0xd70e  36
InterArPfx  0.0.0.2          10.245.71.5      0x80000002  1771  0xa746  36
InterArPfx  0.0.0.3          10.245.71.5      0x80000002  1683  0xf3ba  36
InterArPfx  0.0.0.4          10.245.71.5      0x80000002  1530  0xc3c3  44
InterArPfx  0.0.0.5          10.245.71.5      0x80000002  1383  0x7378  36
InterArPfx  0.0.0.6          10.245.71.5      0x80000002  1233  0x566a  36
InterArPfx  0.0.0.7          10.245.71.5      0x80000002  1083  0xf0c8  36
InterArPfx  0.0.0.8          10.245.71.5      0x80000002   933  0x6d17  44
InterArPfx  0.0.0.9          10.245.71.5      0x80000002   783  0xd69d  44
InterArPfx  0.0.0.10         10.245.71.5      0x80000002   641  0x6776  36
InterArPfx  0.0.0.11         10.245.71.5      0x80000002   570  0x1b83  44
InterArPfx  0.0.0.12         10.245.71.5      0x80000002   500  0x6772  36
InterArPfx  0.0.0.13         10.245.71.5      0x80000002   483  0x1b7f  44
InterArPfx  0.0.0.14         10.245.71.5      0x80000002   406  0x3195  36
InterArPfx  0.0.0.15         10.245.71.5      0x80000002   330  0x4131  44
InterArPfx  0.0.0.16         10.245.71.5      0x80000002   253  0x7fef  44
InterArRtr  0.0.0.1          10.245.71.5      0x80000002   183  0x6408  32
IntraArPfx *0.0.0.1          10.245.71.6      0x80000005   782  0x42b9  52
IntraArPfx *0.0.0.3          10.245.71.6      0x80000003   182  0xfe61  44
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Extern      0.0.0.1          10.245.71.4      0x80000002  1082  0x8648  36
Extern     *0.0.0.1          10.245.71.6      0x80000002  1682  0xde9d  36
    OSPF Link-Local link state database, interface fe-0/0/0.0
 Type       ID               Adv Rtr           Seq      Age  Cksum  Len
Link        0.0.0.4          10.245.71.5      0x80000003   108  0x3b6   56
Link       *0.0.0.2          10.245.71.6      0x80000004  1382  0x782   56
user@router5>  show ospf3 route
Prefix                                  Path   Route       NH   Metric
                                        type   type        type
10.245.71.4                             Inter  AS BR       IP   33
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
10.245.71.5                             Intra  Area BR     IP   20
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
10.245.71.6;0.0.0.2                     Intra  Transit     IP   20
  NH-interface fe-0/0/0.0
9009:1::/64                             Inter  Network     IP   33
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
9009:1::2/128                           Inter  Network     IP   21
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
9009:2::/64                             Inter  Network     IP   34
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
9009:2::2/128                           Inter  Network     IP   22
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
9009:3::/64                             Inter  Network     IP   22
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
9009:4::/64                             Inter  Network     IP   30
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
9009:5::/64                             Inter  Network     IP   21
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
```

```
9009:6::/64                                Intra  Network     IP   20
  NH-interface fe-0/0/0.0
9009:100::/64                              Ext2   Network     IP   0
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
9009:110::/64                              Inter  Network     IP   36
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
9009:120::/64                              Inter  Network     IP   34
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
9009:130::/64                              Inter  Network     IP   24
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
9009:140::/64                              Inter  Network     IP   23
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
feee::10:255:71:1/128                      Inter  Network     IP   34
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
feee::10:255:71:3/128                      Inter  Network     IP   21
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
feee::10:255:71:4/128                      Inter  Network     IP   33
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
feee::10:255:71:5/128                      Inter  Network     IP   20
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000
feee::10:255:71:6/128                      Intra  Network     IP   0
  NH-interface lo0.0
feee::10:255:71:11/128                     Inter  Network     IP   22
  NH-interface fe-0/0/0.0, NH-addr fe80::290:69ff:fe98:9000

user@router5>  show interfaces terse
Interface       Admin Link Proto Local                 Remote
...
fe-0/0/0        up    up
fe-0/0/0.0      up    up   inet  10.19.6.2/24
                           inet6 9009:6::2/64
                                 fe80::290:69ff:fe94:c400/64
...
lo0             up    up
lo0.0           up    up   inet  10.245.71.6         --> 0/0
                                 127.0.0.1           --> 0/0
                           inet6 fe80::2a0:a5ff:fe12:33a2
                                 feee::10:255:71:6
...
```

For More Information

For additional information about OSPFv3 for IPv6, see the following resources:

- *JUNOS Routing Protocols Configuration Guide*
- RFC 2328, *OSPF Version 2*
- RFC 2460, *Internet Protocol, Version 6 (IPv6) Specification*
- RFC 2740, *OSPF for IPv6*
- RFC 3513, *IP Version 6 Addressing Architecture*

Revision History

13 April 2009—9.5R1 Release. Roy Spencer.

15 January 2009—9.4R1 Release. Roy Spencer.

10 October 2008—9.3R1 Release. Roy Spencer.

10 July 2008—9.2R1 Release. Roy Spencer.

10 April 2008—9.1R1 Release. Roy Spencer.

1 February 2008—9.0R1 Release. Fawn Damitio.

17 August 2007—Added support for logical router administrators. 8.5 beta release. Fawn Damitio.

29 June 2007—8.4R1 Release. Fawn Damitio.

27 March 2007—8.3R1 Release. Fawn Damitio.

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

15 September 2006—8.1R1 Release. Richard Hendricks.

29 June 2006—8.0R1 Release. Richard Hendricks.

27 March 2006—7.6R1 Release. Richard Hendricks.

9 January 2006—7.5R1 Release. Richard Hendricks.

14 September 2005—7.4R1 Release. Richard Hendricks.

13 June 2005—7.3R1 Release. Richard Hendricks.

5 April 2005—Added support for the J-series Services Routers and included a cross-reference to the IPSec chapter, 7.2R1 Release. Richard Hendricks.

2 February 2005—7.1R1 Release. Richard Hendricks.

6 October 2004—7.0R1 Release. Richard Hendricks.

6 July 2004—6.4R1 Release. Richard Hendricks.

5 April 2004—6.3R1 Release. Richard Hendricks.

22 December 2003—6.2R1 Release. Richard Hendricks.

22 September 2003—6.1R1 Release. Richard Hendricks.

30 June 2003—6.0R1 Release. Richard Hendricks.

2 April 2003—5.7R1 Release. Richard Hendricks.

27 December 2002—5.6R1 Release. Richard Hendricks.

30 September 2002—5.5R1 Release. Richard Hendricks.

27 August 2002—Initial document written. Richard Hendricks.

Chapter 9

Multitopology Routing

This feature guide chapter contains the following sections:

Overview

Multitopology Routing (MTR) enables you to configure class-based forwarding for different types of traffic, such as voice, video, and data. Each type of traffic is defined by a topology that is used to create a new routing table for that topology. MTR provides the ability to generate forwarding tables based on the resolved entries in the routing tables for the custom topologies you create. In this way, packets of different classes can be routed independently from one another.

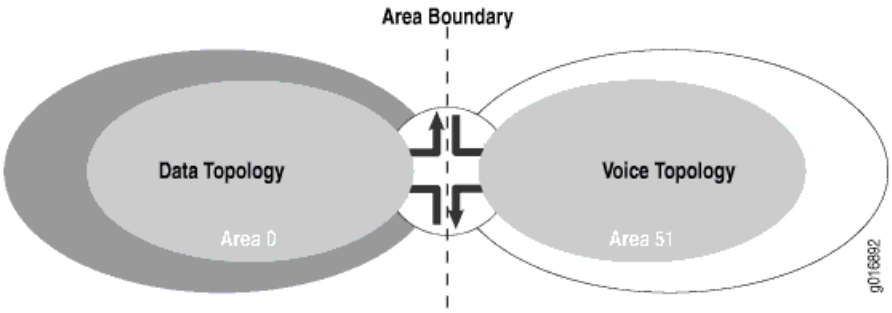
To run MTR, you must configure IP routing. MTR supports OSPFv2, static routes, and BGP. You must configure an interior gateway protocol (IGP), such as OSPF v2 or static routing, Configure BGP to add routes learned through BGP to the appropriate custom topologies. MTR also supports filter-based forwarding, which enables you to match traffic on the ingress interface with a specific type of forwarding class and then forward that traffic to the specified topology.

OSPF in MTR uses a single instance of OSPF to carry connectivity and IP reachability information for different topologies. That information is used to calculate shortest-path first (SPF) trees and routing tables. OSPF for MTR supports protocol extensions that include metrics that correspond to different topologies for link and prefix reachabilty

information. The type-of-service (TOS) metric field is used to advertise the topology-specific metric for links and prefixes belonging to that topology. The TOS field is redefined as MT-ID in the payload of router, summary, and Type 5 and Type 7 AS-external link-state advertisements (LSAs).

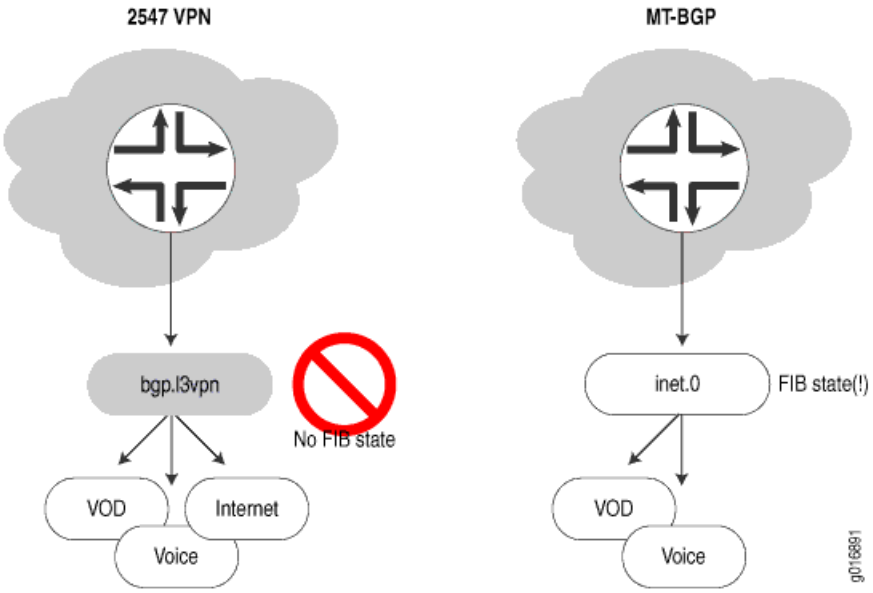
Under MTR, each OSPF interface continues to belong to a single area. Therefore, by default, all topologies share the same area boundaries. As a result, attributes of an area, such as stubbiness, are independent of the topology. By default, all topologies configured for OSPF are enabled on all interfaces. However, you can disable one or more configured topologies on an interface. You can thus allocate an interface for a specific topology. In Figure 19 on page 268, area 51 includes an interface that is uniquely allocated to voice traffic, and area 0 includes an interface that is uniquely allocated to data traffic. Each topology thus corresponds to a different OSPF area that shares a boundary.

Figure 19: MT-OSPF Area Boundary



BGP in MTR provides the ability to resolve BGP routes against configured topologies. An inbound policy is used to select routes for inclusion in the appropriate routing tables for the topologies. The default behavior for virtual private networks (VPNs) that use Multiprotocol Label Switching (MPLS) for forwarding packets over the backbone and BGP for distributing routes over the backbone is to place BGP route updates in the bgp.l3vpn routing table. Figure 20 on page 269 shows a BGP peer operating in an environment that conforms with RFC 2547, *BGP/MPLS VPNs*. The figure shows how a BGP peer configured for MTR performs secondary route resolution.

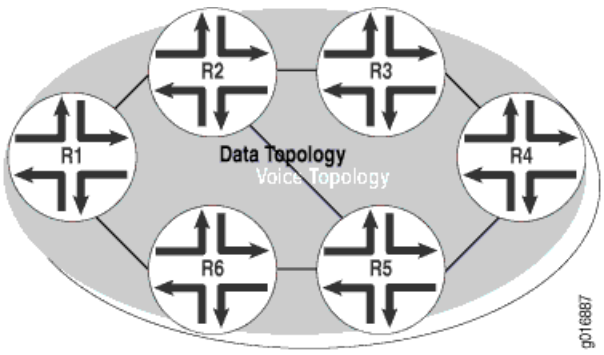
Figure 20: BGP Route Resolution



The BGP peer in a standard VPN topology places prefixes for routes it learns in the bgp.l3vpn routing table, which does not result in automatic updates to the forwarding table. Under BGP in MTR, when BGP receives a route from a peer it attempts to resolve that route against a route in the inet.0 routing table. If the route is resolved, it is placed in that table, which generates a forwarding state. If you have configured a community target identifier that matches the import policy for the topology, routing and forwarding states are added to the tables for the topology.

Because MTR provides support for BGP to perform secondary route resolution, as Figure 21 on page 269 below shows, MTR is able to create two distinct network paths for each type of traffic. Each router advertises BGP routes that need to be resolved against the interior gateway protocol (IGP) routes for each topology. Based on the IGP metrics configured for each topology, for all routes that originate from Router 4 (R4), the upper path between R1 and R4, which traverses R2 and R3, is selected for voice traffic, whereas the lower path between R1 and R4, which traverses R5 and R6, is selected for data traffic.

Figure 21: Route Resolution for MTR



System Requirements

To implement MTR, your system must meet these minimum requirements:

- JUNOS Release 9.0 or later
- Two Juniper Networks M-series, MX-series, or T-series routing platforms

Terms and Acronyms

T

topology A subset of links in a network for which a separate set of routes is calculated. Those routes are installed in a routing table created specifically for a configured topology used to make routing and forwarding decisions.

Configuring Multitopology Routing

To implement MTR, you must configure the following:

- Configuring Topologies on page 270
- Configuring Filter-Based Forwarding on page 271
- Configuring BGP for Multitopology Routing on page 271
- Option: Configuring OSPF for Multitopology Routing on page 272
- Option: Configuring Static Routes for Multitopology Routing on page 272
- Option: Configuring Route Resolution Policy on page 273

Configuring Topologies

You must configure one or more topologies. For each topology, you specify a string value that defines the type of traffic as well as an interface family. You can also enable a topology for IPv4 multicast traffic by including the ipv4-multicast statement. To configure a topology, include the topologies statement at the [edit routing-options hierarchy level:

```
[edit routing-options]
topologies {
   family inet { # inet6 is also supported, but you must use static routes as the IGP.
      topology voice; # This action creates a routing table called :voice.inet.0
         # for all routes destined for the voice topology. A default topology is also
         # automatically created. Default topology routes are added to the inet.0
         # routing table.
   }
}
```

Configuring Filter-Based Forwarding

Configure a firewall filter that forces a lookup against the different routing tables. Any routes that match the forwarding class specified and then match a specified topology are installed in the routing table for that topology. To configure a firewall filter for MTR that performs filter-based forwarding, include the following configuration at the [edit firewall] hierarchy level:

```
[edit firewall]
family inet { # inet6 is also supported.
   filter topology-selection {
      term ef {
         from {
            forwarding-class expedited-forwarding; # The following class types are also
               # supported: assured-forwarding, best-effort, and network-control.
         }
         then {
            topology voice; # Specify the name of a configured topology.
            accept;
         }
      }
   }
}
```

You must apply the filter to an ingress interface. Include the following statements at the [edit interfaces] hierarchy level to apply the filter to an interface:

```
[edit interfaces]
fe-2/2/1 {
   unit 0 {
      family inet {
         filter {
            input topology-selection; # Specify the name of the filter configured under
               # the [edit firewall] hierarchy
         }
      }
   }
}
```

Configuring BGP for Multitopology Routing

Configure BGP to add routes learned through BGP into a configured topology and the default topology, which is created automatically. Routes for the default topology are installed in the inet.0 routing table. To configure BGP for Multitopology Routing, include the following statements:

```
[edit protocols bgp]
group internal {
   type internal;
   family inet {
      unicast {
         topology voice;
```

```
            community target :1:1; # Any route that has :1:1 as its target destination is
               # installed in the routing table for the voice topology. All received routes
               # are also automatically installed in the default topology.
         }
      }
   }
}
```

Option: Configuring OSPF for Multitopology Routing

To implement MTR, you must configure a interior gateway protocol (IGP) to route local network traffic. MTR supports both OSPFv2 and static routes. Only static routes support IPv6 addresses since MTR does not support OSPFv3. Configure OSPF to add routes from the default topology to the routing table for the specified topology. To enable OSPF for MTR, include the **topology** statement at the **[edit protocols ospf]** hierarchy level:

```
[edit ospf protocols]
topology voice { # Specify a topology name configured under the [edit routing-options]
   # hierarchy level.
   topology-id 127; # Specify a topology identifier from 32 through 127.
}
```

Optionally, you can configure a specific metric for a topology for any interface on which OSPF has been enabled. Any topology-specific metric that you configure applies to routes advertised from that interface that belong only to that topology. To configure a topology-specific metric for an OSPF interface, include the following statements at the **[edit protocols ospf]** hierarchy level:

```
[edit protocols ospf]
area 0.0.0.0 {
   interface fe-2/2/1 {
      metric 10; # You can specify a metric for the interface that overrides the default
         # value of 1.
      }
      topology voice {
         metric 15; # Specify a topology-specific metric from 1 through 65,535.
      }
   }
}
```

Option: Configuring Static Routes for Multitopology Routing

To configure a static route for MTR, you must specify the name of the routing table for the topology. Static routes for MTR support IPv4 and IPv6 addresses. To configure a static route for MTR, include the **rib** statement at the **[edit routing-options]** hierarchy level:

```
[edit routing-options]
rib :voice.inet6.0 { # Specify the routing-table name for the voice topology configured
   # at the [edit routing-options topologies] hierarchy level.
   route 200::a4:0/126 next-hop 200::c0a8:1df;
}
```

Option: Configuring Route Resolution Policy

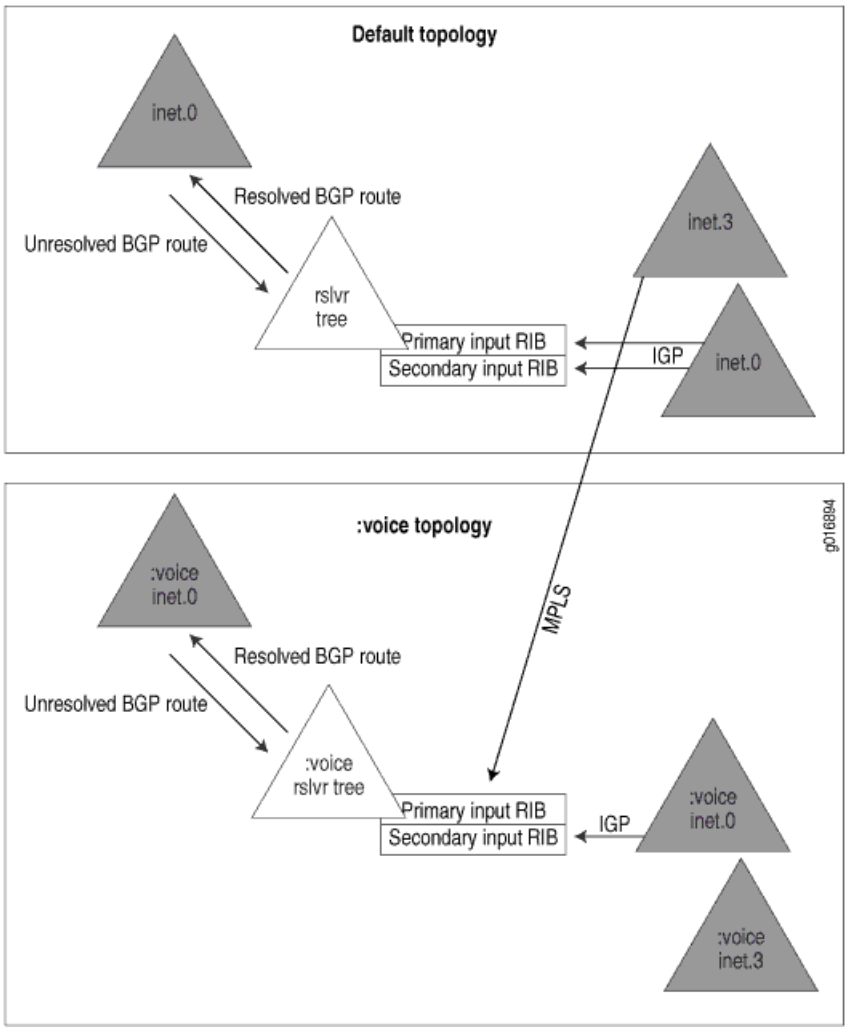
You can optionally configure a route resolution policy so that a routing table accepts routes from specific routing tables. For Multitopology Routing, you might want to configure a policy for the voice topology for IPv4 unicast traffic, for example, :voice.inet.0 to resolve routes through the inet.3 routing table. You might also want to override the default policy and not have the inet.0 routing table use the inet.3 routing table for route resolution. To configure a route resolution policy, include the resolution statement at the [edit routing-options] hierarchy level:

```
[edit routing-options]
resolution {
   rib inet.0 { # Specify the name of the routing table you want to modify.
      resolution-ribs inet.0; # Specify use of the inet.0 table to resolve routes rather
         # than the default policy, which uses the inet.3 table.
   }
   rib :voice.inet.0 { # Specify that the routing table for the voice topology for IPv4
      # unicast traffic be modified.
      resolution-ribs [ inet.3 :voice.inet.0 ]; # Specify that routes be resolved
         # using the inet.3 routing table for the voice topology.
   }
}
```

Example: Multitopology Routing Configuration

In this example, an MPLS network is running RSVP label-switched paths (LSPs) in the core. The network carries both best effort (BE) and expedited forwarding (EF) traffic, and the same destination prefixes are used for both types of traffic. A voice topology is created to enable the network to send EF traffic over the LSPs, but permit BE traffic to traverse the IP path. The voice, or EF traffic, is placed in the voice topology. The data, or BE traffic, is placed in the default topology. The voice and default topologies each create separate routing tables for storing routes, and each routing table creates a separate forwarding table. For each destination prefix, a different route is added to each topology-specific routing table. These routes are BGP routes. You configure filter-based forwarding so that the destination lookup is done in the two topology-specific routing tables based on the DSCP marking of the incoming packet. As Figure 22 on page 274 shows, MTR thus enables you to build two routes to the destination prefixes, one using LSP next hops for voice traffic and one using IP next hops for data traffic. The protocol next hop is resolved differently for each type of traffic. For voice EF traffic, routes from the inet.3 routing table are taken into account, and for BE data traffic, only routes from the inet.0 routing table are taken into account.

Figure 22: Route Resolution in Multitopology Routing



Configure the interfaces:

```
[edit]
interfaces {
   so-0/0/1 {
      unit 0 {
         family inet {
            address 1.13.1.1/24
         }
         family iso;
         family mpls;
      }
   }
   fe-2/2/0 {
      unit 0 {
         family inet {
            address 1.12.1.1/24
         }
```

```
            family iso;
            family mpls;
        }
    }
    fe/2/2/1 {
    unit 0 {
        family inet {
            filter {
                input topo_selection; # Apply a firewall filter on the ingress interface.
                    # This filter performs filter-based forwarding for the voice topology.
            }
            family iso;
            family mpls;
        }
    }
}
```

Configure the voice topology. Configure a route resolution policy so that IPv4 routes for data traffic are resolved through the inet.0 routing table, which functions as the routing table for the default topology. Configure a route resolution policy so that routes for voice traffic are resolved through the routing table for the voice topology (:voice.inet.0) and the MPLS routing table (inet.3).

```
[edit]
routing-options {
    autonomous-system 65300;
    resolution {
        rib inet.0 {
            resolution-ribs inet.0; # Specify use of the inet.0 routing table to resolve
                # IPv4 data traffic. This action prevents this traffic from being resolved using
                # the MPLS routing table (inet.3).
        }
        rib :voice.inet.0 {
            resolution-ribs [ inet.3 :voice.inet.0 ]; # Specify use of the MPLS routing
                # table (inet.3) and the routing table for the voice topology (:voice.inet.0)
                # to resolve IPv4 voice traffic. This action prevents voice traffic from being
                #resolved using the inet.0 routing table.
        }
    }
    topologies {
        family inet {
            topology voice;
        }
    }
}
```

Configure MPLS using RSVP label-switched paths. Configure BGP so that routes learned through BGP are installed in the appropriate routing table. In this example, the topology statement is used to install BGP routes for voice traffic into the routing table for the voice topology (:voice.inet.0). This action overrides the default behavior to resolve BGP routes only through the inet.0 or inet.3 routing tables. Configure an interior gateway protocol (IGP). In this example, you configure OSPF.

```
[edit]
protocols {
```

```
    rsvp {
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
    mpls {
        label-switched-path to_r3 {
            to 10.255.14.222;
            primary test;
        }
        path test {
            1.12.1.2 strict;
        }
        interface all;
    }
    bgp {
        group int {
            type internal;
            local-address 10.255.14.223;
            family inet {
                unicast {
                    topology voice {
                        community 70:1;
                    }
                }
            }
            neighbor 10.255.14.220;
            neighbor 10.255.14.218;
            neighbor 10.255.14.222;
        }
    }
    ospf {
        topology voice topology-id 32;
        traffic-engineering;
        area 0.0.0.0 {
            interface lo0.0 {
                passive;
            }
            interface all;
            interface fxp0.0 {
                disable;
            }
            interface fe-2/2/1.0 {
                metric 1;
            }
            interface fe-2/2/0.0 {
                metric 1;
            }
            interface so-0/0/1.0 {
                metric 1;
            }
        }
    }
}
```

Configure a class-of-service classifier on the ingress interface. In this example, the classifier type is inet-precedence, which evaluates incoming IPv4 packets and requires only the upper three bits of the DSCP field.

```
[edit]
class-of-service {
   interfaces {
      fe-2/2/1 {
         unit 0 {
            classifiers {
               inet-precedence default;
            }
         }
      }
   }
}
```

Configure filter-based forwarding. This filter is applied to the ingress interface. Traffic marked for expedited forwarding is forwarded to the routing table for the voice topology. All other traffic is routing to the routing table for the default topology.

```
[edit]
firewall {
   family inet {
      filter topo_selection {
         term ef {
            from {
               forwarding-class expedited-forwarding;
            }
            then {
               topology voice; # Forward expedited-forwarding traffic to the routing
                  # table for the voice topology (:voice.inet.0).
               accept;
            }
         }
         term default {
            then accept; # Forward all other traffic to the routing table for the default
               # topology (inet.0).
         }
      }
   }
}
```

Verifying Your Work

To verify proper operation of Multitopology Routing, use the following commands:

- show route summary
- show route table *routing-table-name*
- show route rib-groups

For More Information

For additional information about MTR, see the following resources:

- *JUNOS Routing Protocols Configuration Guide*
- RFC 4915, *Multi-Topology (MT) Routing in OSPF*

Revision History

13 April 2009—9.5R1 Release. Roy Spencer.

15 January 2009—9.4R1 Release. Roy Spencer.

10 October 2008—9.3R1 Release. Roy Spencer.

10 July 2008—9.2R1 Release. Roy Spencer.

10 April 2008—9.1R1 Release. Ines Salazar.

1 February 2008—Initial document written. Ines Salazar.

Part 3
Services Interfaces

Chapter 10

Flow Monitoring

This feature guide covers the following topics:

Overview

The flow monitoring application performs traffic flow monitoring and enables lawful interception of packets transiting between two routing platforms. Traffic flows can either be passively monitored by an offline routing platform or actively monitored by a routing platform participating in the network.

Using a Juniper Networks routing platform, a selection of Physical Interface Cards (PICs) for M-series and T-series routing platforms—including the Monitoring Services PIC, Monitoring Services II PIC, Adaptive Services PIC, and MultiServices PICs—and other networking hardware, you can monitor traffic flow and export the monitored traffic. Monitoring traffic allows you to do the following:

- Gather and export detailed information about traffic flows between source and destination nodes in your network.
- Sample all incoming traffic on the monitoring interface and present the data in record format.
- Encrypt or tunnel outgoing records, intercepted traffic, or both.
- Direct filtered traffic to different packet analyzers and present the data in its original format.
- Intercept unwanted traffic, discard it, and perform accounting on the discarded packets.

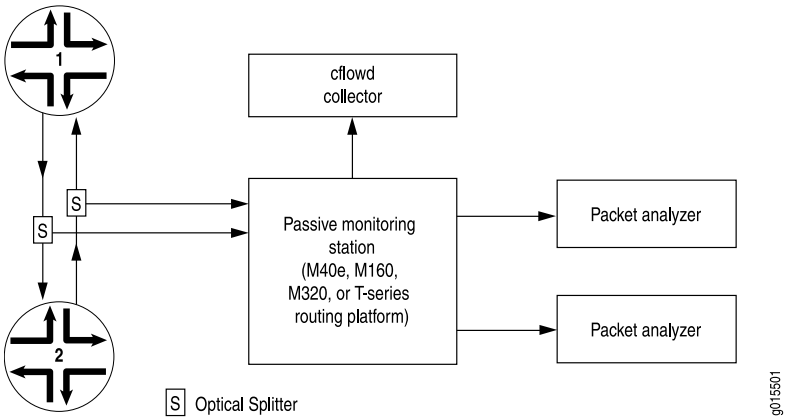
There are two main types of flow monitoring:

Passive Flow Monitoring

Flow monitoring version 5 supports passive flow monitoring. Versions 8 and 9 do not support passive flow monitoring.

The M40e, M160, M320, MX-series, or T-series routing platform that is used for passive flow monitoring does not route packets from monitored interfaces, nor does it run any routing protocols related to those interfaces; it only passes along intercepted traffic and receives traffic flows. Figure 23 on page 284 shows a typical topology for the passive flow monitoring application.

Figure 23: Passive Flow Monitoring Application Topology



Traffic travels normally between Router 1 and Router 2. To redirect IPv4 traffic, you insert an optical splitter on the interface between these two routers. The optical splitter copies and redirects the traffic to the monitoring station. The optical cable connects only the receive port on the monitoring station, never the transmit port. This configuration allows the monitoring station to receive traffic only from the router being monitored but never to transmit it back.

If you are monitoring traffic flow, the Internet Processor II ASIC in the routing platform forwards a copy of the traffic to the Monitoring Services or Monitoring Services II PIC in the monitoring station. If there is more than one Monitoring Services PIC installed, the monitoring station distributes the load of the incoming traffic across the multiple PICs. The Monitoring Services PICs generate flow records in version 5 format, and the records are exported to the flow collector.

When you are performing lawful interception of packets, the Internet Processor II ASIC filters the incoming traffic and forwards it to the Tunnel Services PIC. Filter-based forwarding is then applied to direct the traffic to the packet analyzers. Optionally, the intercepted traffic or the flow records can be encrypted by the ES PIC and then sent to their destination. With additional configuration, flow records can be processed by a flow collector and flows can be captured dynamically.

With MPLS passive monitoring, the routing platform can process MPLS packets with label values that do not have corresponding entries in the mpls.0 routing table. You can divert these unrecognized MPLS packets, remove the MPLS labels, and redirect

the underlying IPv4 packets. This is equivalent to a default route for MPLS packets or a promiscuous label. Because this application does not use a Monitoring Services PIC, see the *JUNOS MPLS Applications Configuration Guide* for more information about MPLS passive monitoring.

Active Flow Monitoring

Flow monitoring versions 5, 8, and 9 support active flow monitoring. For active flow monitoring, the monitoring station participates in the network as an active routing platform. The major actions the routing platform can perform during active flow monitoring are as follows:

- Sampling—The routing platform selects and analyzes only a portion of the traffic.
- Sampling with templates—The routing platform selects, analyzes, and arranges a portion of the traffic into templates.
- Port mirroring—The routing platform copies entire packets and sends the copies to another interface.
- Multiple port mirroring—The routing platform sends multiple copies of monitored packets to multiple export interfaces with the next-hop-group statement at the [edit forwarding-options] hierarchy level.
- Discard accounting—The routing platform accounts for selected traffic before discarding it. Such traffic is not forwarded out of the routing platform. Instead, the traffic is quarantined and deleted.
- Flow-tap processing—The routing platform processes requests for active flow monitoring dynamically by using the Dynamic Tasking Control Protocol (DTCP).

System Requirements

This section describes the system requirements for flow monitoring and contains the following sections:

Passive Flow Monitoring System Requirements

To perform passive flow monitoring, your system must meet these minimum requirements:

- JUNOS Release 8.5 or later for passive flow monitoring support on the MX-series MultiServices routers.
- JUNOS Release 8.4 or later for passive flow monitoring support on the MultiServices 400 PIC (Type 2).
- JUNOS Release 7.6 or later to clear error and flow statistics with the clear passive-monitoring statistics command

- JUNOS Release 7.5 or later for support of the dynamic flow capture (DFC) Management Information Base (MIB)
- JUNOS Release 7.4 or later for dynamic flow capture on Monitoring Services III PICs installed in T-series and M320 routing platforms, and port mirroring of IPv6 packets
- JUNOS Release 7.3 or later for passive flow monitoring on selected Ethernet-based interfaces and filter-based forwarding on output interfaces
- JUNOS Release 7.1 or later for passive flow monitoring and flow collection services on Monitoring Services II PICs installed in T-series and M320 routing platforms
- JUNOS Release 6.4 or later for support of the next-hop IP address field in flow monitoring version 5 records
- JUNOS Release 6.2 or later for ATM2 intelligent queuing (IQ) interface passive monitoring, flow collection services, and MPLS label stripping
- JUNOS Release 6.1 or later for MPLS passive monitoring
- JUNOS Release 6.0 or later for the Monitoring Services II PIC
- JUNOS Release 5.7 or later for the automatic insertion of autonomous system (AS) numbers and SNMP index values for interfaces into flow records
- JUNOS Release 5.4 or later for the Monitoring Services PIC
- M40e, M160, M320, MX-series, or T-series routing platform with an Internet Processor II ASIC or later
- Type 1 enhanced FPCs
- Two optical splitters
- A Tunnel Services PIC (required if you wish to send traffic to more than one analyzer)
- An input interface from the following list:
  - SONET/SDH PIC—OC3, OC12, or OC48
  - ATM2 IQ PIC—OC3 or OC12
  - 4-port Fast Ethernet PIC
  - Gigabit Ethernet PIC—4-port with small form-factor pluggable transceiver (SFP) or 10-port with SFP
  - 1-port 10-Gigabit Ethernet PIC with XENPAK
- Outgoing PICs to connect to the flow collector or packet analyzer
- Flow monitoring version 5 collector
- ES PIC and packet analyzers (optional)

Active Flow Monitoring System Requirements

To implement active flow monitoring, your system must meet these minimum requirements:

- JUNOS 9.3R2 or later for IPv6 support on flow monitoring version 9.
- JUNOS 9.3R2 or later for multiple flows for flow monitoring version 9.
- JUNOS Release 9.0 or later for version 9 flow aggregation to multiple flow servers.
- JUNOS Release 8.5 or later for active flow monitoring support on MultiServices 500 PICs.
- JUNOS Release 8.3 or later for flow monitoring version 9 support, MPLS support, and active flow monitoring support on MultiServices 100 and 400 PICs.
- JUNOS Release 8.2 or later for M120 router support and for flow monitoring version 5 and 8 support on MultiServices 100 and 400 PICs.
- JUNOS Release 8.1 or later for the flow-tap services application on Adaptive Services II PICs installed in M7i, M10i, M20, M40e, M320, and T-series routing platforms
- JUNOS Release 7.4 or later for port mirroring of IPv6 packets
- JUNOS Release 7.3 or later for active flow monitoring on Adaptive Services II PICs installed in TX Matrix platforms
- JUNOS Release 7.0 or later for active flow monitoring on Adaptive Services II PICs installed in T-series and M320 routing platforms
- JUNOS Release 7.0 or later for active flow monitoring on J-series Services Routers
- JUNOS Release 6.0 or later for the Adaptive Services PIC
- JUNOS Release 5.7 or later for the automatic insertion of AS numbers and SNMP index values for input and output interfaces into records, port mirroring to multiple ports, and discard accounting
- JUNOS Release 5.6 or later for the Monitoring Services PIC
- M5, M7i, M10, M10i, M20, M40e, M120, M160, M320, or T-series routing platform with an Internet Processor II ASIC or later; or a J-series Services Router
- Type 1 enhanced FPCs
- Two M-series or T-series PICs or J-series Physical Interface Modules (PIMs) of your choice: One to receive incoming traffic and one to forward outgoing traffic (the second PIC or PIM is not necessary for discard accounting)
- Export PICs to connect to the collector or packet analyzer
- Tunnel Services PIC (required for multiple port mirroring or mo- interface load balancing)
- Flow collector version 5, 8, or 9
- ES PIC and packet analyzers (optional)

Active Flow Monitoring PIC Specifications

For Monitoring Services PIC specifications, see Table 8 on page 288 and Table 9 on page 288. For Adaptive Services PIC specifications, see Table 10 on page 289. For MultiServices PIC specifications, see Table 11 on page 289 and Table 12 on page 289.

Table 8: Monitoring Services PIC Specifications

Specification	Description
Physical dimensions	Single-wide PIC that occupies one PIC slot
Connectors	DB-9 diagnostic serial console port
Status LED	One tricolor: ■ Off—The PIC is offline; it is safe to remove it from the chassis. ■ Green—The PIC is operating normally. ■ Amber—The PIC is initializing. ■ Red—The PIC has an error or failure; no further harm can be done by removing it from the chassis.
Application LED	One tricolor: ■ Off—The service is not running. ■ Green—The service is running under acceptable load. ■ Amber—The service is overloaded.

Table 9: Monitoring Services II PIC Specifications

Specification	Description
Physical dimensions	Single-wide PIC that occupies one PIC slot
Connectors	N/A
Status LED	One tricolor: ■ Off—The PIC is offline; it is safe to remove it from the chassis. ■ Green—The PIC is operating normally. ■ Amber—The PIC is initializing. ■ Red—The PIC has an error or failure; no further harm can be done by removing it from the chassis.
Application LED	One tricolor: ■ Off—The flow collector is not running. ■ Green—The flow collector is running under acceptable load. ■ Amber—The flow collector is overloaded.

Table 10: Adaptive Services PIC Specifications

Specification	Description
Physical dimensions	Single-wide PIC that occupies one PIC slot
Connectors	N/A
Status LED	One tricolor: ■ Off—The PIC is offline; it is safe to remove it from the chassis. ■ Green—The PIC is operating normally. ■ Amber—The PIC is initializing. ■ Red—The PIC has an error or failure; no further harm can be done by removing it from the chassis.
Application LED	One tricolor: ■ Off—The flow collector is not running. ■ Green—The flow collector is running under acceptable load. ■ Amber—The flow collector is overloaded.

Table 11: MultiServices 100 PIC

Specification	Description
Physical dimensions	Single-wide PIC that occupies one PIC slot
Connectors	N/A
Status LED	One tricolor: ■ Off—The PIC is offline; it is safe to remove it from the chassis. ■ Green—The PIC is operating normally. ■ Amber—The PIC is initializing. ■ Red—The PIC has an error or failure; no further harm can be done by removing it from the chassis.
Application LED	One tricolor: ■ Off—The service is not running. ■ Green—The service is running under acceptable load. ■ Amber—The service is overloaded.

Table 12: MultiServices 400 PIC

Specification	Description
Physical dimensions	Single-wide PIC that occupies one PIC slot
Connectors	N/A

Table 12: MultiServices 400 PIC *(continued)*

Specification	Description
Status LED	One tricolor: ■ Off—The PIC is offline; it is safe to remove it from the chassis. ■ Green—The PIC is operating normally. ■ Amber—The PIC is initializing. ■ Red—The PIC has an error or failure; no further harm can be done by removing it from the chassis.
Application LED	One tricolor: ■ Off—The service is not running. ■ Green—The service is running under acceptable load. ■ Amber—The service is overloaded.

Table 13: MultiServices 500 PIC

Specification	Description
Physical dimensions	Single-wide PIC that occupies one PIC slot
Connectors	N/A
Status LED	One tricolor: ■ Off—The PIC is offline; it is safe to remove it from the chassis. ■ Green—The PIC is operating normally. ■ Amber—The PIC is initializing. ■ Red—The PIC has an error or failure; no further harm can be done by removing it from the chassis.
Application LED	One tricolor: ■ Off—The service is not running. ■ Green—The service is running under acceptable load. ■ Amber—The service is overloaded.

Terms and Acronyms

A

active flow monitoring Technique to lawfully intercept and observe specified data network traffic on an active routing platform participating in the network.

Adaptive Services PIC Advanced PIC that handles active flow monitoring, Network Address Translation (NAT), stateful firewall, and intrusion detection functions. For more information on the Adaptive Services PIC, see the *JUNOS Services Interfaces Configuration Guide*.

C

cflowd Version 5 and version 8 flow monitoring process that captures flow information from network traffic and exports this data into summary tables. Once captured, flow data can be analyzed as needed. For more information about cflowd, see http://www.caida.org.

content destination A recipient of monitored packets sent by a DTCP or dynamic flow capture-enabled monitoring station.

control source A dynamic flow capture client that wants to monitor electronic data or voice transfer over the network. The control source sends filter requests to the dynamic flow capture-enabled monitoring station by using DTCP.

D

DTCP (Dynamic Tasking Control Protocol) Protocol used to specify filtering criteria in a dynamic flow capture environment.

dynamic flow capture Technique that allows DTCP-enabled control sources to send specified filtering criteria in real time to a monitoring station. The monitoring station passively monitors the specified traffic flows on demand and sends the captured packets to content destinations.

E

ES PIC PIC that handles encryption and security services (such as IP Security [IPSec]).

F

flow collector interface Converted Monitoring Services II PIC that processes multiple flow records into compressed ASCII data files and exports these files to an FTP server.

M

Monitoring Services II PIC Advanced PIC that handles passive flow monitoring functions.

Monitoring Services III PIC Advanced PIC that handles dynamic flow capture functions.

Monitoring Services PIC Original PIC that handles passive and active flow monitoring functions.

MultiServices 100 PIC Also referred to as MultiServices PIC Type 1. Advanced PIC that handles active flow capture functions.

MultiServices 400 PIC Also referred to as MultiServices PIC Type 2. Advanced PIC that handles active flow capture functions.

MultiServices 500 PIC Also referred to as MultiServices PIC Type 3. Advanced PIC that handles active flow capture functions.

P

passive flow monitoring Technique to lawfully intercept and observe specified data network traffic on a passive flow monitoring station not participating in the network.

Configuring Passive Flow Monitoring

Table 14 on page 292 shows which Juniper Networks PICs and routing platforms support passive flow monitoring. The PICs receive passively monitored network traffic from an input interface (SONET/SDH, ATM2 IQ, Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet), convert the received packets into flow records, and export them to a flow server for further analysis.

Table 14: Passive Flow Monitoring PIC Support

PIC Type	M40e	M160	T-series/ M320
Monitoring Services PIC	Yes	Yes	No
Monitoring Services II PIC	Yes	Yes	Yes
Monitoring Services III PIC	Yes	Yes	Yes
MultiServices 400 PIC (Type 2)	Yes	No	Yes

The key configuration hierarchy statement for passive flow monitoring is the monitoring statement found at the [edit forwarding-options] hierarchy level. At minimum, you must configure a VRF routing instance to direct the traffic to a monitoring services interface for flow processing.

However, there are several options you can use that add complexity to passive flow monitoring. For example, you can configure the routing platform to direct traffic into a routing instance and deliver the traffic into a monitoring group. You can also use port mirroring and filter-based forwarding to copy and redirect traffic. Optionally, you can configure the monitoring station to encrypt flow output before it is sent to a flow server for processing, to send flow records to a flow collector, or to process on-demand monitoring requests with dynamic flow capture.

The following sections explain the variety of passive flow monitoring topics:

Monitoring Traffic with a VRF Instance and a Monitoring Group

The first way you can implement passive flow monitoring is to direct traffic into a VRF routing instance and use a monitoring group to export this traffic to a flow server for analysis. Complete the following tasks:

Specifying a Firewall Filter to Select Traffic to Monitor

When you define a firewall filter, you select the initial traffic to be monitored. To configure a firewall filter, include the filter statement at the [edit firewall family inet] hierarchy level. All filtered traffic to be monitored must be accepted.

```
[edit]
firewall {
    family inet {
        filter input-monitoring-filter {
            term 1 {
                from {
                    destination-address {
                        10.7.0.0/16;
                    }
                }
                then {
                    count counter1;
                    accept;
                }
            }
            term 2 {
```

```
            from {
                destination-address {
                    10.6.0.0/16;
                }
            }
            then {
                count counter2;
                accept;
            }
        }
    }
}
```

Configuring Input Interfaces, Monitoring Services Interfaces, and Export Interfaces

After creating the input filter, you need to configure the interfaces where traffic will enter the routing platform. To enable passive flow monitoring for SONET/SDH input interfaces, include the passive-monitor-mode statement at the [edit interfaces so-*fpc/pic/port* unit *unit-number*] hierarchy level. This mode disables the routing platform from participating in the network as an active device. On SONET/SDH interfaces, passive monitor mode suppresses SONET keepalives.

For ATM2 IQ interfaces, passive monitor mode suppresses the sending and receiving of ATM Operations, Administration, and Maintenance (OAM) and Integrated Local Management Interface (ILMI) control messages. To enable passive flow monitoring for ATM2 IQ input interfaces, include the passive-monitor-mode statement at the [edit interfaces at-*fpc/pic/port*] hierarchy level. ATM passive monitoring supports the following interface encapsulation types: Cisco-compatible ATM Network Layer Protocol ID (NLPID) (atm-cisco-nlpid), ATM NLPID (atm-nlpid), ATM Point-to-Point Protocol (PPP) over ATM Adaptation Layer 5 (AAL5)/ logical link control (LLC) (atm-ppp-llc), ATM PPP over raw AAL5 (atm-ppp-vc-mux), ATM LLC/ subnetwork attachment point (SNAP) (atm-snap), and ATM virtual circuit (VC) multiplexing (atm-vc-mux).

Ethernet-based interfaces support both per-port passive monitoring and per-VLAN passive monitoring. For Fast Ethernet interfaces, include the passive-monitor-mode statement at the [edit interfaces fe-*fpc/pic/port*] hierarchy level. For Gigabit Ethernet interfaces, include the passive-monitor-mode statement at the [edit interfaces ge-*fpc/pic/port*] hierarchy level. On Ethernet-based interfaces, passive monitor mode disables the Routing Engine from receiving packets and prevents the routing table from transmitting packets. You can verify this by the presence of the No-receive and No-transmit interface flags in the output of the show interfaces (fe | ge)-*fpc/pic/port* command.

NOTE: The following restrictions apply to passive flow monitoring on Ethernet-based interfaces:

- No special encapsulation types are allowed, so you must configure Ethernet encapsulations only.
- When you configure the passive-monitor-mode statement, destination MAC address filters applied to incoming interfaces are disabled by default.
- The flow-control statement at the [edit interfaces ge-*fpc/pic/port* gigether-options] or [edit interfaces fe-*fpc/pic/port* fastether-options] hierarchy level does not work when passive flow monitoring is enabled.

In addition to passive monitor mode, apply the previously defined firewall filter to the interface with the filter statement at the [edit interfaces *interface-name-fpc/pic/port* unit *unit-number* family inet] hierarchy level:

```
[edit]
interfaces {
   so-0/0/0 {
      description “SONET/SDH input interface”;
      encapsulation ppp;
      unit 0 {
         passive-monitor-mode;
         family inet {
            filter {
               input  input-monitoring-filter;
            }
         }
      }
   }
   at-1/0/0 {
      description “ATM2 IQ input interface”;
      passive-monitor-mode;
      atm-options {
         pic-type atm2;
         vpi 0 {
            maximum-vcs 255;
         }
      }
      unit 0 {
         encapsulation atm-snap;
         vci 0.100;
         family inet {
            filter {
               input input-monitoring-filter;
            }
         }
      }
   }
   ge-2/0/0 {
      description “Gigabit Ethernet input interface”;
      passive-monitor-mode;
      unit 0 {
```

```
            family inet {
               filter {
                  input input-monitoring-filter;
               }
            }
         }
      }
   }
```

Configure the interfaces on the Monitoring Services PIC or Monitoring Services II PIC with the family inet statement at the [edit interfaces mo-*fpc/pic/port* unit *unit-number*] hierarchy level. The statement allows the interfaces to process IPv4 traffic received from the input interfaces.

When you use VRF instances, you need to configure two logical interfaces. The first (unit 0) is part of the inet.0 routing table and sources the flow packets. The second (unit 1) is configured as part of the VRF instance so the monitoring services interface can serve as a valid next hop for packets received in the instance.

You can also capture options packets and time-to-live (TTL) exceeded information when the monitoring services interface processes flow records. To configure, include the receive-options-packets and receive-ttl-exceeded statements at the [edit interfaces mo-*fpc/pic/port* unit *unit-number* family inet] hierarchy level:

```
[edit]
interfaces {
   mo-4/0/0 {
      unit 0 {
         family inet {
            receive-options-packets;
            receive-ttl-exceeded;
         }
      }
      unit 1 {
         family inet;
      }
   }
   mo-4/1/0 {
      unit 0 {
         family inet;
      }
      unit 1 {
         family inet;
      }
   }
   mo-4/2/0 {
      unit 0 {
         family inet;
      }
      unit 1 {
         family inet;
      }
   }
   mo-4/3/0 {
      unit 0 {
```

```
            family inet;
        }
        unit 1 {
            family inet;
        }
    }
}
```

You must also configure the export interface where flow packets exit the monitoring station and are sent to the flow server.

On output interfaces, you can apply a firewall filter that leads to a filter-based forwarding routing instance. This is useful if you want to port-mirror traffic to multiple Monitoring Services PICs or flow collection services interfaces. To configure, include the output statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet filter] hierarchy level. For more information, see "Using Filter-Based Forwarding to Export Monitored Traffic to Multiple Destinations" on page 304.

```
[edit]
interfaces
fe-3/0/0 {
    description "export interface to flow server";
    unit 0 {
        family inet;
        address ip-address;
        filter {
            output output-filter-name;
        }
    }
}
```

Establishing a VRF Instance for the Monitored Traffic

After the firewall filter and interfaces are ready, create a VPN routing and forwarding (VRF) instance. The filtered traffic enters the VRF instance and is shared only between the input interfaces and the monitoring services output interfaces. In this case, a group of four monitoring services interfaces is used as the next hop.

```
[edit]
routing-instances {
    monitoring-vrf {
        instance-type vrf;
        interface so-0/0/0.0;
        interface so-0/1/0.0;
        interface mo-4/0/0.1;
        interface mo-4/1/0.1;
        interface mo-4/2/0.1;
        route-distinguisher 69:1;
        vrf-import monitoring-vrf-import;
        vrf-export monitoring-vrf-export;
        routing-options {
            static {
                route 0.0.0.0/0 next-hop [mo-4/0/0.1 mo-4/1/0.1 mo-4/2/0.1];
            }
```

```
        }
    }
}
```

Configuring a Monitoring Group to Send Traffic to the Flow Server

You collect flow records by specifying output interfaces in a monitoring group. In general, the monitoring services interfaces are the output interfaces. The logical unit number on the output interfaces when used in conjunction with a VRF instance must be 1. To configure, include the output statement at the [edit forwarding-options monitoring *group-name* family inet] hierarchy level.

NOTE: Because routing instances determine the input interface, the input statement at the [edit forwarding-options monitoring *group-name* family inet] hierarchy level has been removed in JUNOS Release 6.0 and later. If you have a configuration that contains this old statement, we recommend that you update your configuration and remove the statement.

As part of the mo-*fpc/pic/port* statement at the [edit forwarding-options monitoring *group-name* family inet output interface] hierarchy level, you must specify a source address for transmission of flow information. You can use the routing platform ID IP address, the IP address of the input interface, or any local IP address of your choice as the source address. If you provide a different source-address statement for each monitoring services output interface, you can track which interface processes a particular flow record.

All other statements at this level (engine-id, engine-type, input-interface-index, and output-interface-index) are dynamically generated, but can be configured manually. To reset outgoing interface or incoming interface indexes that were once configured manually, configure the input-interface-index or outgoing-interface-index statements with a value of 0 at the [edit forwarding-options monitoring *group-name* family inet output interface *interface-name*] hierarchy level.

To specify the flow server IP address and port number, include the flow-server *ip-address* port *port-number*statement at the [edit forwarding-options monitoring *group-name* family inet output] hierarchy level. You can specify up to eight flow servers in a monitoring group and the IP address for each server must be unique. flow records are exported and load-balanced between all active flow servers.

Once you configure the VRF and monitoring group statements, traffic enters the input interfaces, passes to the monitoring services interfaces for processing, and is discarded. The resulting flow description packets exit the monitoring station through the export interface. If you want traffic to travel to destinations other than the monitoring services interfaces, or need to establish additional analysis, see the section “Copying and Redirecting Traffic with Port Mirroring and Filter-Based Forwarding” on page 301.

NOTE: You must complete interface configuration on the Monitoring Services or Monitoring Services II PIC before an interface can be added into a monitoring group. For more information, see "Configuring Input Interfaces, Monitoring Services Interfaces, and Export Interfaces" on page 294.

```
[edit]
forwarding-options {
   monitoring group1 {
      family inet {
         output {
            export-format cflowd-version-5;
            flow-active-timeout 60;
            flow-inactive-timeout 30;
            flow-server 192.168.245.1 port 2055;
            flow-server 192.168.245.2 port 2055;
            interface mo-4/0/0.1 {
               engine-id 1;
               engine-type 1;
               input-interface-index 44;
               output-interface-index 54;
               source-address 192.168.245.1;
            }
            interface mo-4/1/0.1 {
               engine-id 2;
               engine-type 1;
               input-interface-index 45;
               output-interface-index 55;
               source-address 192.168.245.1;
            }
            interface mo-4/2/0.1 {
               engine-id 3;
               engine-type 1;
               input-interface-index 46;
               output-interface-index 56;
               source-address 192.168.245.1;
            }
         }
      }
   }
}
```

Configuring Policy Options

When you use a group of next hops in your monitoring group, you can load-balance traffic and distribute it to the export interfaces if you configure policy options. To configure, include the load-balance per-packet statement at the [edit policy-options policy-statement *policy-name* then] hierarchy level. You can also reject import and export of VRF routes by including the reject statement at the [edit policy-options policy-statement *policy-name* then] hierarchy level.

```
[edit]
routing-options {
```

```
    forwarding-table {
        export pplb;
    }
}
policy-options {
    policy-statement monitoring-vrf-import {
        then {
            reject;
        }
    }
    policy-statement monitoring-vrf-export {
        then {
            reject;
        }
    }
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
    }
}
```

Option: Stripping MPLS Labels on ATM, Ethernet-Based, and SONET/SDH Interfaces

Because flow monitoring can be performed only on IPv4 packets, any packets containing MPLS labels must have the labels removed before monitoring can occur. To remove MPLS labels from packets as they enter an ATM2 IQ, Ethernet-based, or SONET/SDH interface, include the pop-all-labels statement at the [edit interfaces *interface-name-fpc/pic/port* (atm | fastether | gigether | sonet)-options mpls] hierarchy level. If you use static MPLS labels, we recommend you assign label values from 10000 through 99999 to avoid using the label ranges reserved by the JUNOS software.

To remove a specified number of labels from selected packets with MPLS labels, include the required-depth statement at the [edit interfaces *interface-name-fpc/pic/port* (atm | fastether | gigether | sonet)-options mpls pop-all-labels] hierarchy level. A required-depth value of 1 removes labels from all packets containing only one MPLS label, a value of 2 removes labels from all packets containing only two MPLS labels, and a value of [1 2] removes labels from all packets containing either one or two MPLS labels. The required-depth value of [1 2] is the default setting. When you configure the required-depth statement, you must configure the same value for all ports on the same PIC.

The labels are removed and discarded as soon as they arrive at the interface. As a result, no MPLS filters can be applied to the stripped labels, no statistics are generated for the labels, and you cannot apply an IP filter to the incoming packets. No Tunnel Services PIC is required to perform MPLS label stripping.

```
[edit]
interfaces {
    at-/fpc/pic/port  {
        atm-options {
            mpls {
```

```
            pop-all-labels {
               required-depth 1;
            }
         }
      }
   }
   (fe | ge)-fpc/pic/port {
      (fastether | gigether)-options {
         mpls {
            pop-all-labels {
               required-depth [1 2];
            }
         }
      }
   }
   so-fpc/pic/port {
      sonet-options {
         mpls {
            pop-all-labels {
               required-depth 2;
            }
         }
      }
   }
}
```

Copying and Redirecting Traffic with Port Mirroring and Filter-Based Forwarding

This section discusses additional techniques you can use with the passive flow monitoring application:

- In addition to flow analysis, you can analyze a copy of the original traffic with a single packet analyzer. To implement this technique, divert traffic with a filter-based forwarding routing instance and send the monitored traffic through a physical interface to the packet analyzer.
- You can cluster the traffic into different groups and redirect this traffic to multiple packet analyzers. For example, you can break traffic flows into TCP groups and UDP groups and send these groups of packets to different analyzers. To accomplish this, you use port mirroring and send a copy of the original traffic to a Tunnel PIC. Then you can apply a firewall filter, split the traffic into your desired groups, and send these groups toward different exit interfaces leading to the packet analyzers. This technique provides maximum flexibility for traffic analysis.
- For secure transmission of the copied or grouped traffic, you can encrypt the diverted traffic with an ES PIC and send this traffic to a packet analyzer over an IP Security (IPSec) tunnel.

To implement the filter-based forwarding enhancement methods, see the following sections:

- Specifying Port Mirroring Input and Output on page 302
- Creating a Firewall Filter to Split the Port-Mirrored Traffic into Different Instances on page 303

Specifying Port Mirroring Input and Output

This step works in conjunction with the action specified by the port-mirror statement configured at the [edit firewall family (inet | inet6) filter *filter-name* term *term-name* then] hierarchy level. At this point, you select input and output statements to determine where the copies of the IPv4 or IPv6 packets are sent. To configure, include the input and output statements at the [edit forwarding-options port-mirroring family *family-name*] hierarchy level. The traffic to be monitored is copied, port-mirrored, and sent to the packet analyzer for analysis. On M-series routers, you can port-mirror either IPv4 or IPv6 packets at one time. On M320 and T-series routing platforms, you can port-mirror both IPv4 and IPv6 packets simultaneously.

The port-mirrored copy of the traffic can travel only to a single next hop. As a result, only one type of analysis can be performed if the packets are sent to a packet analyzer through a physical next hop. If more than one type of analysis is desired, a tunnel interface must be used as the next hop for port mirroring. When the mirrored copy of the traffic arrives at the virtual tunnel interface, it can be filtered, split into groups, and redirected to multiple exit interfaces and packet analyzers.

For your input requirements, include the rate and run-length statements at the [edit forwarding-options port-mirroring family *family-name* input] hierarchy level. For your output requirements, specify the target interface with the interface statement at the [edit forwarding-options port-mirroring family *family-name* output] hierarchy level.

By default, a filter cannot be applied to an interface where port-mirrored traffic is received. To allow the tunnel services interface to be used as a filtered next hop, include the no-filter-check statement at the [edit forwarding-options port-mirroring family *family-name* output] hierarchy level.

```
[edit]
forwarding-options {
    port-mirroring {
        family (inet | inet6) {
            input {
                rate 1;
                run-length 5;
            }
            output {
                interface vt-0/2/0.0;
                no-filter-check;
            }
        }
    }
```

```
}
```

NOTE: Before JUNOS Release 7.4, you could configure the input and output statements at the [edit forwarding-options port-mirroring] hierarchy level. However, this older syntax has been revised to extend port-mirroring support to IPv6 packets. If you have a configuration that contains the older syntax, we recommend that you update your configuration to the new syntax listed above.

Creating a Firewall Filter to Split the Port-Mirrored Traffic into Different Instances

If you need to split the copy of the monitored traffic into separate groups and send these filtered packets to different analyzers, devise a firewall filter that selects some traffic for sampling and some traffic for discarding. In this case, UDP traffic is sent into one routing instance, TCP traffic is diverted into a second routing instance, and all other traffic is discarded. In a later step, you will define the filter-based forwarding routing instances specified in the then statements shown in this filter.

```
[edit]
firewall {
    family inet {
        filter tunnel-interface-filter {
            term tcp {
                from {
                    protocol tcp;
                }
                then {
                    count tcp;
                    routing-instance tcp-routing-table;
                }
            }
            term udp {
                from {
                    protocol udp;
                }
                then {
                    count udp;
                    routing-instance udp-routing-table;
                }
            }
            term rest {
                then {
                    count rest;
                    discard;
                }
            }
        }
    }
}
```

Applying the Firewall Filter to a Tunnel PIC Interface

Once the firewall filter is defined, apply it as an input filter on a tunnel interface. This is required if the firewall filter defines two or more types of traffic or export interfaces. However, if the firewall filter only specifies one type of traffic and one export interface, you can apply the filter directly to the export interface.

```
[edit]
interfaces {
    vt-0/2/0 {
        unit 0 {
            family inet {
                filter {
                    input tunnel-interface-filter;
                }
            }
        }
    }
}
```

Using Filter-Based Forwarding to Export Monitored Traffic to Multiple Destinations

The firewall filter called tunnel-interface-filter that you made earlier sends UDP traffic into one filter-based forwarding routing instance called udp-routing-table, sends TCP traffic into a second filter-based forwarding routing instance called tcp-routing-table, and discards all other packets. Here you will configure the filter-based forwarding instances.

Configure an export interface for each of your routing instances by including a static next hop. To configure, include the route statement at the [edit routing-instances *instance-name* routing-options static] hierarchy level and specify a next-hop address or interface.

```
[edit]
routing-instances {
    tcp-routing-table {
        instance-type forwarding;
        routing-options {
            static {
                route 0.0.0.0/0 next-hop es-3/1/0.0;
            }
        }
    }
    udp-routing-table {
        instance-type forwarding;
        routing-options {
            static {
                route 0.0.0.0/0 next-hop 10.9.1.2;
            }
        }
    }
```

```
}
```

Configuring a Routing Table Group to Add Interface Routes into the Forwarding Instance

Next, import the interface routes into the forwarding instance. This step is necessary because the next hops specified in the forwarding instances must be installed in the forwarding instances themselves. To configure, include the import-rib statement at the [edit routing-options rib-groups *group-name*] hierarchy level. The export statement at the [edit routing-options forwarding-table] hierarchy level and the pplb policy enables load balancing.

```
[edit]
routing-options {
   interface-routes {
      rib-group inet bc-vrf;
   }
   rib-groups {
      bc-vrf {
         import-rib [inet.0 tcp-routing-table.inet.0 udp-routing-table.inet.0];
      }
   }
   forwarding-table {
      export pplb;
   }
}
policy-options {
   policy-statement pplb {
      then {
         load-balance per-packet;
      }
   }
}
```

Option: Using an ES PIC to Send Traffic to a Packet Analyzer

You can send some or all of the traffic securely to the packet analyzer using IPSec and an ES PIC. In this case, the TCP traffic is encrypted, sent over an IPSec tunnel, and received by the packet analyzer. For more information on configuring IPSec on the ES PIC, see "IPSec" on page 397 or the *JUNOS System Basics Configuration Guide*.

```
[edit]
interfaces {
   es-3/1/0 {
      unit 0 {
         tunnel {
            source 10.8.8.1;
            destination 10.8.8.2;
         }
         family inet {
            ipsec-sa sa-esp;
            address 3.3.3.1/32 {
               destination 3.3.3.2;
```

```
            }
          }
        }
      }
      fe-3/2/1 {
        unit 0 {
          family inet {
            address 10.8.8.1/30;
          }
        }
      }
    }
    security {
      ipsec {
        proposal esp-sha1-3des {
          protocol esp;
          authentication-algorithm hmac-sha1-96;
          encryption-algorithm 3des-cbc;
          lifetime-seconds 180;
        }
        policy esp-group2 {
          perfect-forward-secrecy {
            keys group2;
          }
          proposals esp-sha1-3des;
        }
        security-association sa-esp {
          mode tunnel;
          dynamic {
            ipsec-policy esp-group2;
          }
        }
      }
      ike {
        proposal ike-esp {
          authentication-method pre-shared-keys;
          dh-group group2;
          authentication-algorithm sha1;
          encryption-algorithm 3des-cbc;
          lifetime-seconds 180;
        }
        policy 10.8.8.2 {
          mode aggressive;
          proposals ike-esp;
          pre-shared-key ascii-text "$9$qmQnuORrlMBIds2oiH0BIESe";
        }
      }
    }
```

Option: Applying a Firewall Filter to an Output Interface

On output interfaces, you can apply a firewall filter that leads to a filter-based forwarding routing instance. This is useful if you want to port-mirror traffic to multiple Monitoring Services PICs or flow collection services interfaces. To configure, include

the output statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet filter] hierarchy level.

```
[edit]
interfaces
fe-3/1/0 {
   description “export interface to flow collection services interfaces”;
   unit 0 {
      family inet;
      address ip-address;
      filter {
         output output-filter-name;
      }
   }
}
```

Using a Flow Collector Interface to Process and Export Multiple Flow Records

Basic passive monitoring can sometimes create a large number of flow records. However, you can manage multiple flow records with a flow collector interface. You can create a flow collector interface from a Monitoring Services II PIC. The flow collector interface combines multiple flow records received from a monitoring services interface into a compressed ASCII data file and exports the file to an FTP server.

To convert a Monitoring Services II PIC into a flow collector interface, include the flow-collector statement at the [edit chassis fpc *fpc-slot* pic *pic-slot* monitoring-services application] hierarchy level. To restore the monitoring functions of a Monitoring Services II PIC, include the monitor statement at the [edit chassis fpc *fpc-slot* pic *pic-slot* monitoring-services application] hierarchy level.

After you commit the configuration to convert the PIC between the monitor and flow-collector service types, you must take the PIC offline and then bring the PIC back online. Rebooting the routing platform does not enable the new service type. You can use the Monitoring Services II PIC for either flow collection or monitoring, but not both types of service simultaneously.

A flow collector interface, designated by the cp-*fpc/pic/port* interface name, requires three logical interfaces for correct operation. Units 0 and 1 are used respectively as export channels 0 and 1 to send the compressed ASCII data files to an FTP server. You must include a class-of-service (CoS) configuration for these two export channels to provide adequate bandwidth for file transmission. Unit 2 is used as a flow receive channel to receive flow records from a monitoring services interface.

NOTE: Unlike conventional interfaces, IP addresses for flow collector logical interfaces set up a point-to-point connection between the Routing Engine and the flow collector. The address statement at the [edit interfaces cp-*fpc/pic/port* unit *unit-number* family inet] hierarchy level corresponds to the IP address of the Routing Engine. Likewise, the destination statement at the [edit interfaces cp-*fpc/pic/port* unit *unit-number* family inet address *ip-address*] hierarchy level corresponds to the IP address of the flow collector interface. As a result, you must configure the destination statement for Units 0 and 1 (export channels 0 and 1) with *local* addresses that can reach the FTP server. Similarly, configure the destination statement for Unit 2 (flow receive channel) with a *local* IP address so it can reach the monitoring services interface that sends flow records.

To activate flow collector services after the Monitoring Services II PIC is converted into a flow collector, include the flow-collector statement at the [edit services] hierarchy level. You also need to configure several additional components:

- Destination of the FTP server—Determines where the compressed ASCII data files are sent after the flow records are collected and processed. To specify the destination FTP server, include the destinations statement at the [edit services flow-collector] hierarchy level. You can specify up to two FTP server destinations and include the password for each configured server. If two FTP servers are configured, the first server in the configuration is the primary server and the second is a backup server.
- File specifications—Preset data file formats, name formats, and transfer characteristics. Files are sent by FTP to the destination FTP server when the timer expires or when a preset number of records are received, whichever comes first. To set the data file format, include the data-format statement at the [edit services flow-collector file-specification *file-name*] hierarchy level. The default data format is flow-compressed. To set the export timer and file size thresholds, include the transfer statement at the [edit services flow-collector file-specification *file-name*] hierarchy level and specify values for the timeout and record-level options. The default values are 600 seconds for timeout and 500,000 records for record-level.

  To set the filename format, include the name-format statement at the [edit services flow-collector file-specification *file-name*] hierarchy level. Common name format macros that you can use in your configuration are included in Table 15 on page 308.

Table 15: Name Format Macros

Field	Expansion
{*am_pm*}	AM or PM
{*date*}	Expands to the current date, using the {month}, {day}, and {year} macros.
{*day*}	01 to 31
{*day_abbr*}	Sun through Sat
{*day_full*}	Sunday through Saturday

Table 15: Name Format Macros *(continued)*

Field	Expansion
{*generation_number*}	Expands to a unique, sequential number for each new file created.
{*hour_12*}	01 to 12
{*hour_24*}	00 to 23
{*ifalias*}	Expands to a description string for the logical interface.
{*minute*}	00 to 59
{*month*}	01 to 12
{*month_abbr*}	Jan through Dec
{*month_full*}	January through December
{*num_zone*}	-2359 to +2359
{*second*}	00 to 60
{*time*}	Expands to the time the file is created, using the {hour_24}, {minute}, and {second} macros.
{*time_zone*}	Time zone code name of the locale (gmt, pst, and so on.)
{*year*}	1970, 2008, and so on.
{*year_abbr*}	00 to 99

- Input interface-to-flow collector interface mappings—Match an input interface with a flow collector interface and applies the preset file specifications to the input interface. To configure the default flow collector and file specifications for all input interfaces, include the file-specification and collector statements at the [edit services flow-collector interface-map] hierarchy level. To override the default settings and apply flow collector and file specifications to a specific input interface, include the file-specification and collector statements at the [edit services flow-collector interface-map *interface-name*] hierarchy level.
- Transfer log settings—Allow you to configure the destination FTP server where log files containing the transfer activity history for a flow collector interface are to be archived, the name for the log file, and the amount of time the routing platform waits before sending the log file to the FTP server. To configure, include the archive-sites, filename-prefix, and maximum-age statements at the [edit services flow-collector transfer-log-archive] hierarchy level. The default value for the maximum-age statement is 120 minutes, with a range of 1 to 360 minutes. Also, you can configure up to five FTP archive site servers to receive log files.
- Miscellaneous settings—Allow you to configure values for the IP address of the analyzer, an identifier for the analyzer, the maximum number of times the flow collector interface attempts to send transfer log files to the FTP server, and the amount of time the flow collector interface waits between retry attempts. To

configure, include the analyzer-address, analyzer-id, retry, and retry-delay statements at the [edit services flow-collector] hierarchy level. The range for the retry statement is 0 through 10 retry attempts. The default for the retry-delay statement is 30 seconds and the range is 0 through 60 seconds.

To specify a flow collector interface as the destination for flow records coming from a Monitoring Services or Monitoring Services II PIC, include the collector-pic statement at the [edit forwarding-options monitoring *group-name* family inet output flow-export-destination] hierarchy level. You can select either the flow collector interface or a flow server as the destination for flow records, but you cannot select both destination types simultaneously.

There is also a Juniper Networks enterprise Management Information Base (MIB) for the flow collector interface. The Flow Collector Services MIB allows you to use SNMP to monitor the flow collector interface. The MIB provides statistics on files, records, memory, FTP, and error states of a flow collector interface. It also provides SNMP traps for unavailable destinations, unsuccessful file transfers, flow overloading, and memory overloading. For more information, see the *JUNOS Network Management Configuration Guide* or view the enterprise-specific Juniper Networks MIBs at http://www.juniper.net/techpubs/software/junos/mibs.html.

In summary, to implement the flow collector service, include statements at the [edit chassis], [edit interfaces], [edit forwarding-options], and [edit services] hierarchy levels. The excerpt on the following pages shows the flow collector service configuration hierarchy. For a full configuration example, see “Example: Flow Collector Interface Configuration” on page 333.

```
[edit]
chassis {
    fpc fpc-slot {
        pic pic-slot {
            monitoring-services {
                application flow-collector;
            }
        }
    }
}
interfaces {
    cp-fpc/pic/port {
        description ”flow_collector_interface”;
        unit 0 {
            family inet {
                address ip-address {
                    destination  ip-address;
                }
            }
        }
        unit 1 {
            family inet {
                address ip-address {
                    destination ip-address;
                }
            }
        }
```

```
        unit 2 {
          family inet {
            address ip-address {
              destination ip-address;
            }
          }
        }
      }
      interface-fpc/pic/port {
        description “export_interface”;
        unit 0 {
          family inet {
            address ip-address;
          }
        }
      }
      mo-fpc/pic/port  {
        description “monitoring_services_interface”;
        unit 0 {
          family inet;
        }
      }
      SONET/SDH, ATM2 IQ, or Ethernet-based-interface-fpc/pic/port {
        description “ input_interface”;
        encapsulation encapsulation-type;
        passive-monitor-mode; # Apply to the logical interface for SONET/SDH
      }
    }
    forwarding-options {
      monitoring group1 {
        family inet {
          output {
            export-format cflowd-version-5;
            flow-active-timeout value;
            flow-inactive-timeout value;
            flow-export-destination collector-pic;
            interface mo-fpc/pic/port {
              source-address ip-address;
            }
          }
        }
      }
    }
    services {
      flow-collector {
        analyzer-address ip-address;
        analyzer-id name;
        retry value;
        retry-delay seconds;
        destinations {
          "ftp://username@ftp-server-address-1//directory/" {
            password "encrypted-password";
          }
          "ftp://username@ftp-server-address-2//directory/" {
            password "encrypted-password";
          }
```

```
        }
        file-specification {
            file-specification-name {
            }
            data-format flow-compressed;
            transfer timeout value record-level size;
        }
    }
    interface-map {
        file-specification file-specification-name;
        collector cp-fpc/pic/port;
        interface-name {
            file-specification file-specification-name;
            collector cp-fpc/pic/port;
        }
    }
    transfer-log-archive {
        filename-prefix filename;
        maximum-age timeout-value;
        archive-sites {
            "ftp://username@ip-address//directory/" {
                password "encrypted-password";
            }
        }
    }
}
```

Using a Dynamic Flow Capture Interface to Monitor Traffic On Demand

Dynamic flow capture enables you to capture packet flows based on filtering criteria that you specify in real time. Unlike traditional flow monitoring that requires filtering criteria to be established before operation, dynamic flow capture uses an on demand control protocol that allows you to modify the filtering criteria as network conditions change.

The dynamic flow capture architecture consists of one or more *control sources* that send Dynamic Tasking Control Protocol (DTCP) requests to a *monitoring station*. The requests contain filtering criteria that specify which incoming traffic should be monitored, and the monitoring station forwards any packets that match the filter criteria to a set of one or more *content destinations*.

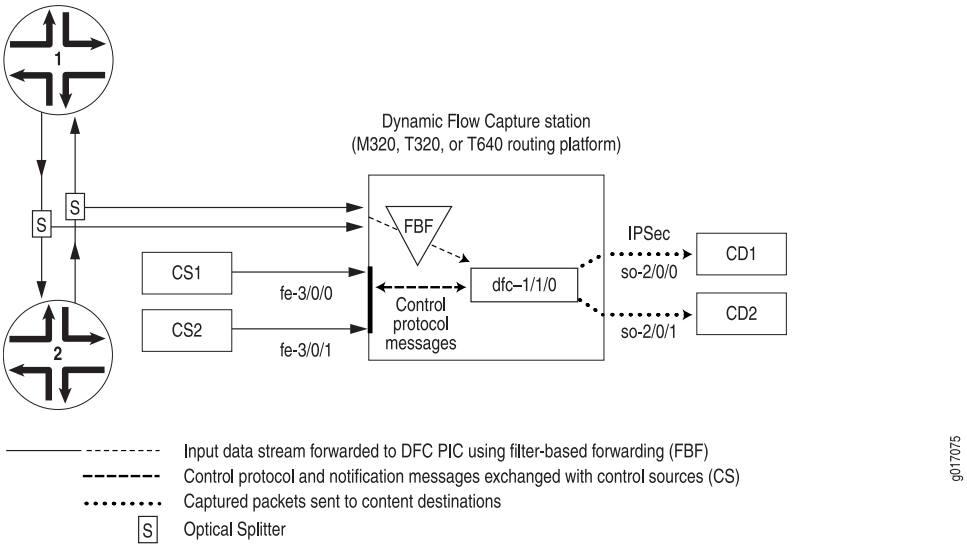
- Control source—A client that wants to monitor electronic data or voice transfer over the network. The control source sends filter requests to the Juniper Networks routing platform using DTCP. The control source is identified by a unique identifier and an optional list of IP addresses.
- Monitoring station—A Juniper Networks T-series or M320 routing platform configured with one or more Monitoring Services III PICs which support dynamic flow capture processing. The monitoring station processes the requests from the control sources, creates the filters, monitors incoming data flows, and sends the matched packets to the appropriate content destinations.
- Content destination—Recipient of the matched packets from the monitoring station. Typically the matched packets are sent using an IPSec tunnel from the

monitoring station to another router connected to the content destination. The content destination and the control source can be located on the same host.

NOTE: The DFC PIC forwards the entire packet content to the content destination, rather than just a content record.

Figure 24 on page 313 shows a sample topology that contains control sources, a monitoring station, and content destinations.

Figure 24: Dynamic Flow Capture Topology



To configure dynamic flow capture, perform the following tasks:

- Configuring the Capture Group on page 313
- Configuring the Content Destination on page 314
- Configuring the Control Source on page 314
- Configuring the Dynamic Flow Capture Interface on page 315
- Option: Configuring Thresholds on page 316
- Option: Configuring System Logging on page 317
- Option: Monitoring Dynamic Flow Capture by Using SNMP on page 317

Configuring the Capture Group

A dynamic flow capture capture group defines a profile of dynamic flow capture configuration information. The static configuration includes information about control sources, content destinations, and notification destinations. Dynamic configuration is added through interaction with control sources using a control protocol.

To configure a capture group, include the capture-group statement at the [edit services dynamic-flow-capture] hierarchy level:

```
[edit services dynamic-flow-capture]
capture-group client-name {
    content-destination identifier {
        address address;
        ttl hops;
    }
    control-source identifier {
        allowed-destinations [ destination ];
        no-syslog;
        notification-targets [ address address port  port-number ];
        service-port port-number;
        shared-key value;
        source-addresses [ address ];
    }
    input-packet-rate-threshold rate;
    interfaces interface-name;
    pic-memory-threshold percentage percentage;
}
```

To specify the capture-group, assign it a unique *client-name* that associates the information with the requesting control sources.

Configuring the Content Destination

You must specify a destination for the packets that match dynamic flow capture filter criteria. To configure, include the content-destination statement at the [edit services dynamic-flow-capture capture-group *client-name*] hierarchy level:

```
[edit services dynamic-flow-capture capture-group client-name]
content-destination  identifier {
    address address;
    ttl hops;
}
```

Assign the content-destination a unique *identifier*. In addition, you must specify its IP address, and you can optionally set a time-to-live (TTL) value for the IP-IP header:

- address—The dynamic flow capture interface appends an IP header with this destination address on the matched packet (with its own IP header and contents intact) and sends it out to the content destination.
- ttl—By default, the TTL value is 255. Its range is from 0 through 255.

Configuring the Control Source

You configure information about the control source, including allowed source addresses and destinations and authentication key values. To configure the control source information, include the control-source statement at the [edit services dynamic-flow-capture] hierarchy level:

```
[edit services dynamic-flow-capture capture-group client-name]
control-source identifier {
    allowed-destinations [ destination-identifier ];
    no-syslog;
    notification-targets [ address address port port-number ];
    service-port port-number;
    shared-key value;
    source-addresses [ address ];
}
```

Assign the control-source statement with a unique *identifier*. You can also include values for the following statements:

- allowed-destinations—One or more content destination identifiers to which this control source can request matched data to be sent in its control protocol requests. If you do not specify any content destinations, all available destinations are allowed.
- notification-targets—One or more destinations to which the dynamic flow capture interface can log information about control protocol-related events and other events such as PIC startup messages. You configure each notification-target entry with an IP address value and a User Datagram Protocol (UDP) port number.
- service-port—UDP port number to which the control protocol requests are directed. Control protocol requests that are not directed to this port are discarded by dynamic flow capture interfaces.
- shared-key—A 20-byte authentication key value shared between the control source and the dynamic flow capture monitoring station.
- source-addresses—One or more allowed IP addresses from which the control source can send control protocol requests to the dynamic flow capture monitoring station. These are /32 addresses.

Configuring the Dynamic Flow Capture Interface

You specify the interface that interacts with the control sources configured in the same dynamic flow capture group. A Monitoring Services III PIC can belong to only one capture group, and you can configure only one PIC for each group.

To configure a dynamic flow capture interface, include the interfaces statement at the [edit services dynamic-flow-capture] hierarchy level:

```
[edit services dynamic-flow-capture capture-group client-name]
interfaces interface-name;
```

You specify dynamic flow capture interfaces using the dfc- identifier at the [edit interfaces] hierarchy level. Three logical units are required on each dynamic flow capture interface, numbered 0, 1, and 2. You cannot configure any other logical interfaces.

- unit 0 processes control protocol requests and responses.
- unit 1 receives monitored data.
- unit 2 transmits the matched packets to the destination address.

The following example shows the configuration necessary to set up a dynamic flow capture interface:

```
[edit interfaces dfc-0/0/0]
unit 0 {
   family inet {
      address 10.1.0.0/32 { # Address of the Routing Engine for the DFC PIC.
         destination 10.36.100.1; # Address of DFC PIC; used by the
            # control source to correspond with the monitoring station.
      }
   }
}
unit 1 { # Receives data packets on this logical interface.
   family inet;
}
unit 2 { # Sends copies of matched packets from this logical interface.
   family inet;
}
```

In addition, you must configure the dynamic flow capture application to run on the DFC PIC in the correct chassis location. The following example shows this configuration at the [edit chassis] hierarchy level:

```
[edit chassis]
fpc 0 {
   pic 0 {
      monitoring-services application dynamic-flow-capture;
   }
}
```

For more information on configuring chassis properties, see the *JUNOS System Basics Configuration Guide*.

Option: Configuring Thresholds

You can optionally specify threshold values for situations in which warning messages will be recorded in the system log:

- Input packet rate to the dynamic flow capture interfaces
- Memory usage on the dynamic flow capture interfaces

To configure, include the input-packet-rate-threshold or pic memory-threshold statements at the [edit services dynamic-flow-capture capture-group *client-name*] hierarchy level:

```
[edit services dynamic-flow-capture capture-group  client-name]
input-packet-rate-threshold rate;
pic-memory-threshold percentage percentage;
```

If these statements are not configured, no threshold messages are logged. The threshold settings are configured for the capture group as a whole.

Option: Configuring System Logging

By default, control protocol activity is logged as a separate system log facility, dfc. To modify the filename or level at which control protocol activity is recorded, include the following statements at the [edit syslog] hierarchy level:

```
[edit syslog]
file dfc.log {
    dfc any;
}
```

To cancel logging, include the no-syslog statement at the [edit services dynamic-flow-capture *capture-group client-name* control-source *identifier*] hierarchy level:

```
[edit services dynamic-flow-capture capture-group client-name control-source identifier]
no-syslog;
```

Option: Monitoring Dynamic Flow Capture by Using SNMP

In JUNOS Release 7.5 and later, the Dynamic Flow Capture MIB provides a way to monitor dynamic flow capture information by using Simple Network Management Protocol (SNMP). The MIB provides the same information that you can view with the show services dynamic-flow-capture content-destination, show services dynamic-flow-capture control-source, and show services dynamic-flow-capture statistics commands. For more information, see the *JUNOS Network Management Configuration Guide*.

Hardware and Software Considerations

There are several hardware and software considerations when you implement passive flow monitoring. When defining the hardware requirements of the monitoring station, keep in mind the following:

- The input interfaces on the monitoring station must be SONET/SDH interfaces (OC3, OC12, or OC48), ATM2 IQ interfaces (OC3 or OC12), 4-port Fast Ethernet interfaces, Gigabit Ethernet interfaces with SFP (4-port or 10-port), or 1-port 10-Gigabit Ethernet interfaces with XENPAK.
- To monitor the flows in both directions for a single interface, the monitoring station must have two SONET/SDH, ATM2 IQ, or Ethernet-based receive ports, one for each direction of flow. In Figure 23 on page 284, the monitoring station needs one port to monitor the traffic flowing from Router 1 to Router 2, and a second port to monitor the traffic flowing from Router 2 to Router 1.
- The Monitoring Services PICs must be installed in a Type 1 enhanced FPC slot.
- Type 1 and Type 2 Tunnel Services PICs are supported.
- Use an ES PIC to encrypt the flow export.

When defining a traffic monitoring strategy, keep in mind the following:

- The monitoring station collects only IPv4 packets. All other packet formats are discarded and not counted.
- You can set the amount of time a data flow can be inactive before the monitoring station terminates the flow and exports the flow data. To set the timer, include the flow-inactive-timeout statement at the [edit forwarding-options monitoring *group-name* family inet output] hierarchy level. The timer value can be from 15 seconds through 1800 seconds, with a default value of 60 seconds.

You can also configure the monitoring station to collect periodic flow reports for flows that last longer than the configured active timeout. To set this activity timer, include the flow-active-timeout statement at the [edit forwarding-options monitoring *group-name* family inet output] hierarchy level. The timer value can be from 60 seconds through 1800 seconds, with a default value of 180 seconds.

- Multiple expired flows are exported together, if possible. A UDP packet is sent when one of the following conditions is met:
  - When 30 flows are contained in the current packet, the flows are exported.
  - If there are fewer than 30 flows but the export timer expires, the flows are exported one second after the timer expires.
- TCP and UDP flows are considered differently:
  - TCP flows watch for a segment containing the FIN bit and a subsequent acknowledgement (ACK) to detect the end of a flow. Alternately, a TCP reset (RST) can also indicate the end of a flow. When these TCP combinations are detected, the flow expires. The FIN+ACK and RST cases cover most TCP stream closures. For all other flows, an inactive timeout is needed.
  - All non-TCP flows, such as UDP, depend on timeout mechanisms for export.
- The default MTU value for SONET/SDH interfaces is 4474 bytes; for Gigabit Ethernet and Fast Ethernet interfaces, it is 1500 bytes. If the monitoring station receives packets exceeding 4474 bytes, they are discarded; no fragmentation is performed. Note that the supported MTU size on the Gigabit Ethernet or Fast Ethernet PICs might exceed 1500 bytes, depending on the type of PIC.
- Any incoming traffic that is discarded is not forwarded to packet analyzers.
- The interfaces on the monitoring station that collect intercepted traffic must be configured with Cisco HDLC or PPP encapsulation.
- You must always use a standard interface (for example, one that follows the usual *interface-name-fpc/pic/slot* format) to send flow records to a flow server. Flow data generated by the Monitoring Services or Monitoring Services II PICs will not be delivered to the server across the fxp0 interface.
- You can send version 5 records to multiple flow servers. You can configure up to eight servers and flow traffic is load-balanced between the servers in a round-robin fashion. If one of the servers ceases operation, flow traffic load-balances automatically between the remaining active servers. To configure, include up to eight flow-server statements at the [edit forwarding-options monitoring *group-name* output] hierarchy level.

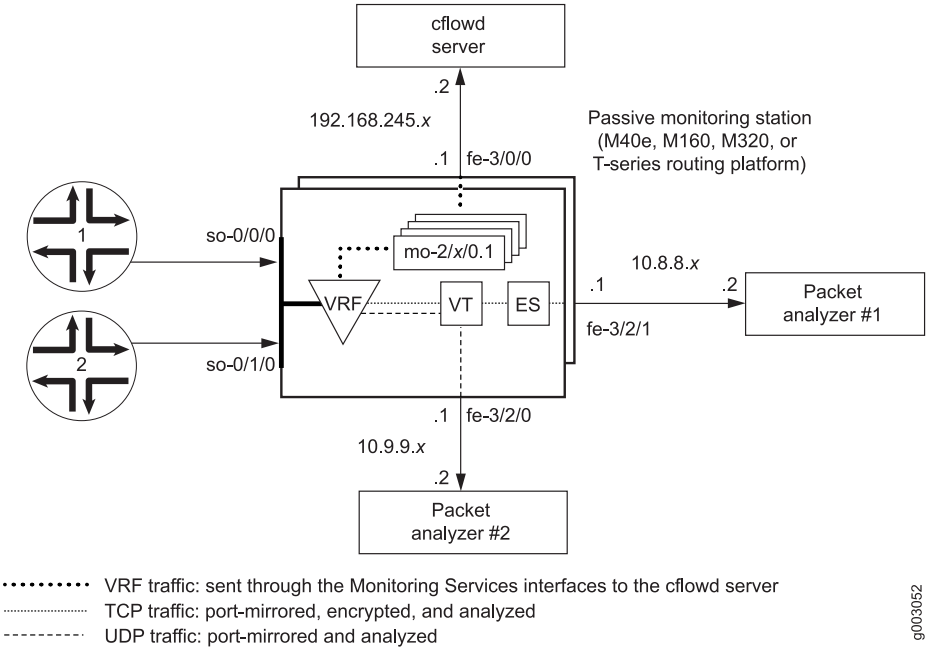
Passive Flow Monitoring Configuration Examples

This section contains configuration examples and commands you can issue to verify a passive flow monitoring configuration:

- Example: Passive Flow Monitoring Configuration on page 319
- Example: Flow Collector Interface Configuration on page 333
- Example: Dynamic Flow Capture Configuration on page 343

Example: Passive Flow Monitoring Configuration

Figure 25: Passive Flow Monitoring—Topology Diagram



In Figure 25 on page 319, traffic enters the monitoring station through interfaces so-0/0/0 and so-0/1/0. After the firewall filter accepts the traffic to be monitored, the packets enter a VRF instance.

The original packets travel within the VRF instance to the Monitoring Services PIC for flow processing. The final flow packets are sent from the monitoring services interfaces out the fe-3/0/0 interface to a flow server.

A copy of the accepted traffic is port-mirrored to the Tunnel PIC. As the copied packets enter the tunnel interface, a second firewall filter separates TCP and UDP packets and places them into two filter-based forwarding instances. The UDP instance directs the UDP packets to a packet analyzer attached to fe-3/2/0. The TCP instance sends the TCP packets to the ES PIC for encryption and the ES PIC sends the packets to a second packet analyzer connected to fe-3/2/1.

Your first step is to define a firewall filter to select packets for monitoring. All filtered traffic must be accepted, and the port-mirror statement at the [edit firewall family inet filter *filter-name* term *term-name* then] hierarchy level facilitates port mirroring.

Next, configure the input SONET/SDH interfaces and apply the firewall filter that you just defined. The passive-monitor-mode statement disables SONET keepalives on the SONET/SDH interfaces and enables passive flow monitoring.

Configure all other interfaces that you will use with the monitoring application, including the monitoring services interfaces, the export interfaces, the tunnel interface, and the ES interface. Once the interfaces are in place, configure a VRF instance and monitoring group to direct the original packets from the input interfaces to the monitoring services interfaces for processing. The resulting flow description packets exit fe-3/0/0 to reach the flow server.

Next, configure statements to port-mirror the monitored traffic to a tunnel interface. Design a firewall filter that selects some of this copied traffic for further analysis and some of the traffic for discarding. In this case, isolate TCP and UDP traffic and direct these two flows into separate filter-based forwarding routing instances. Remember to apply the filter to the tunnel interface to enable the separation of TCP traffic from UDP traffic. Also, import the interface routes into the forwarding instances with a routing table group.

In the filter-based forwarding instances, define static route next hops. The next hop for the TCP instance is the ES interface and the next hop for the UDP instance is the packet analyzer connected to fe-3/2/0. Finally, configure IPSec so that the next hop for the TCP traffic is the second packet analyzer attached to fe-3/2/1.

```
[edit]
interfaces {
   so-0/0/0 { # Traffic enters the router on this interface.
      description " input interface";
      encapsulation ppp;
      unit 0 {
         passive-monitor-mode; # Disables SONET keepalives.
         family inet {
            filter {
               input input-monitoring-filter; # The firewall filter is applied here.
            }
         }
      }
   }
   so-0/1/0 { # Traffic enters the router on this interface.
      description " input interface";
      encapsulation ppp;
      unit 0 {
         passive-monitor-mode;  # Disables SONET keepalives.
         family inet {
            filter {
               input input-monitoring-filter;  # The firewall filter is applied here.
            }
         }
      }
   }
   es-3/1/0 {  # This is where the TCP traffic enters the ES PIC.
```

```
    unit 0 {
      tunnel {
        source 10.8.8.1;
        destination 10.8.8.2;
      }
      family inet {
        ipsec-sa sa-esp;
        address 3.3.3.1/32 {
          destination 3.3.3.2;
        }
      }
    }
}
fe-3/0/0 { # Flow records exit here and travel to the flow server.
    description " export interface to the flow server";
    unit 0 {
      family inet;
      address 192.168.245.1/30;
    }
}
fe-3/2/0 { # This export interface for UDP traffic leads to a packet analyzer.
    description " export interface to the packet analyzer";
    unit 0 {
      family inet {
        address 10.9.9.1/30;
      }
    }
}
fe-3/2/1 {  # This IPSec tunnel source exports TCP traffic to a packet analyzer.
    unit 0 {
      family inet {
        address 10.8.8.1/30;
      }
    }
}
mo-4/0/0 { # This marks the beginning of the monitoring services interfaces.
    unit 0 { # Unit 0 is part of the inet.0 routing table and generates flow records.
      family inet;
    }
    unit 1 { # Unit 1 receives monitored traffic and is part of the VRF instance.
      family inet;
    }
}
mo-4/1/0 {
    unit 0 { # Unit 0 is part of the inet.0 routing table and generates flow records.
      family inet;
    }
    unit 1 { # Unit 1 receives monitored traffic and is part of the VRF instance.
      family inet;
    }
}
mo-4/2/0 {
    unit 0 { # Unit 0 is part of the inet.0 routing table and generates flow records.
      family inet;
    }
    unit 1 { # Unit 1 receives monitored traffic and is part of the VRF instance.
```

```
            family inet;
        }
    }
    mo-4/3/0 {
        unit 0 { # Unit 0 is part of the inet.0 routing table and generates flow records.
            family inet;
        }
        unit 1 { # Unit 1 receives monitored traffic and is part of the VRF instance.
            family inet;
        }
    }
    vt-0/2/0 {  # The tunnel services interface receives the port-mirrored traffic.
        unit 0 {
            family inet {
                filter {
                    input tunnel-interface-filter;  # The filter splits traffic into TCP and UDP
                }
            }
        }
    }
}
forwarding-options {
    monitoring group1 {  # Monitored traffic is processed by the monitoring services
        family inet { # interfaces and flow records are sent to the flow server.
            output {
                export-format cflowd-version-5;
                flow-active-timeout 60;
                flow-inactive-timeout 30;
                flow-server 192.168.245.2 port 2055; # IP address and port for server.
                interface mo-4/0/0.1 {  # Use monitoring services interfaces for output.
                    engine-id 1; # engine and interface-index statements are optional.
                    engine-type 1;
                    input-interface-index 44;
                    output-interface-index 54;
                    source-address 192.168.245.1; # This is the IP address of fe-3/0/0.
                }
                interface mo-4/1/0.1 {
                    engine-id 2; # engine and interface-index statements are optional.
                    engine-type 1;
                    input-interface-index 45;
                    output-interface-index 55;
                    source-address 192.168.245.1; # This is the IP address of fe-3/0/0.
                }
                interface mo-4/2/0.1 {
                    engine-id 3; # engine and interface-index statements are optional.
                    engine-type 1;
                    input-interface-index 46;
                    output-interface-index 56;
                    source-address 192.168.245.1; # This is the IP address of fe-3/0/0.
                }
                interface mo-4/3/0.1 {
                    engine-id 4; # engine and interface-index statements are optional.
                    engine-type 1;
                    input-interface-index 47;
                    output-interface-index 57;
                    source-address 192.168.245.1; # This is the IP address of fe-3/0/0.
```

```
            }
          }
        }
      }
      port-mirroring {  # Copies the traffic and sends it to the Tunnel Services PIC.
        family inet {
          input {
            rate 1;
            run-length 1;
          }
          output {
            interface vt-0/2/0.0;
            no-filter-check;
          }
        }
      }
}
routing-options { # This installs the interface routes into the forwarding instances.
    interface-routes {
      rib-group inet bc-vrf;
    }
    rib-groups {
      bc-vrf {
        import-rib [inet.0 tcp-routing-table.inet.0 udp-routing-table.inet.0];
      }
    }
    forwarding-table {
      export pplb; # Applies per-packet load balancing to the forwarding table.
    }
}
policy-options {
    policy-statement monitoring-vrf-import {
      then reject;
    }
    policy-statement monitoring-vrf-export {
      then reject;
    }
    policy-statement pplb {
      then {
        load-balance per-packet;
      }
    }
}
security { # This sets IPSec options for the ES PIC.
    ipsec {
      proposal esp-sha1-3des {
        protocol esp;
        authentication-algorithm hmac-sha1-96;
        encryption-algorithm 3des-cbc;
        lifetime-seconds 180;
      }
      policy esp-group2 {
        perfect-forward-secrecy {
          keys group2;
        }
        proposals esp-sha1-3des;
```

```
        }
        security-association sa-esp {
            mode tunnel;
            dynamic {
                ipsec-policy esp-group2;
            }
        }
    }
    ike {
        proposal ike-esp {
            authentication-method pre-shared-keys;
            dh-group group2;
            authentication-algorithm sha1;
            encryption-algorithm 3des-cbc;
            lifetime-seconds 180;
        }
        policy 10.8.8.2 {
            mode aggressive;
            proposals ike-esp;
            pre-shared-key ascii-text "$9$qmQnuORrlMBIds2oiH0BIESe";
        }
    }
}
firewall {
    family inet {
        filter input-monitoring-filter { # This filter selects traffic to send into the VRF
            term 1 { # instance and prepares the traffic for port mirroring.
                from {
                    destination-address {
                        10.7.0.0/16;
                    }
                }
                then {
                    port-mirror;
                    accept;
                }
            }
            term 2 {
                from {
                    destination-address {
                        10.6.0.0/16;
                    }
                }
                then accept;
            }
        }
        filter tunnel-interface-filter {  # This filter breaks the port-mirrored traffic into two
            term tcp { # filter-based forwarding instances: TCP packets and UDP packets.
                from {
                    protocol tcp;
                }
                then { # This counts TCP packets and sends them into a TCP instance.
                    count tcp;
                    routing-instance tcp-routing-table;
                }
            }
```

```
        term udp {
            from {
                protocol udp;
            }
            then { # This counts UDP packets and sends them into a UDP instance.
                count udp;
                routing-instance udp-routing-table;
            }
        }
        term rest {
            then {
                count rest;
                discard;
            }
        }
    }
  }
}
routing-instances {
    monitoring-vrf { # This is the VRF instance where you send the traffic. It contains
        instance-type vrf; # the input interface and the monitoring services interfaces.
        interface so-0/0/0.0; # Traffic enters the router on these input interfaces.
        interface so-0/1/0.0;
        interface mo-4/0/0.1;
        interface mo-4/1/0.1; # These are output interfaces (use them as
        interface mo-4/2/0.1; # output interfaces in your monitoring group).
        interface mo-4/3/0.1;
        route-distinguisher 69:1;
        vrf-import monitoring-vrf-import;
        vrf-export monitoring-vrf-export;
        routing-options { # Sends traffic to a group of monitoring services interfaces.
            static {
                route 0.0.0.0/0 next-hop [mo-4/0/0.1 mo-4/1/0.1
                    mo-4/2/0.1 mo-4/3/0.1];
            }
        }
    }
    tcp-routing-table {  # This is the filter-based forwarding instance for TCP traffic.
        instance-type forwarding;
        routing-options {  # The next hop is the ES PIC.
            static {
                route 0.0.0.0/0 next-hop es-3/1/0.0;
            }
        }
    }
    udp-routing-table { # This is the filter-based forwarding instance for UDP traffic.
        instance-type forwarding;
        routing-options { # The next hop is the second packet analyzer.
            static {
                route 0.0.0.0/0 next-hop 10.9.1.2;
            }
        }
    }
}
```

Verifying Your Work

To verify that your configuration is correct, use the following commands on the monitoring station that is configured for passive flow monitoring:

- show route 0/0
- show passive-monitoring error
- show passive-monitoring flow
- show passive-monitoring memory
- show passive-monitoring status
- show passive-monitoring usage

To clear statistics for the show passive-monitoring error and show passive-monitoring flow commands, issue the clear passive-monitoring (all | *interface-name*) command.

You can also view passive flow monitoring status with the Simple Network Management Protocol (SNMP). The following Management Information Base (MIB) tables are supported:

- jnxPMonErrorTable—Corresponds to the show passive-monitoring error command.
- jnxPMonFlowTable—Corresponds to the show passive-monitoring flow command.
- jnxPMonMemoryTable—Corresponds to the show passive-monitoring memory command.

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

```
user@mon-station> show route 0/0
 <skip inet.0>
```

We are only concerned with the routing-instance route.

```
bc-vrf.inet.0:  1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
bc-vrf.inet.0:+ = Active Route, - = Last Active, * = Both
0.0.0.0/0          *[Static/5] 5d 17:34:57
                      via mo-4/0/0.1
                    > via mo-4/1/0.1
                      via mo-4/2/0.1
                      via mo-4/3/0.1
tcp-rt.inet.0: 13 destinations, 13 routes (12 active, 0 holddown, 1
hidden)
+ = Active Route, - = Last Active, * = Both
0.0.0.0/0          *[Static/5] 19:24:39
                    > via es-3/1/0.0
  : <other interface routes>
udp-rt.inet.0: 13 destinations, 13 routes (12 active, 0 holddown, 1
hidden)
+ = Active Route, - = Last Active, * = Both
0.0.0.0/0          *[Static/5] 19:24:39
```

```
                    > to 10.9.1.2 via fe-3/2/0.0
: <other interface routes>
```

NOTE: For all show passive-monitoring commands, the output obtained when using a wildcard (such as *) or the all option is based on the configured interfaces listed at the [edit forwarding-options monitoring *group-name*] hierarchy level. In the output from the configuration example, you see information only for the configured interfaces mo-4/0/0, mo-4/1/0, mo-4/2/0, and mo-4/3/0.

Many of the statements you can configure in a monitoring group, such as engine-id and engine-type, are visible in the output of the show passive-monitoring commands.

Table 16: Output Fields for the show passive-monitoring error Command

Field	Explanation
Packets dropped (no memory)	Number of packets dropped because of memory.
Packets dropped (not IP)	Number of non-IP packets dropped.
Packets dropped (not IPv4)	Number of packets dropped because they failed the IPv4 check.
Packets dropped (header too small)	Number of packets dropped because the packet length or IP header length was too small.
Memory allocation failures	Number of flow record memory allocation failures. A small number reflects failures to replenish the free list. A large number indicates the monitoring station is almost out of memory space.
Memory free failures	Number of flow record memory frees.
Memory free list failures	Number of flow records received from free list that failed. Memory is nearly exhausted or too many new flows greater than 128K are being created in one second.
Memory warning	The flows have exceeded 1 million packets per second (Mpps) on a Monitoring Services PIC or 2 Mpps on a Monitoring Services II PIC. The response can be Yes or No.
Memory overload	The memory has been overloaded. The response is Yes or No.
PPS overload	In packets per second, whether the PIC is receiving more traffic than the configured threshold. The response can be Yes or No.
BPS overload	In bytes per second, whether the PIC is receiving more traffic than the configured threshold. The response can be Yes or No.

```
user@mon-station> show passive-monitoring error all
Passive monitoring interface: mo-4/0/0, Local interface index: 44
  Error information
    Packets dropped (no memory): 0, Packets dropped (not IP): 0
    Packets dropped (not IPv4): 0, Packets dropped (header too small): 0
    Memory allocation failures: 0, Memory free failures: 0
    Memory free list failures: 0
    Memory warning: No, Memory overload: No, PPS overload: No, BPS overload: No

Passive monitoring interface: mo-4/1/0, Local interface index: 45
  Error information
    Packets dropped (no memory): 0, Packets dropped (not IP): 0
    Packets dropped (not IPv4): 0, Packets dropped (header too small): 0
    Memory allocation failures: 0, Memory free failures: 0
    Memory free list failures: 0
    Memory warning: No, Memory overload: No, PPS overload: No, BPS overload: No

Passive monitoring interface: mo-4/2/0, Local interface index: 46
  Error information
    Packets dropped (no memory): 0, Packets dropped (not IP): 0
    Packets dropped (not IPv4): 0, Packets dropped (header too small): 0
    Memory allocation failures: 0, Memory free failures: 0
    Memory free list failures: 0
    Memory warning: No, Memory overload: No, PPS overload: No, BPS overload: No

Passive monitoring interface: mo-4/3/0, Local interface index: 47
  Error information
    Packets dropped (no memory): 0, Packets dropped (not IP): 0
    Packets dropped (not IPv4): 0, Packets dropped (header too small): 0
    Memory allocation failures: 0, Memory free failures: 0
    Memory free list failures: 0
    Memory warning: No, Memory overload: No, PPS overload: No, BPS overload: No
```

Table 17: Output Fields for the show passive-monitoring flow Command

Field	Explanation
Flow packets	Number of packets received by an operational PIC.
Flow bytes	Number of bytes received by an operational PIC.
Flow packets 10-second rate	Number of packets per second handled by the PIC and displayed as a 10-second average.
Flow bytes 10-second rate	Number of bytes per second handled by the PIC and displayed as a 10-second average.
Active flows	Number of currently active flows tracked by the PIC.
Total flows	Total number of flows received by an operational PIC.
Flows exported	Total number of flows exported by an operational PIC.

Table 17: Output Fields for the show passive-monitoring flow Command *(continued)*

Field	Explanation
Flows packets exported	Total number of flow packets exported by an operational PIC.
Flows inactive timed out	Total number of flows that are exported because of inactivity.
Flows active timed out	Total number of long-lived flows that are exported because of an active timeout.

```
user@mon-station> show passive-monitoring flow all
Passive monitoring interface: mo-4/0/0, Local interface index: 44
  Flow information
    Flow packets: 6533434, Flow bytes: 653343400
    Flow packets 10-second rate: 0, Flow bytes 10-second rate: 0
    Active flows: 0, Total flows: 1599
    Flows exported: 1599, Flows packets exported: 55
    Flows inactive timed out: 1599, Flows active timed out: 0

Passive monitoring interface: mo-4/1/0, Local interface index: 45
  Flow information
    Flow packets: 6537780, Flow bytes: 653778000
    Flow packets 10-second rate: 0, Flow bytes 10-second rate: 0
    Active flows: 0, Total flows: 1601
    Flows exported: 1601, Flows packets exported: 55
    Flows inactive timed out: 1601, Flows active timed out: 0

Passive monitoring interface: mo-4/2/0, Local interface index: 46
  Flow information
    Flow packets: 6529259, Flow bytes: 652925900
    Flow packets 10-second rate: 0, Flow bytes 10-second rate: 0
    Active flows: 0, Total flows: 1599
    Flows exported: 1599, Flows packets exported: 55
    Flows inactive timed out: 1599, Flows active timed out: 0

Passive monitoring interface: mo-4/3/0, Local interface index: 47
  Flow information
    Flow packets: 6560741, Flow bytes: 656074100
    Flow packets 10-second rate: 0, Flow bytes 10-second rate: 0
    Active flows: 0, Total flows: 1598
    Flows exported: 1598, Flows packets exported: 55
    Flows inactive timed out: 1598, Flows active timed out: 0
```

Table 18: Output Fields for the show passive-monitoring memory Command

Field	Explanation
Allocation count	Number of flow records allocated.
Free count	Number of flow records freed.

Table 18: Output Fields for the show passive-monitoring memory Command *(continued)*

Field	Explanation
Maximum allocated	Maximum number of flow records allocated since the monitoring station booted. This number represents the peak number of flow records allocated at a time.
Allocations per second	Flow records allocated per second during the last statistics interval on the PIC.
Frees per second	Flow records freed per second during the last statistics interval on the PIC.
Total memory used	Total amount of memory currently used (in bytes).
Total memory free	Total amount of memory currently free (in bytes).

```
user@mon-station> show passive-monitoring memory all

Passive monitoring interface: mo-4/0/0, Local interface index: 44
  Memory utilization
    Allocation count: 1600, Free count: 1599, Maximum allocated: 1600
    Allocations per second: 3200, Frees per second: 1438
    Total memory used (in bytes): 103579176, Total memory free (in bytes): 163914184

Passive monitoring interface: mo-4/1/0, Local interface index: 45
  Memory utilization
    Allocation count: 1602, Free count: 1601, Maximum allocated: 1602
    Allocations per second: 3204, Frees per second: 1472
    Total memory used (in bytes): 103579176, Total memory free (in bytes): 163914184

Passive monitoring interface: mo-4/2/0, Local interface index: 46
  Memory utilization
    Allocation count: 1600, Free count: 1599, Maximum allocated: 1600
    Allocations per second: 3200, Frees per second: 1440
    Total memory used (in bytes): 103579176, Total memory free (in bytes): 163914184

Passive monitoring interface: mo-4/3/0, Local interface index: 47
  Memory utilization
    Allocation count: 1599, Free count: 1598, Maximum allocated: 1599
    Allocations per second: 3198, Frees per second: 1468
    Total memory used (in bytes): 103579176, Total memory free (in bytes): 163914184
```

Table 19: Output Fields for the show passive-monitoring status Command

Field	Explanation
Interface state	Indicates whether the interface is monitoring (operating properly), disabled (administratively disabled), or not monitoring (not configured).

Table 19: Output Fields for the show passive-monitoring status Command *(continued)*

Field	Explanation
Group index	Integer that represents the monitoring group of which the PIC is a member. (This does not indicate the number of monitoring groups.)
Export interval	Configured export interval for flow records, in seconds.
Export format	Configured export format (only v5 is currently supported).
Protocol	Protocol the PIC is configured to monitor (only IPv4 is currently supported).
Engine type	Configured engine type that is inserted in output flow packets.
Engine ID	Configured engine ID that is inserted in output flow packets.
Route record count	Number of routes recorded.
IFL to SNMP index count	Number of logical interfaces mapped to an SNMP index.
AS count	Number of AS boundaries that the flow has crossed.
Time set	Indicates whether the time stamp is in place.
Configuration set	Indicates whether the monitoring configuration is set.
Route record set	Indicates whether routes are being recorded
IFL SNMP map set	Indicates whether logical interfaces are being mapped to an SNMP index.

```
user@mon-station> show passive-monitoring status all
Passive monitoring interface: mo-4/0/0, Local interface index: 44
  Interface state: Monitoring
  Group index: 0
  Export interval: 15 secs, Export format: cflowd v5
  Protocol: IPv4, Engine type: 1, Engine ID: 1
  Route record count: 13, IFL to SNMP index count: 30, AS count: 1
  Time set: Yes, Configuration set: Yes
  Route record set: Yes, IFL SNMP map set: Yes

Passive monitoring interface: mo-4/1/0, Local interface index: 45
  Interface state: Monitoring
  Group index: 0
  Export interval: 15 secs, Export format: cflowd v5
  Protocol: IPv4, Engine type: 1, Engine ID: 2
  Route record count: 13, IFL to SNMP index count: 30, AS count: 1
  Time set: Yes, Configuration set: Yes
  Route record set: Yes, IFL SNMP map set: Yes

Passive monitoring interface: mo-4/2/0, Local interface index: 46
  Interface state: Monitoring
  Group index: 0
  Export interval: 15 secs, Export format: cflowd v5
  Protocol: IPv4, Engine type: 1, Engine ID: 3
  Route record count: 13, IFL to SNMP index count: 30, AS count: 1
```

```
  Time set: Yes, Configuration set: Yes
  Route record set: Yes, IFL SNMP map set: Yes

Passive monitoring interface: mo-4/3/0, Local interface index: 47
  Interface state: Monitoring
  Group index: 0
  Export interval: 15 secs, Export format: cflowd v5
  Protocol: IPv4, Engine type: 1, Engine ID: 4
  Route record count: 13, IFL to SNMP index count: 30, AS count: 1
  Time set: Yes, Configuration set: Yes
  Route record set: Yes, IFL SNMP map set: Yes
```

Table 20: Output Fields for the show passive-monitoring usage Command

Field	Explanation
Uptime	Time, in milliseconds, that the PIC has been operational.
Interrupt time	Cumulative time that the PIC spent in processing packets since the last PIC reset.
Load (5 second)	CPU load on the PIC averaged over 5 seconds. The number is a percentage obtained by dividing the time spent on active tasks by the total elapsed time.
Load (1 minute)	CPU load on the PIC averaged over 1 minute. The number is a percentage obtained by dividing the time spent on active tasks by the total elapsed time.

```
user@mon-station> show passive-monitoring usage *
Passive monitoring interface: mo-4/0/0, Local interface index: 44
  CPU utilization
    Uptime: 653155 milliseconds, Interrupt time: 40213754 microseconds
    Load (5 second): 20%, Load (1 minute): 17%

Passive monitoring interface: mo-4/1/0, Local interface index: 45
  CPU utilization
    Uptime: 652292 milliseconds, Interrupt time: 40223178 microseconds
    Load (5 second): 22%, Load (1 minute): 15%

Passive monitoring interface: mo-4/2/0, Local interface index: 46
  CPU utilization
    Uptime: 649491 milliseconds, Interrupt time: 40173645 microseconds
    Load (5 second): 22%, Load (1 minute): 10098862%

Passive monitoring interface: mo-4/3/0, Local interface index: 47
  CPU utilization
    Uptime: 657328 milliseconds, Interrupt time: 40368704 microseconds
    Load (5 second): 1%, Load (1 minute): 15%
```

Example: Flow Collector Interface Configuration

Figure 26: Flow Collector Interface Topology Diagram

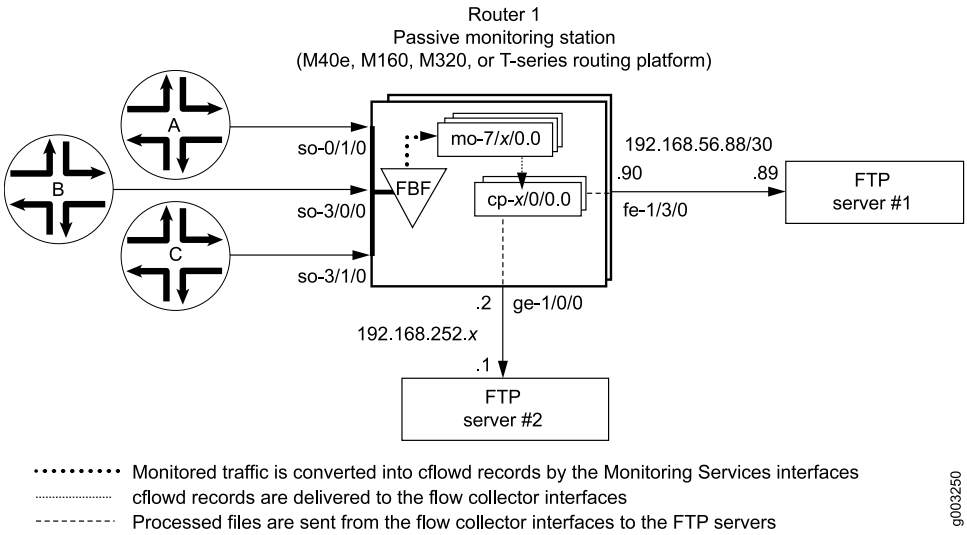


Figure 26 on page 333 shows the path traveled by monitored traffic as it passes through the routing platform. Packets arrive at input interfaces so-0/1/0, so-3/0/0, and so-3/1/0. The raw packets are directed into a filter-based forwarding routing instance and processed into flow records by the monitoring services interfaces mo-7/1/0, mo-7/2/0, and mo-7/3/0. The flow records are compressed into files at the flow collector interfaces cp-6/0/0 and cp-7/0/0 and sent to the FTP server for analysis. Finally, a mandatory class-of-service (CoS) configuration is applied to export channels 0 and 1 on the flow collector interfaces to manage the outgoing processed files.

Router 1

```
[edit]
chassis {
   fpc 6 {
      pic 0 {
         monitoring-services {
            application flow-collector; # This converts a Monitoring Services II PIC
         }        # into a flow collector interface.
      }
   }
   fpc 7 {
      pic 0 {
         monitoring-services {
            application flow-collector; # This converts a Monitoring Services II PIC
         }           # into a flow collector interface.
      }
   }
}
interfaces {
   cp-6/0/0 {
      unit 0 {# Logical interface .0 on a flow collector interface is export
```

```
        family inet { # channel 0 and sends records to the FTP server.
          filter {
            output cp-ftp; # Apply the CoS filter here.
          }
          address 10.0.0.1/32 {
            destination 10.0.0.2;
          }
        }
      }
      unit 1 { # Logical interface .1 on a flow collector interface is export
        family inet { # channel 1 and sends records to the FTP server.
          filter {
            output cp-ftp; # Apply the CoS filter here.
          }
          address 10.1.1.1/32 {
            destination 10.1.1.2;
          }
        }
      }
      unit 2 { # Logical interface .2 on a flow collector interface is the flow
        family inet { # receive channel that communicates with the Routing Engine.
          address 10.2.2.1/32 { # Do not apply a CoS filter on logical interface .2.
            destination 10.2.2.2;
          }
        }
      }
    }
    cp-7/0/0 {
      unit 0 { # Logical interface .0 on a flow collector interface is export
        family inet { # channel 0 and sends records to the FTP server.
          filter {
            output cp-ftp; # Apply the CoS filter here.
          }
          address 10.3.3.1/32 {
            destination 10.3.3.2;
          }
        }
      }
      unit 1 { # Logical interface .1 on a flow collector interface is export
        family inet { # channel 1 and sends records to the FTP server.
          filter {
            output cp-ftp; # Apply the CoS filter here.
          }
          address 10.4.4.1/32 {
            destination 10.4.4.2;
          }
        }
      }
      unit 2 { # Logical interface .2 on a flow collector interface is the flow
        family inet { # receive channel that communicates with the Routing Engine.
          address 10.5.5.1/32 { # Do not apply a CoS filter on logical interface .2.
            destination 10.5.5.2;
          }
        }
      }
    }
```

```
fe-1/3/0 { # This is the exit interface leading to the first FTP server.
   unit 0 {
      family inet {
         address 192.168.56.90/30;
      }
   }
}
ge-1/0/0 { # This is the exit interface leading to the second FTP server.
   unit 0 {
      family inet {
         address 192.168.252.2/24;
      }
   }
}
mo-7/1/0 {  # This is the first interface that creates flow records.
   unit 0 {
      family inet;
   }
}
mo-7/2/0 { # This is the second interface that creates flow records.
   unit 0 {
      family inet;
   }
}
mo-7/3/0 { # This is the third interface that creates flow records.
   unit 0 {
      family inet;
   }
}
so-0/1/0 { # This is the first input interface that receives traffic to be monitored.
   encapsulation ppp;
   unit 0 {
      passive-monitor-mode; # This allows the interface to be passively monitored.
      family inet {
         filter {
            input catch; # The filter-based forwarding filter is applied here.
         }
      }
   }
}
so-3/0/0 { # This is the second interface that receives traffic to be monitored.
   encapsulation ppp;
   unit 0 {
      passive-monitor-mode; # This allows the interface to be passively monitored.
      family inet {
         filter {
            input catch; # The filter-based forwarding filter is applied here.
         }
      }
   }
}
so-3/1/0 { # This is the third interface that receives traffic to be monitored.
   encapsulation ppp;
   unit 0 {
      passive-monitor-mode; # This allows the interface to be passively monitored.
      family inet {
```

```
            filter {
               input catch; # The filter-based forwarding filter is applied here.
            }
         }
      }
   }
}
forwarding-options {
   monitoring group1 { # Always define your monitoring group here.
      family inet {
         output {
            export-format cflowd-version-5;
            flow-active-timeout 60;
            flow-inactive-timeout 15;
            flow-export-destination collector-pic; # Sends records to the flow collector.
            interface mo-7/1/0.0 {
               source-address 192.168.252.2;
            }
            interface mo-7/2/0.0 {
               source-address 192.168.252.2;
            }
            interface mo-7/3/0.0 {
               source-address 192.168.252.2;
            }
         }
      }
   }
}
routing-options {
   interface-routes {
      rib-group inet common;
   }
   rib-groups {
      common {
         import-rib [ inet.0 fbf_instance.inet.0 ];
      }
   }
   forwarding-table {
      export pplb;
   }
}
policy-options {
   policy-statement pplb {
      then {
         load-balance per-packet;
      }
   }
}
class-of-service { # A class-of-service configuration for the flow collector interface
   interfaces { # is mandatory when implementing flow collector services.
      cp-6/0/0 {
         scheduler-map cp-map;
      }
      cp-7/0/0 {
         scheduler-map cp-map;
      }
```

```
    }
    scheduler-maps {
        cp-map {
            forwarding-class best-effort scheduler Q0;
            forwarding-class expedited-forwarding scheduler Q1;
            forwarding-class network-control scheduler Q3;
        }
    }
    schedulers {
        Q0 {
            transmit-rate remainder;
            buffer-size percent 90;
        }
        Q1 {
            transmit-rate percent 5;
            buffer-size percent 5;
            priority strict-high;
        }
        Q3 {
            transmit-rate percent 5;
            buffer-size percent 5;
        }
    }
}
firewall {
    family inet {
        filter cp-ftp { # This filter provides CoS for flow collector interface traffic.
            term t1 {
                then forwarding-class expedited-forwarding;
            }
        }
    }
    filter catch { # This firewall filter sends incoming traffic into the
        interface-specific; # filter-based forwarding routing instance.
        term def {
            then {
                count counter;
                routing-instance fbf_instance;
            }
        }
    }
}
routing-instances {
    fbf_instance { # This instance sends traffic to the monitoring services interface.
        instance-type forwarding;
        routing-options {
            static {
                route 0.0.0.0/0 next-hop mo-7/1/0.0;
            }
        }
    }
}
services {
    flow-collector { # Define properties for flow collector interfaces here.
        analyzer-address 10.10.10.1; # This is the IP address of the analyzer.
        analyzer-id server1; # This helps to identify the analyzer.
```

```
        retry 3; # Maximum number of attempts by the PIC to send a file transfer log.
        retry-delay 30; # The time interval between attempts to send a file transfer log.
        destinations { # This defines the FTP servers that receive flow collector output.
            "ftp://user@192.168.56.89//tmp/collect1/" {  # The primary FTP server.
                password "$9$IXJK8xN-w2oZdbZDHmF3OO1"; # SECRET-DATA
            }
            "ftp://user@192.168.252.1//tmp/collect2/" { # The second FTP server.
                password "$9$eIbvL7-dsgaGVwGjkP3nOBI"; # SECRET-DATA
            }
        }
        file-specification {  # Define sets of flow collector characteristics here.
            def-spec {
            }
            data-format flow-compressed; # The default compressed output format.
        }
        f1 {
            name-format "cFlowd-py69Ni69-0-%D_%T-%I_%N.bcp.bi.gz";
            data-format flow-compressed; # The default compressed output format.
            transfer timeout 1800 record-level 1000000; # Here are configured values.
        }
    }
    interface-map { # Allows you to map interfaces to flow collector interfaces.
        file-specification def-spec; # Flows generated for default traffic are sent to the
        collector cp-7/0/0; # default flow collector interface cp-7/0/0.
        so-0/1/0.0 {# Flows generated for the so-0/1/0 interface are sent
            collector cp-6/0/0; # to cp-6/0/0, and the file-specification used is “default”.
        }
        so-3/0/0.0 { # Flows generated for the so-3/0/0 interface are sent
            file-specification f1; # to cp-6/0/0, and the file-specification used is "f1."
            collector cp-6/0/0;
        }
        so-3/1/0.0; # Because no settings are defined, flows generated for this
    }
    transfer-log-archive { # Sends flow collector interface log files to an FTP server.
        filename-prefix so_3_0_0_log;
        maximum-age 15;
        archive-sites {
            "ftp://user@192.168.56.89//tmp/transfers/" {
                password "$9$IFaEyevMXNVsWLsgaU.m6/C";
            }
        }
    }
}
```

Verifying Your Work

To verify that your flow collector configuration is working, use the following commands on the monitoring station that is configured for flow collection:

- clear services flow-collector statistics
- request services flow-collector change-destination (primary | secondary)
- request services flow-collector test-file-transfer
- show services flow-collector file interface (detail | extensive | terse)

- show services flow-collector (detail | extensive)
- show services flow-collector input interface (detail | extensive | terse)

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

```
user@router1> show services flow-collector input interface cp-6/0/0 detail
Interface                      Packets        Bytes
mo-7/1/0.0                        6170      8941592

user@router1>  show services flow-collector interface all detail
Flow collector interface: cp-6/0/0
Interface state: Collecting flows
  Packets     Bytes     Flows Uncompressed  Compressed     FTP bytes FTP files
                                    Bytes       Bytes
     6736   9757936    195993     21855798     3194148             0         0
Flow collector interface: cp-7/0/0
Interface state: Collecting flows
  Packets     Bytes     Flows Uncompressed  Compressed     FTP bytes FTP files
                                    Bytes       Bytes
        0         0         0            0           0             0         0

user@router1>  show services flow-collector input interface cp-6/0/0 extensive
Interface                      Packets        Bytes
mo-7/1/0.0                        6260      9074096

user@router1>  show services flow-collector interface cp-6/0/0 extensive
Flow collector interface: cp-6/0/0
Interface state: Collecting flows
Memory:
    Used: 19593212, Free: 479528656
Input:
    Packets: 6658, per second: 0, peak per second: 0
    Bytes: 9647752, per second: 12655, peak per second: 14311
    Flow records processed: 193782, per second: 252, peak per second: 287
Allocation:
    Blocks allocated: 174, per second: 0, peak per second: 0
    Blocks freed: 0, per second: 0, peak per second: 0
    Blocks unavailable: 0, per second: 0, peak per second: 0
Files:
    Files created: 1, per second: 0, peak per second: 0
    Files exported: 0, per second: 0, peak per second: 0
    Files destroyed: 0, per second: 0, peak per second: 0
Throughput:
    Uncompressed bytes: 21075152, per second: 52032, peak per second: 156172
    Compressed bytes: 3079713, per second: 7618, peak per second: 22999
Packet drops:
    No memory: 0, Not IP: 0
    Not IPv4: 0, Too small: 0
    Fragments: 0, ICMP: 0
    TCP: 0, Unknown: 0
    Not JUNOS flow: 0
File Transfer:
    FTP bytes: 0, per second: 0, peak per second: 0
    FTP files: 0, per second: 0, peak per second: 0
    FTP failure: 0
Export channel: 0
    Current server: Secondary
    Primary server state: OK, Secondary server state: OK
```

```
Export channel: 1
    Current server: Secondary
    Primary server state: OK, Secondary server state: OK

user@router1>  show services flow-collector file interface cp-6/0/0 terse
File name                                                    Flows State
cFlowd-py69Ni69-0-20031112_014301-so_3_0_0_0.bcp.bi.gz          185643 Active

user@router1>  show services flow-collector file interface cp-6/0/0 detail
Filename: cFlowd-py69Ni69-0-20031112_014301-so_3_0_0_0.bcp.bi.gz
  Throughput:
    Flow records: 187067, Uncompressed bytes: 21121960, Compressed bytes: 2965643

  Status:
    State: Active, Transfer attempts: 0

user@router1>  show services flow-collector file interface cp-6/0/0 extensive
Filename: cFlowd-py69Ni69-0-20031112_014301-so_3_0_0_0.bcp.bi.gz
  Throughput:
    Flow records: 188365, per second: 238, peak per second: 287
    Uncompressed bytes: 21267756, per second: 27007, peak per second: 32526
    Compressed bytes: 2965643, per second: 0, peak per second: 22999
  Status:
    Compressed blocks: 156, Block count: 156
    State: Active, Transfer attempts: 0
```

To clear statistics for a flow collector interface, issue the clear services flow-collector statistics interface (all | *interface-name*) command.

Another useful flow collector option allows you to change the FTP server from primary to secondary and test for FTP transfers. To force the flow collector interface to use a primary or secondary FTP server, include the primary or secondary option when you issue the request services flow-collector change-destination interface cp-*fpc/pic/port* command.

If you configure only one primary server and issue this command with the primary option, you receive the error message "Destination change not needed." If the secondary server is not configured and you issue this command with the secondary option, you receive the error message "Destination not configured." Otherwise, when both servers are configured properly, successful output appears as follows.

```
user@router1> request services flow-collector change-destination interface
cp-6/0/0 primary
Flow collector interface: cp-6/0/0
Interface state: Collecting flows
Destination change successful

user@router1>  request services flow-collector change-destination interface
cp-6/0/0 secondary
Flow collector interface: cp-6/0/0
Interface state: Collecting flows
Destination change successful
```

Other options for the request services flow-collector change-destination interface cp-*fpc/pic/port* command are immediately (which forces an instant switchover), gracefully (the default behavior that allows a gradual switchover), clear-files (which purges existing data files), and clear-logs (which purges existing log files).

To verify that transfer log files are being scheduled for delivery to the FTP servers, issue the request services flow-collector test-file-transfer *filename* interface cp-*fpc/pic/port* command. Include the desired export channel (zero or one) and target FTP server (primary or secondary) with this command.

```
user@router> request services flow-collector test-file-transfer test_file
interface cp-6/0/0 channel-one primary
Flow collector interface: cp-6/0/0
Interface state: Collecting flows
Response: Test file transfer successfully scheduled
```

Another way you can check for the success of your file transfers is by analyzing the transfer log. A transfer log sends detailed information about files that are collected and processed by the flow collector interface. Table 21 on page 341 explains the various fields available in the transfer log.

Table 21: Flow Collector Interface Transfer Log Fields

Field	Explanation
fn	Filename
sz	File size
nr	Number of records
ts	Time stamp with the format of year (4 digits), month (2 digits), day (2 digits), hours (2 digits), minutes (2 digits), and seconds (2 digits).
sf	Success flag—The values are 1 for success and 0 for failure.
ul	Server URL
rc	FTP result code
er	FTP error text
tt	Transfer time

This is an example of a successful transfer log:

```
fn="cFlowd-py69Ni69-0-20040227_230438-at_4_0_0_4_3.bcp.bi.gz":sz=552569
:nr=20000:ts="20040227230855":sf=1:ul="ftp://10.63.152.1/tmp/server1/:"rc=250:
er="":tt=3280
```

This is an example of a transfer log when an FTP session fails:

```
fn="cFlowd-py69Ni69-0-20040227_230515-at_4_0_0_2_8.bcp.bi.gz":sz=560436
:nr=20000:ts="20040227230855":sf=1:ul="ftp://10.63.152.1/tmp/server1/:"rc=250
:er="":tt=3290
```

As the flow collector interface receives and processes flow records, the PIC services logging process (fsad) handles the following tasks:

- When the flow collector interface transfers a file to the FTP server, a temporary log file is created in the /var/log/flowc directory. The temporary log file has this filenaming convention:

  <hostname>_<filename_prefix>_YYYYMMDD_hhmmss.tmp

  hostname is the hostname of the transfer server, *filename_prefix* is the same value defined with the filename-prefix statement at the [edit services flow-collector transfer-log-archive] hierarchy level, *YYYYMMDD* is the year, month, and date, and *hhmmss* is the timestamp indicating hours, minutes, and seconds.

- After the log file has been stored in the routing platform for the length of time specified by the maximum-age statement at the [edit services flow-collector transfer-log-archive] hierarchy level (the default is 120 minutes), the temporary log file is converted to an actual log file and the temporary file is deleted. The new log file retains the same naming conventions, except the extension is **.log*.

- When the final log file is created and compressed, the PIC services logging process (fsad) tries to send the log file from the /var/log/flowc directory to an FTP server. You can specify up to five FTP servers to receive the log files by including the archive-sites statement at the [edit services flow-collector transfer-log-archive] hierarchy level. The logging process attempts to send the log file to one server at a time, in order of their appearance in the configuration. Upon the first successful transfer, the log file is deleted and the logging process stops sending log files to the remaining FTP servers in the list.

- If the log file transfer is not successful, the log file is moved to the /var/log/flowc/failed directory. Every 30 minutes, the logging process tries to resend the log files. After the log files are transferred successfully, they are deleted from the /var/log/flowc/failed directory.

NOTE: If the memory for a flow collector interface is full, the interface might drop incoming packets.

After the flow collector interface successfully delivers the processed information file to the FTP server, you can analyze the file. The file contains detailed information about the flows collected and processed by the flow collector interface. Table 22 on page 342 explains the various fields available in the flow collector interface file.

Table 22: Flow Collector Interface File Fields in Order of Appearance

Field	Explanation
linkDir	Link directory—A randomly generated number used to identify the record
analyzer-address	Analyzer address
analyzer-ID	Analyzer identifier

Table 22: Flow Collector Interface File Fields in Order of Appearance *(continued)*

Field	Explanation
ifAlias	Interface identifier
source-address	Source address
destination-address	Destination address
packets	Number of packets
bytes	Number of bytes
start-time	Start time
end-time	End time
source-port	Source port
destination-port	Destination port
tcp_flag	TCP flag
protocol	IP protocol number
src_AS_number	Source AS number
dst_AS_number	Destination AS number

This is an example of output from a flow collector interface file:

```
11799241612374557782|10.10.10.1|server1|at_4_0_0_4|192.168.10.100|10.0.0.1|8|
3136|1077926402|1077926402|8224|12336|27|6|0|0
```

Example: Dynamic Flow Capture Configuration

The following example shows a complete dynamic flow capture configuration. On Router 1, configure the dynamic flow capture interface, the interfaces that connect to the control source and content destination, and the interface that receives passively monitored traffic. Then, configure the capture group and specify your control source and content destination requirements. Next, configure filter-based forwarding (FBF) to send monitored traffic to logical unit 1 of the dynamic flow capture interface. Finally, configure a firewall filter and routing table groups to complete the configuration.

```
[edit]
interfaces {
   dfc-0/0/0 { # DFC PIC that processes requests from the control source.
      unit 0 {
         family inet {
            address 2.1.0.0/32 { # Address of the Routing Engine for the DFC PIC.
               destination 10.36.100.1; # Address of DFC PIC; used by
```

```
            } # the control source to communicate with the monitoring station.
          }
        }
        unit 1 { # This logical interface receives data packets.
          family inet;
        }
        unit 2 {  # This logical interface sends out copies of matched packets.
          family inet;
        }
      }
      fe-4/1/2 { # Interface that receives filtering requests from cs1.
        unit 0 {
          family inet {
            address 10.36.41.2/30;
          }
        }
      }
      ge-7/0/0 {  # Interface that sends monitored packets to cd1.
        unit 0 {
          family inet {
            address 10.36.70.1/30;
          }
        }
      }
      so-1/2/0 {  # Interface that receives traffic to be monitored.
        encapsulation ppp;
        unit 0 {
          passive-monitor-mode; # Enables this interface to be passively monitored.
          family inet {
            filter {
              input catch;
            }
          }
        }
      }
    }
    services {
      dynamic-flow-capture {
        capture-group g1 {
          interfaces dfc-0/0/0; # Specifies which interface to use for DFC processing.
          input-packet-rate-threshold 90k; # Traffic threshold for system log messages.
          pic-memory-threshold percentage 80; # Memory threshold for log messages.
          control-source cs1 { # Specifies addresses and ports for the control source.
            source-addresses 10.36.41.1;
            service-port 2400;
            notification-targets {
              10.36.41.1 port 2100;
            }
            shared-key "$9$ASxdsYoX7wg4aHk";
            allowed-destinations cd1;
          }
          content-destination cd1 { # Specifies content destination addresses and TTL.
            address 10.36.70.2;
            ttl 244;
          }
        }
```

```
    }
  }
  firewall {
    filter catch { # Places monitored traffic into the filter-based forwarding instance.
      interface-specific;
      term def {
        then {
          count counter;
          routing-instance fbf_inst;
        }
      }
    }
  }
  routing-instances {
    fbf_inst { # Sends matching traffic to the DFC PIC for processing.
      instance-type forwarding;
      routing-options {
        static {
          route 0.0.0.0/0 next-hop dfc-0/0/0.1;
        }
      }
    }
  }
  routing-options {
    interface-routes {
      rib-group inet common;
    }
    rib-groups {
      common { # Shares routes between the instance and the main routing table.
        import-rib [ inet.0 fbf_inst.inet.0 ];
      }
    }
    forwarding-table {
      export pplb;
    }
  }
```

Verifying Your Work

To verify that your dynamic flow capture configuration is operating correctly, issue the following command:

show services dynamic-flow-capture capture-group *group-name* control-source source-identifier *source-id* (detail)

The following section shows the output of this command when used with the configuration example.

Router 1

```
user@router1> show services dynamic-flow-capture control-source capture-group g1 source-identifier cs2
detail
```

```
Capture group: g1, Control source: cs2
Criteria added: 1, Criteria add failed: 0
Active criteria: 2
Static criteria: 0, Dynamic criteria: 2
Control protocol requests: 3
      Add        Delete          List       Refresh         No-op
Requests           1             0             1             0             1
Failed             0             0             0             0             0

Add request rate: 0
Add request peak rate: 1
Bandwidth across all criteria: 0
Total notifications: 0
Restart: 0, Rollover: 0, No-op: 0, Timeout: 0, Congestion: 0, Congestion delete: 0,
Dups dropped: 0
Criteria deleted: 0
Timeout idle: 0, Timeout total: 0, Packets: 0, Bytes: 0
Sequence number: 242
```

To clear dynamic flow capture criteria belonging to a particular control source, issue the clear services dynamic-flow-capture command. For more information on other dynamic flow capture-related operational mode commands, see the *JUNOS System Basics and Services Command Reference*.

Configuring Active Flow Monitoring

In active flow monitoring, the routing platform participates in both the monitoring application and in the normal routing functionality of the network. Although the Monitoring Services PIC was designed initially for use as an offline passive flow monitoring tool, it can also be used in an active flow monitoring topology.

Table 23 on page 346 shows which Juniper Networks PICs and corresponding routing platforms support active flow monitoring. For more information on Juniper Networks PICs, see the PIC guide that corresponds to your routing platform.

Table 23: Passive and Active Flow Monitoring PIC Support

PIC Type and Service	J-series	M5/M10	M7i/M10i	M20	M40e	M120	M160	T-series/ M320	TX Matrix
Monitoring Services PIC: active flow monitoring	No	Yes (version 8 only)	Yes	Yes	Yes	No	Yes (version 8 only)	No	No
Monitoring Services II PIC: flow collection services	No	No	No	No	Yes	No	Yes (version 8 only)	No	No
Adaptive Services PIC: active flow monitoring	No	Yes (version 8 only)	Yes	Yes	Yes	No	Yes (version 8 only)	No	No

Table 23: Passive and Active Flow Monitoring PIC Support *(continued)*

PIC Type and Service	J-series	M5/M10	M7i/M10i	M20	M40e	M120	M160	T-series/ M320	TX Matrix
Adaptive Services II PIC: active flow monitoring	No	Yes (version 8 only)	Yes	Yes	Yes	Yes	Yes (version 8 only)	Yes	Yes
Adaptive Services II PIC: flow-tap services	No	No	Yes	Yes	Yes	Yes	No	Yes	No
MultiServices 100 PIC: active flow monitoring	No	No	Yes	No	Yes	No	No	Yes	Yes
MultiServices 400 PIC: active flow monitoring	No	No	No	No	Yes	Yes	No	Yes	Yes
MultiServices 500 PIC: active flow monitoring	No	No	No	No	Yes	Yes	No	Yes	Yes
JUNOS software-enabled active flow monitoring	Yes (version 8 only)	No	No	No	No	No	No	No	No

Specified packets can be filtered and sent to the monitoring interface. For the Monitoring Services PIC, the interface name contains the mo- prefix. For the Adaptive Services PICs and MultiServices PICs, the interface name contains the sp- prefix.

NOTE: If you upgrade from the Monitoring Services PIC to the Adaptive Services PIC or MultiServices PIC for active flow monitoring, you must modify the interface name of your monitoring interface from mo-*fpc/pic/port* to sp-*fpc/pic/port*.

The major active flow monitoring actions you can configure at the [edit forwarding-options] hierarchy level are as follows:

- Sampling, with the [edit forwarding-options sampling] hierarchy. This option extracts limited information (such as the source and destination IP address) from a copy of some of the packets in a flow, while the original packets are forwarded to the intended destination.
- Templates, with the [edit forwarding-options sampling] and [edit services monitoring] hierarchies. With active flow monitoring version 9, you can use templates to organize the data gathered from sampling.
- Discard accounting, with the [edit forwarding-options accounting] hierarchy. This option quarantines unwanted packets, creates flow monitoring records that describe the packets, and discards the packets instead of forwarding them.

- Port mirroring, with the [edit forwarding-options port-mirroring] hierarchy. This option makes one full copy of all packets in a flow and delivers the copy to a single destination.
- Multiple port mirroring, with the [edit forwarding-options next-hop-group] hierarchy. This option allows multiple copies of selected traffic to be delivered to multiple destinations. (Multiple port mirroring requires a Tunnel Services PIC.)
- Flow-tap services processing, with the [edit services flow-tap] hierarchy. This option sends copies of packets that match dynamic filter criteria to one or more content destinations.

Unlike passive flow monitoring, you do not need to configure a monitoring group. Instead, you can send filtered packets to a monitoring services or adaptive services interface (mo- or sp-) by using sampling or discard accounting. Optionally, you can configure port mirroring or multiple port mirroring to direct packets to additional interfaces.

These active flow monitoring options provide a wide variety of actions that can be performed on network traffic flows. However, the following restrictions apply:

- The routing platform can perform either sampling *or* port mirroring at any one time.
- The routing platform can perform either forwarding *or* discard accounting at any one time.

Because the Monitoring Services PIC, Adaptive Services PIC, and MultiServices PIC allow only one action to be performed at any one time, the following configuration options are available:

- Sampling and forwarding
- Sampling and discard accounting
- Port mirroring and forwarding
- Port mirroring and discard accounting
- Sampling and port mirroring on different sets of traffic

To configure active flow monitoring, complete these steps:

- Defining a Firewall Filter to Select Traffic for Active Flow Monitoring on page 349
- Configuring the Interfaces That Will Be Actively Monitored on page 350
- Enabling the Monitoring Services, Adaptive Services, or Multiservices Interfaces and the Export Interface on page 350
- Collecting Flow Records on page 351
- Option: Configuring Port Mirroring on page 357
- Option: Configuring Port Mirroring with Filter-Based Forwarding and a Monitoring Group on page 358

- Option: Sending Traffic to Multiple Export Interfaces by Using Next-Hop Groups on page 359
- Option: Using the Flow-Tap Application to Send Packets to a Mediation Device on page 360

Defining a Firewall Filter to Select Traffic for Active Flow Monitoring

The first step in active flow monitoring is to configure the match conditions for acceptable traffic or quarantined traffic. Common match actions for active flow monitoring include sample, discard accounting, port-mirror, and accept. To configure, include the desired action statements and a counter as part of the then statement in a firewall filter and apply the filter to an interface.

In sampling, the routing platform reviews a portion of the traffic and sends reports about this sample to the flow monitoring server. Discard accounting traffic is counted and monitored, but not forwarded out of the routing platform. Port-mirrored traffic is copied and sent to another interface. Accepted traffic is forwarded to the intended destination.

Most of these match combinations are valid. However, you can either port-mirror or sample with the same traffic at the same time, but not perform more than one action simultaneously on the same packets.

```
[edit]
firewall {
    family inet {
        filter active_filter {
            term quarantined_traffic {
                from {
                    source-address {
                        10.36.1.2/32;
                    }
                }
                then {
                    count quarantined-counter;
                    sample;
                    discard accounting;
                }
            }
            term copy_and_forward_the_rest {
                then {
                    port-mirror;
                    accept;
                }
            }
        }
    }
}
```

Configuring the Interfaces That Will Be Actively Monitored

Configure the input interfaces and apply the firewall filter that you defined earlier. Unlike passive flow monitoring, the input interfaces for active flow monitoring are not restricted, so you can select most standard network interfaces (such as ATM1 or Ethernet-based interfaces) as the input.

If you configure active flow monitoring with sampling, you can configure an interface filter in place of a firewall filter with the sampling statement at the [edit interfaces *interface-name-fpc/pic/port* unit *unit-number* family inet] hierarchy level.

```
[edit]
interfaces {
   so-2/2/0 {
      unit 0 {
         family inet {
            filter {
               input active_filter;
            }
            address 10.36.11.2/32 {
               destination 10.36.11.1;
            }
            sampling {
               (input | output | [input output]);
            }
         }
      }
   }
}
```

Enabling the Monitoring Services, Adaptive Services, or Multiservices Interfaces and the Export Interface

You configure the monitoring services, adaptive services, or multiservices interfaces with the family inet statement so they can process IPv4 traffic. However, you must remember that a monitoring services interface uses an mo- prefix and adaptive services and multiservices interfaces use an sp- prefix.

```
[edit]
interfaces {
   sp-2/0/0 {
      unit 0 {
         family inet {
            address 10.36.100.1/32 {
               destination 10.36.100.2;
            }
         }
      }
   }
}
```

Active flow monitoring records leave the routing platform through an export interface to reach the flow monitoring server.

```
[edit]
interfaces {
   fe-1/0/0 {
      unit 0 {
         family inet {
            address 10.60.2.2/30;
         }
      }
   }
}
```

Collecting Flow Records

Traffic flows can be exported in flow monitoring version 5, 8, and 9 formats for active flow monitoring. The default export format for flow monitoring records is version 5. To change the export format to flow monitoring version 8, include the **version 8** statement at either the [**edit forwarding-options accounting** *name* **output flow-server** *flow-server-address*] or the [**edit forwarding-options sampling output flow-server** *flow-server-address*] hierarchy level. To change the export format to flow monitoring version 9, include the **version9 template** *template-name* statement at the [**edit forwarding-options sampling output flow-server** *flow-server-address*] hierarchy level. For more information on flow record formats, see "Flow Monitoring Output Formats" on page 377.

To capture flow data generated by the Monitoring Services PIC, Adaptive Services PIC, or MultiServices PIC and export it to a flow server, you can use one of the following active flow monitoring methods:

- Collecting Flow Records with a Sampling Group on page 351
- Collecting Flow Records with an Accounting Group on page 353
- Replicating Routing Engine-Based Sampling to Multiple Flow Servers on page 353
- Collecting Flow Records with a Template on page 354
- Routing Engine-Based Sampling to Multiple Flow Servers on page 356
- Replicating Version 9 Flow Aggregation to Multiple Flow Servers on page 356
- Option: Configuring an Aggregate Export Timer on page 357

Collecting Flow Records with a Sampling Group

If your needs for active flow monitoring are simple, you can collect flow records with a sampling group. Sampling does not require you to configure a monitoring group (as required in passive flow monitoring) because you can configure flow server information in the **sampling** hierarchy. When you wish to sample traffic, include the **sampling** statement at the [**edit forwarding-options**] hierarchy level.

The typical sampling configuration has one input interface and one export interface. The input interface is activated by the **then sample** statement in a firewall filter term.

This match condition directs traffic to the sampling process. Alternatively, you can use an interface-based filter in place of a firewall filter if you include the sampling statement at the [edit interfaces *interface-name-fpc/pic/port* unit *unit-number* family inet] hierarchy level.

There are two types of sampling available: PIC-based sampling and Routing Engine-based sampling. PIC-based sampling occurs when a monitoring services or adaptive services interface is the target for the output of the sampling process. To enable PIC-based sampling, include the interface statement at the [edit forwarding-options sampling output] hierarchy level and specify a monitoring services or adaptive services interface as the output interface. If an output interface is not specified in the sampling configuration, sampling is performed by the Routing Engine.

To specify a flow server in a sampling configuration, include the flow-server statement at the [edit forwarding-options sampling output] hierarchy level. You must specify the IP address, port number, and flow monitoring version of the destination flow server. Routing Engine-based sampling supports flow aggregation of up to eight flow servers (version 5 servers and version 8 only) at a time. The export packets are replicated to all flow servers configured to receive them. In contrast, PIC-based sampling allows you to specify just one version 5 flow server and one version 8 server simultaneously. Flow servers operating simultaneously must have different IP addresses.

As part of the output interface statements, you must configure a source address. In contrast, the interface-level statements of engine-id and engine-type are both added automatically. However, you can override these values with manually configured statements to track different flows with a single flow collector, as needed. When you configure sampling, SNMP input and output interface index information is captured in flow records by default.

```
[edit]
forwarding-options {
   sampling {
      input {
         family inet {
            rate 1;
         }
      }
      output {
         flow-server 10.60.2.1 {
            port 2055;
            version 5;
         }
         flow-inactive-timeout 15;
         flow-active-timeout 60;
         interface sp-2/0/0 {
            engine-id 5;
            engine-type 55;
            source-address 10.60.2.2;
         }
      }
   }
}
```

Collecting Flow Records with an Accounting Group

To perform discard accounting on specified traffic, you can collect flow records with the accounting statement at the [edit forwarding-options] hierarchy level. Like sampling, your topology must be simple (for example, one input interface and one export interface).

Again, you can collect flow records by specifying input and output interfaces. You can configure the input interface to perform discard accounting by applying a firewall filter that contains the then discard accounting statement. This match condition directs the filtered traffic to be converted into flow records and exported for analysis by the monitoring services or adaptive services interface. The original packets are then sent to the discard process. For the output, remember to specify the IP address and port of your flow server and the services interface you plan to use for processing flow records.

You must configure a source address, but the engine-id and engine-type output interface statements are added automatically. You can override these values manually to track different flows with a single flow collector. SNMP input and output interface index information is captured in flow records by default when you configure discard accounting.

```
[edit]
forwarding-options {
    accounting counter1 {
        output {
            flow-inactive-timeout 65;
            flow-active-timeout 65;
            flow-server 10.60.2.1 {
                port 2055;
                version 8;
                aggregation {
                    protocol-port;
                    source-destination-prefix;
                }
            }
            interface sp-2/0/0 {
                engine-id 1;
                engine-type 11;
                source-address 10.60.2.2;
            }
        }
    }
}
```

Replicating Routing Engine-Based Sampling to Multiple Flow Servers

Routing Engine-based sampling supports up to eight flow servers for both flow monitoring version 5 and version 8 configurations. The total number of flow servers is limited to eight, regardless of how many are configured for version 5 or version 8.

When you configure version 5 or version 8 sampling, the export packets are replicated to all flow servers configured to receive them. If two flow servers are configured to receive version 5 records, both flow servers will receive records for a specified flow.

NOTE: With Routing-Engine-based sampling, if multiple flow servers are configured with version 8 export format, all of them must use the same aggregation type (for example, all flow servers receiving version 8 export could be configured for source-destination aggregation type).

The following configuration example allows replication of export packets to two flow servers.

```
[edit]
forwarding-options {
    sampling {
        input {
            family inet {
                rate 1;
            }
        }
        output {
            flow-server 10.10.3.2 {
                port 2055;
                version 5;
                source-address 192.168.164.119;
            }
            flow-server 172.17.20.62 {
                port 2055;
                version 5;
                source-address 192.168.164.119;
            }
        }
    }
}
```

Collecting Flow Records with a Template

Flow monitoring version 9, which is based upon RFC 3954, provides a way to organize flow data into templates. Version 9 also provides a way to actively monitor MPLS traffic in addition to IPv4 and IPv6 traffic. Version 9 is not supported on the AS-I PIC.

To activate templates in flow monitoring, you must configure a template and include that template in the version 9 flow monitoring configuration. Version 9 does not work in conjunction with versions 5 and 8.

To configure a version 9 template, include the template *template-name* statement at the [edit services flow-monitoring version9] hierarchy level. The JUNOS software supports three different templates: ipv4-template, ipv6–template, mpls-template, and mpls-ipv4-template. To view the fields selected in each of these templates, see “Version 9 Formats and Fields” on page 387.

```
[edit]
```

```
services flow-monitoring {
   version9 { # Specifies flow monitoring version 9.
      template mpls { # Specifies template you are configuring.
         template-refresh-rate {
            packets 6000; # The default is 4800 packets and the range is 1–480000
            # packets.
         seconds 90; # The default is 60 seconds and the range is 1–600 seconds.
            option–refresh-rate {
            packets 3000; # The default is 4800 packets and the range is 1–480000
               # packets.
            seconds 30; # The default is 60 seconds and the range is 1–600.
            flow-active-timeout 60; # The default is 60 seconds and the range is
               # 10–600.
            flow-inactive-timeout 30; # The default is 60 seconds and the range 10–600.
            template-refresh-rate seconds 10; # The default is 60 seconds and the
               # range is 10–600
            option-refresh-rate seconds 10; # The default is 60 seconds and the range
               # is 10–600 seconds.
            mpls-template {
               label-positions [1 | 2 | 3]; # Specifies label position for the MPLS template.
            }
         }
      }
   }
}
```

You can export to multiple templates at a time to a maximum of eight flow servers for AS PICs and one flow server for all other PICs. To assign a template to a flow output, include the template *template-name* statement at the [edit forwarding options sampling output flow-server version9] hierarchy level:

```
[edit]
forwarding-options {
   sampling {
      input {
         family mpls {
            rate 1;
            run-length 1;
         }
      }
      output {
         flow-server 10.60.2.1 { # The IP address and port of the flow server.
            port 2055;
            source-address 3.3.3.1;
            version9 { # Records are sent to the flow server using version 9 format.
               template { # Indicates a template will organize records.
                  mpls; # Records are sent to the MPLS template.
               }
            }
         }
      }
   }
}
```

Routing Engine-Based Sampling to Multiple Flow Servers

Routing Engine-based sampling supports up to eight flow servers for both version 5 and version 8 configurations. The total number of collectors is limited to eight, regardless of how many are configured for version 5 or version 8. When you configure sampling, the export packets are replicated to all collectors configured to receive them. If two collectors are configured to receive version 5 records, both collectors will receive records for a specified flow.

The following configuration example allows replication of export packets to two collectors.

```
forwarding-options {
   sampling {
      input {
         family inet {
            rate 1;
         }
      }
      output {
         cflowd 10.10.3.2 {
            port 2055;
            version 5;
            source-address 192.168.164.119;
         }
         cflowd 172.17.20.62 {
            port 2055;
            version 5;
            source-address 192.168.164.119;
         }
      }
   }
}
```

Replicating Version 9 Flow Aggregation to Multiple Flow Servers

With this feature, you can configure up to eight flow servers to receive packets for a version 9 flow monitoring template. Once a flow server is configured to receive this data, it will also receive the following periodic version 9 flow monitoring updates:

- Options data
- Template definition

With RE-based sampling, if multiple collectors are configured with version 8 export format, all of them must use the same aggregation-type

The option and template definition refresh period is configured on a per-template basis at the [edit services flow-monitoring] hierarchy level.

The following configuration example allows replication of version 9 export packets to two flow servers.

```
forwarding-options {
   sampling {
      input {
         family inet {
            rate 1;
         }
      }
      output {
         flow-server 10.10.3.2 {
            port 2055;
            version9 {
               template {
                  ipv4;
               }
            }
         }
         flow-server 172.17.20.62 {
            port 2055;
            version9 {
               template {
                  ipv4;
               }
            }
         }
         flow-inactive-timeout 30;
         flow-active-timeout 60;
         interface sp-4/0/0 {
            source-address 10.10.3.4;
         }
      }
   }
}
```

Option: Configuring an Aggregate Export Timer

When you use flow monitoring version 8 records for active flow monitoring, you can configure an aggregate export timer. To configure this timer, include the aggregate-export-interval statement at the [edit forwarding-options sampling output] hierarchy level. The timer value has a default minimum setting of 90 seconds and a maximum value of 1800 seconds.

```
[edit]
forwarding-options {
   sampling {
      output {
         aggregate-export-interval duration;
      }
   }
}
```

Option: Configuring Port Mirroring

You can copy packets and reroute them to another interface by using port mirroring. To send packet copies to an interface, include the interface statement at the [edit

forwarding-options port-mirroring family *family-name* output] hierarchy level and specify the interface to receive the traffic.

You can even send port-mirrored traffic to a monitoring services or adaptive services interface. If you choose this option, accepted traffic is copied and the packet copies are sent to the services interface for flow processing.

To configure how often packets are copied from the monitored traffic, include the rate statement at the [edit forwarding-options port-mirroring family *family-name* input] hierarchy level. A rate of 1 port-mirrors every packet, while a rate of 10 port-mirrors every tenth packet.

```
[edit]
forwarding-options {
   port-mirroring {
      family (inet | inet6) {
         input {
            rate 1;
         }
         output {
            interface sp-2/0/0.0;
         }
      }
   }
}
```

Option: Configuring Port Mirroring with Filter-Based Forwarding and a Monitoring Group

For active flow monitoring, you can load-balance traffic across multiple Monitoring Services PICs using the same method as passive flow monitoring. The only difference is that you do not configure the input interface with the passive-monitor-mode statement at the [edit interfaces *interface-name*] hierarchy level.

To load-balance traffic for active flow monitoring, port-mirror the incoming packets to a tunnel services interface. Redirect this copy of the traffic to a filter-based forwarding instance by applying a firewall filter to the tunnel services interface. Configure the instance to send the traffic to a group of monitoring services interfaces. Finally, use a monitoring group to send flow records from the monitoring services interfaces to a flow server.

NOTE: When you load-balance port-mirrored traffic across several Monitoring Services interfaces, there are some limitations:

- The original Monitoring Services PIC supports this method. You cannot use a Monitoring Services II PIC.
- You must use the suite of show passive-monitoring commands to monitor traffic. The show services accounting commands are not supported.
- Because load-balanced traffic is routed through the Tunnel Services PIC, the total throughput of the load-balanced traffic coming from the Monitoring Services PICs cannot exceed the bandwidth of the tunnel interface.

For detailed information on this method, see "Copying and Redirecting Traffic with Port Mirroring and Filter-Based Forwarding" on page 301.

Option: Sending Traffic to Multiple Export Interfaces by Using Next-Hop Groups

To send port-mirrored traffic to multiple flow servers or packet analyzers, you can use the next-hop-group statement. The routing platform can make up to 16 copies of traffic per group and send the traffic to the next-hop group members you configure. A maximum of 30 groups can be configured on a routing platform at any given time. The port-mirrored traffic can be sent to any interface, except aggregated SONET/SDH, aggregated Ethernet, loopback (lo0), or administrative (fxp0) interfaces. To configure multiple port mirroring with next-hop groups, include the next-hop-group statement at the [edit forwarding-options] hierarchy level.

You must port-mirror the initial traffic to a tunnel interface so that it can be filtered and duplicated. Also, you need configure only the interface names for point-to-point interfaces, but you must configure the interface names and a next hop for multipoint interfaces (such as Ethernet).

```
[edit]
forwarding-options {
    port-mirroring {
        family inet {
            input {
                rate 1;
            }
            output {
                interface vt-3/3/0.1;
                no-filter-check;
            }
        }
    }
    next-hop-group ftp-traffic {
        interface so-4/3/0.0;
        interface so-0/3/0.0;
    }
    next-hop-group http-traffic {
        interface ge-1/1/0.0 {
            next-hop 10.12.1.2;
```

```
        }
        interface ge-1/2/0.0 {
            next-hop 10.13.1.2;
        }
    }
    next-hop-group default-collect {
        interface so-7/0/0.0;
        interface so-7/0/1.0;
    }
}
```

NOTE: Next-hop groups are supported on M-series routers only, except the M120 router and the M320 router.

Option: Using the Flow-Tap Application to Send Packets to a Mediation Device

Dynamic flow capture enables you to capture passively monitored packet flows on the basis of dynamic filtering criteria, using Dynamic Tasking Control Protocol (DTCP) requests. The flow-tap application extends the use of DTCP to intercept IPv4 packets in an active flow monitoring station and send a copy of packets that match filter criteria to one or more content destinations. Flow-tap data can be used for lawful intercept purposes and provides flexible trend analysis for detection of new security threats. The flow-tap application is supported on M-series and T-series routing platforms, except M160 routers and TX Matrix platforms.

NOTE: For information about dynamic flow capture, see “Using a Dynamic Flow Capture Interface to Monitor Traffic On Demand” on page 312. For information about DTCP, see Internet draft draft-cavuto-dtcp-01.txt at http://www.ietf.org/internet-drafts.

For detailed information about the flow-tap application, see the following sections:

Flow-Tap Architecture

The flow-tap architecture consists of one or more *mediation devices* that send requests to a Juniper Networks routing platform to monitor incoming data. Any packets that match specific filter criteria are forwarded to a set of one or more *content destinations*:

- Mediation device—A client that monitors electronic data or voice transfer over the network. The mediation device sends filter requests to the Juniper Networks routing platform using the DTCP. The clients are not identified for security reasons, but have permissions defined by a set of special login classes.
- Monitoring platform—A Juniper Networks M-series or T-series routing platform containing one or more Adaptive Services (AS) PICs, which are configured to support the flow-tap application. The monitoring platform processes the requests from the mediation devices, applies the dynamic filters, monitors incoming data flows, and sends the matched packets to the appropriate content destinations.
- Content destination—Recipient of the matched packets from the monitoring platform. Typically the matched packets are sent using an IP Security (IPSec) tunnel from the monitoring platform to another router connected to the content destination. The content destination and the mediation device can be physically located on the same host.
- Dynamic filters—The Packet Forwarding Engine automatically generates a firewall filter that is applied to all IPv4 routing instances. Each term in the filter includes a flow-tap action that is similar to the existing sample or port-mirroring actions. As long as one of the filter terms matches an incoming packet, the router copies the packet and forwards it to the AS PIC that is configured for flow-tap service. The AS PIC runs the packet through the client filters and sends a copy to each matching content destination. For security, filters installed by one client are not visible to others and the CLI configuration does not reveal the identity of the monitored target.

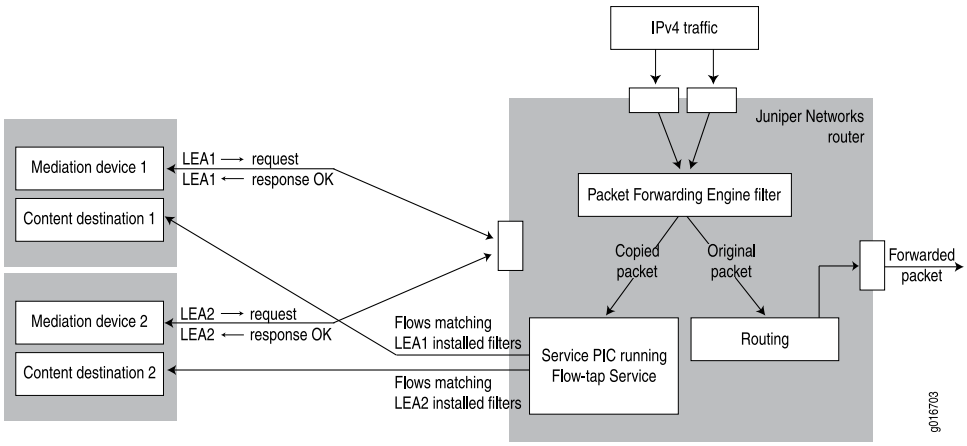
  Following is a sample filter configuration; note that it is dynamically generated by the router (no user configuration is required):

```
filter combined_LEA_filter {
    term LEA1_filter {
        from {
            source-address 1.2.3.4;
            destination-address 3.4.5.6;
        }
        then {
            flow-tap;
        }
    }
    term LEA2_filter {
        from {
            source-address 10.1.1.1;
            source-port 23;
        }
        then {
            flow-tap;
        }
```

```
    }
}
```

Figure 27 on page 362 shows a sample topology that uses two mediation devices and two content destinations.

Figure 27: Flow-Tap Topology Diagram



Configuring the Flow-Tap Interface

To configure an AS PIC interface for the flow-tap service, include the interface statement at the [edit services flow-tap] hierarchy level:

```
interface sp-fpc/pic/port.unit-number;
```

You can assign any AS PIC in the active monitoring station for flow-tap service, and use any logical unit on the PIC.

NOTE: You cannot configure dynamic flow capture and flow-tap features on the same router simultaneously.

You must also configure the logical interface at the [edit interfaces] hierarchy level:

```
interface sp-fpc/pic/port {
    unit logical-unit-number {
        family inet;
    }
}
```

Configuring Flow-Tap Security Properties

You can add an extra level of security to DTCP transactions between the mediation device and the router by enabling DTCP sessions on top of the SSH layer. To configure, include the flow-tap-dtcp statement at the [edit system services] hierarchy level:

```
flow-tap-dtcp {
    ssh {
        connection-limit value;
        rate-limit value;
    }
}
```

To configure client permissions for viewing and modifying flow-tap configurations and for receiving tapped traffic, include the **permissions** statement at the [edit system login class *class-name*] hierarchy level:

```
permissions [ permissions ];
```

The permissions needed to use flow-tap features are as follows:

- **flow-tap**—Can view flow-tap configuration.
- **flow-tap-control**—Can modify flow-tap configuration.
- **flow-tap-operation**—Can tap flows.

You can also specify user permissions on a RADIUS server, for example:

```
Bob Auth-Type := Local, User-Password = = “abc123”
Juniper-User-Permissions = “flow-tap-operation”
```

For details on [edit system] and RADIUS configuration, see the *JUNOS System Basics Configuration Guide*.

Flow-Tap Application Restrictions

The following restrictions apply to flow-tap services:

- You cannot configure dynamic flow capture and flow-tap services on the same router simultaneously.
- When the dynamic flow capture process or an AS PIC configured for flow-tap processing restarts, all filters are deleted and the mediation devices are disconnected.
- Only the first fragment of an IPv4 fragmented packet stream is sent to the content destination.
- Port mirroring might not work in conjunction with flow-tap processing.
- If the flow-tap application is configured, you cannot configure the filter action **then syslog** for any firewall filter running on the same platform.

Example: Flow-Tap Configuration

The following example shows all the parts of a complete flow-tap configuration.

```
services {
    flow-tap {
        interface sp-1/2/0.100;
```

```
    }
  }
  interfaces {
    sp-1/2/0 {
      unit 100 {
        family inet;
      }
    }
  }
  system {
    services {
      flow-tap-dtcp {
        ssh {
          connection-limit 5;
          rate-limit 5;
        }
      }
    }
    login {
      class ft-class {
        permissions flow-tap-operation;
      }
      user ft-user1 {
        class ft-class;
        authentication {
          encrypted-password “xxxx”;
        }
      }
    }
  }
```

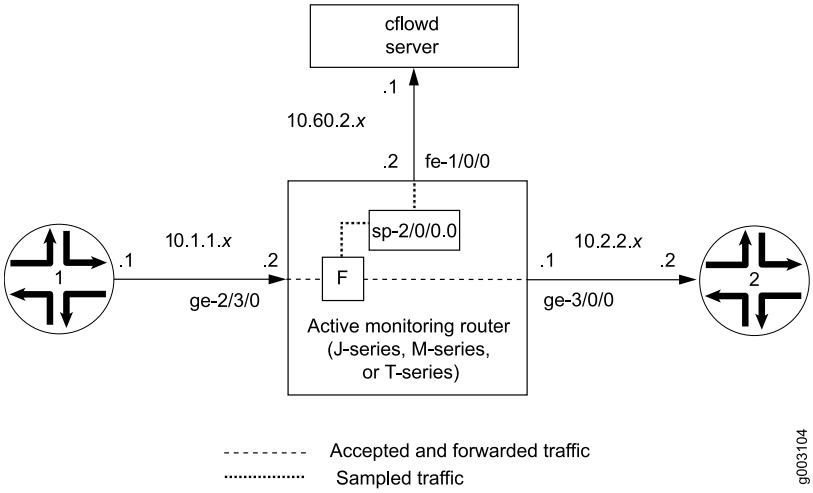
Active Flow Monitoring Configuration Examples

This section contains configuration examples and commands you can issue to verify an active flow monitoring configuration:

- Example: Sampling Configuration on page 365
- Example: Sampling and Discard Accounting Configuration on page 368
- Example: Multiple Port Mirroring with Next-Hop Groups Configuration on page 373

Example: Sampling Configuration

Figure 28: Active Flow Monitoring—Sampling Configuration Topology Diagram



In Figure 28 on page 365, traffic from Router 1 arrives on the monitoring router's Gigabit Ethernet ge-2/3/0 interface. The exit interface on the monitoring router that leads to destination Router 2 is ge-3/0/0. In active flow monitoring, both the input interface and exit interface can be any interface type (such as SONET/SDH, Gigabit Ethernet, and so on). The export interface leading to the flow server is fe-1/0/0.

Configure a firewall filter to sample, count, and accept all traffic. Apply the filter to the input interface, and configure the exit interface (for traffic forwarding), the adaptive services interface (for flow processing), and the export interface (for exporting flow records).

Configure sampling at the [edit forwarding-options] hierarchy level. Include the IP address and port of the flow server with the flow-server statement and specify the adaptive services interface to be used for flow record processing with the interface statement at the [edit forwarding-options sampling] hierarchy level.

Router 1

```
[edit]
interfaces {
   sp-2/0/0 { # This adaptive services interface creates the flow records.
      unit 0 {
         family inet {
            address 10.5.5.1/32 {
               destination 10.5.5.2;
            }
         }
      }
   }
   fe-1/0/0 { # This is the interface where records are sent to the flow server.
      unit 0 {
         family inet {
            address 10.60.2.2/30;
```

```
            }
        }
    }
    ge-2/3/0 { # This is the input interface where all traffic enters the router.
        unit 0 {
            family inet {
                filter {
                    input catch_all; # This is where the firewall filter is applied.
                }
                address 10.1.1.1/20;
            }
        }
    }
    ge-3/0/0 { # This is the interface where the original traffic is forwarded.
        unit 0 {
            family inet {
                address 10.2.2.1/24;
            }
        }
    }
}
forwarding-options {
    sampling { # Traffic is sampled and sent to a flow server.
        input {
            family inet {
                rate 1; # Samples 1 out of  x  packets (here, a rate of 1 sample per packet).
            }
        }
        output {
            flow-server 10.60.2.1 { # The IP address and port of the flow server.
                port 2055;
                version 5; # Records are sent to the flow server using version 5 format.
            }
            flow-inactive-timeout 15;
            flow-active-timeout 60;
            interface sp-2/0/0 { # Adding an interface here enables PIC-based sampling.
                engine-id 5; # Engine statements are dynamic, but can be configured.
                engine-type 55;
                source-address 10.60.2.2; # You must configure this statement.
            }
        }
    }
}
firewall {
    family inet {
        filter catch_all { # Apply this filter on the input interface.
            term default {
                then {
                    sample;
                    count counter1;
                    accept;
                }
            }
        }
    }
}
```

Verifying Your Work

To verify that your configuration is correct, use the following commands on the monitoring station that is configured for active flow monitoring:

- show services accounting errors
- show services accounting (flow | flow-detail)
- show services accounting memory
- show services accounting packet-size-distribution
- show services accounting status
- show services accounting usage
- show services accounting aggregation template template-name *name* (detail | extensive | terse) (version 9 only)

Most active flow monitoring operational mode commands contain equivalent output information to the following passive flow monitoring commands:

- show services accounting errors = show passive-monitoring error
- show services accounting flow = show passive-monitoring flow
- show services accounting memory = show passive-monitoring memory
- show services accounting status = show passive-monitoring status
- show services accounting usage = show passive-monitoring usage

The active flow monitoring commands can be used with most active flow monitoring applications, including sampling, discard accounting, port mirroring, and multiple port mirroring. However, you can use the passive flow monitoring commands only with configurations that contain a monitoring group at the [edit forwarding-options monitoring] hierarchy level.

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

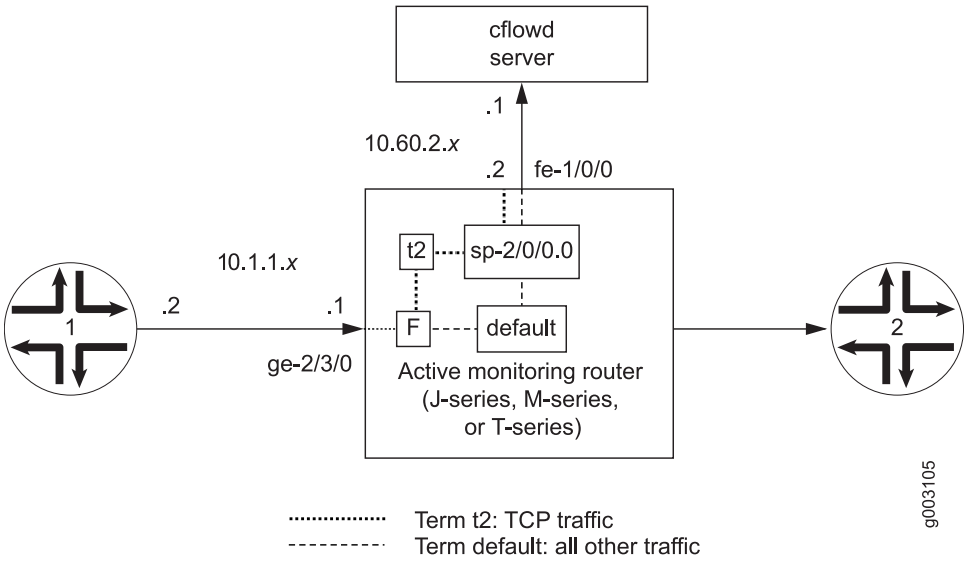
```
user@router> show services accounting errors
Service Accounting interface: sp-2/0/0, Local interface index: 542
Service name: (default sampling)
  Error information
    Packets dropped (no memory): 0, Packets dropped (not IP): 0
    Packets dropped (not IPv4): 0, Packets dropped (header too small): 0
    Memory allocation failures: 0, Memory free failures: 0
    Memory free list failures: 0
    Memory overload: No, PPS overload: No, BPS overload: Yes

user@router> show services accounting flow-detail limit 10
Service Accounting interface: sp-2/0/0, Local interface index: 468
Service name: (default sampling)
Protocol   Source         Source  Destination Destination    Packet        Byte
           Address          Port  Address            Port     count       count
udp(17)    10.1.1.2           53  10.0.0.1             53      4329     3386035
```

```
ip(0)      10.1.1.2          0  10.0.0.2          0      4785     3719654
ip(0)      10.1.1.2          0  10.0.1.2          0      4530     3518769
udp(17)    10.1.1.2          0  10.0.7.1          0      5011     3916767
tcp(6)     10.1.1.2         20  10.3.0.1         20         1        1494
tcp(6)     10.1.1.2         20  10.168.80.1      20         1         677
tcp(6)     10.1.1.2         20  10.69.192.1      20         1         446
tcp(6)     10.1.1.2         20  10.239.240.1     20         1        1426
tcp(6)     10.1.1.2         20  10.126.160.1     20         1         889
tcp(6)     10.1.1.2         20  10.71.224.1      20         1        1046

user@router>  show services accounting memory
Service Accounting interface: sp-2/0/0, Local interface index: 468
Service name: (default sampling)
  Memory utilization
    Allocation count: 437340, Free count: 430681, Maximum allocated: 6782
    Allocations per second: 3366, Frees per second: 6412
    Total memory used (in bytes): 133416928, Total memory free (in bytes):
133961744

user@router>  show services accounting packet-size-distribution
Service Accounting interface: sp-2/0/0, Local interface index: 468
Service name: (default sampling)
Range start     Range end     Number of packets     Percentage packets
         64            96               1705156                    100

user@router>  show services accounting status
Service Accounting interface: sp-2/0/0, Local interface index: 468
Service name: (default sampling)
  Interface state: Monitoring
  Group index: 0
  Export interval: 60 secs, Export format: cflowd v5
  Protocol: IPv4, Engine type: 55, Engine ID: 5
  Route record count: 13, IFL to SNMP index count: 30, AS count: 1
  Time set: Yes, Configuration set: Yes
  Route record set: Yes, IFL SNMP map set: Yes

user@router>  show services accounting usage
Service Accounting interface: sp-2/0/0, Local interface index: 468
Service name: (default sampling)
  CPU utilization
    Uptime: 4790345 milliseconds, Interrupt time: 1668537848 microseconds
    Load (5 second): 71%, Load (1 minute): 63%
```

Example: Sampling and Discard Accounting Configuration

Discard accounting allows you to sample traffic, send it to a flow server for analysis, and discard all packets without forwarding them to their intended destination. Discard accounting is enabled with the discard accounting *group-name* statement in a firewall filter at the [edit firewall family inet filter *filter-name* term *term-name* then] hierarchy level. Then, the filter is applied to an interface with the filter statement at the [edit interfaces *interface-name* unit *unit-number* family inet] hierarchy level and processed with the output statement at the [edit forwarding-options accounting *group-name*] hierarchy level.

Figure 29: Active Flow Monitoring—Sampling and Discard Accounting Topology Diagram



In Figure 29 on page 369, traffic from Router 1 arrives on the monitoring router's Gigabit Ethernet ge-2/3/0 interface. The export interface leading to the flow server is fe-1/0/0 and there is no exit interface.

In this example, TCP traffic is sent to one accounting group and all other traffic is diverted to a second group. After being sampled and counted, the two types of traffic are acted upon by the sampling and accounting processes. These processes create flow records and send the records to the version 8 flow server for analysis. Because multiple types of traffic are sent to the same server, we recommend that you configure the engine-id, engine-type, and source-address statements manually in your accounting and sampling hierarchies. This way, you can differentiate between traffic types when they arrive at the flow server.

```
[edit]
interfaces {
   sp-2/0/0 { # This adaptive services interface creates the flow records.
      unit 0 {
         family inet {
            address 10.5.5.1/32 {
               destination 10.5.5.2;
            }
         }
      }
   }
   fe-1/0/0 { # This is the interface where records are sent to the flow server.
      unit 0 {
         family inet {
            address 10.60.2.2/30;
         }
      }
   }
   ge-2/3/0 { # This is the input interface where traffic enters the router.
```

```
        unit 0 {
            family inet {
                filter {
                    input catch_all;
                }
                address 10.1.1.1/20;
            }
        }
    }
}
forwarding-options {
    sampling { # The router samples the traffic.
        input {
            family inet {
                rate 100; # One out of every 100 packets is sampled.
            }
        }
        output { # The sampling process creates and exports flow records.
            flow-server 10.60.2.1 { # You can configure a variety of settings.
                port 2055;
                version 8;
                aggregation { # Aggregation is unique to flow version 8.
                    protocol-port;
                    source-destination-prefix;
                }
            }
            aggregate-export-interval 90;
            flow-inactive-timeout 60;
            flow-active-timeout 60;
            interface sp-2/0/0 { # This statement enables PIC-based sampling.
                engine-id 5; # Engine statements are dynamic, but can be configured.
                engine-type 55;
                source-address 10.60.2.2; # You must configure this statement.
            }
        }
    }
    accounting counter1 { # This discard accounting process handles default traffic.
        output { # This process creates and exports flow records.
            flow-inactive-timeout 65;
            flow-active-timeout 65;
            flow-server 10.60.2.1 { # You can configure a variety of settings.
                port 2055;
                version 8;
                aggregation { # Aggregation is unique to version 8.
                    protocol-port;
                    source-destination-prefix;
                }
            }
            interface sp-2/0/0 {  # This statement enables PIC-based discard accounting.
                engine-id 1; # Engine statements are dynamic, but can be configured.
                engine-type 11;
                source-address 10.60.2.3; # You must configure this statement.
            }
        }
    }
    accounting t2 { # The second discard accounting process handles the TCP traffic.
```

```
        output { # This process creates and exports flow records.
            aggregate-export-interval 90;
            flow-inactive-timeout 65;
            flow-active-timeout 65;
            flow-server 10.60.2.1 { # You can configure a variety of settings for the server.
                port 2055;
                version 8;
                aggregation { # Aggregation is unique to version 8.
                    protocol-port;
                    source-destination-prefix;
                }
            }
            interface sp-2/0/0 { # This statement enables PIC-based discard accounting.
                engine-id 2; # Engine statements are dynamic, but can be configured.
                engine-type 22;
                source-address 10.60.2.4;# You must configure this statement.
            }
        }
    }
}
firewall {
    family inet {
        filter catch_all { # Apply the firewall filter on the input interface.
            term t2 { # This places TCP traffic into one group for sampling and
                from { # discard accounting.
                    protocol tcp;
                }
                then {
                    count c2;# The count action counts traffic as it enters the router.
                    sample; # The sample action sends the traffic to the sampling process.
                    discard accounting t2; # The discard accounting discards traffic.
                }
            }
            term default { # Performs sampling and discard accounting on all other traffic.
                then {
                    count counter; # The count action counts traffic as it enters the router.
                    sample# The sample action sends the traffic to the sampling process.
                    discard accounting counter1; # This activates discard accounting.
                }
            }
        }
    }
}
```

Verifying Your Work

To verify that your configuration is correct, use the following commands on the monitoring station that is configured for active flow monitoring:

- show services accounting aggregation (for version 8 flows only)
- show services accounting errors
- show services accounting (flow | flow-detail)
- show services accounting memory

- show services accounting packet-size-distribution
- show services accounting status
- show services accounting usage

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

```
user@router> show services accounting flow name t2
Service Accounting interface: sp-2/0/0, Local interface index: 468
Service name: t2
  Flow information
    Flow packets: 56130820, Flow bytes: 3592372480
    Flow packets 10-second rate: 13024, Flow bytes 10-second rate: 833573
    Active flows: 600, Total flows: 600
    Flows exported: 28848, Flows packets exported: 960
    Flows inactive timed out: 0, Flows active timed out: 35400

user@router>  show services accounting
Service Name:
    (default sampling)
    counter1
    t2

user@router>  show services accounting aggregation protocol-port detail name t2
Service Accounting interface: sp-2/0/0, Local interface index: 468
Service name: t2

  Protocol: 6, Source port: 20, Destination port: 20
  Start time: 442794, End time: 6436260
  Flow count: 1, Packet count: 4294693925, Byte count: 4277471552

user@router>  show services accounting aggregation source-destination-prefix name

t2 limit 10 order packets
Service Accounting interface: sp-2/0/0, Local interface index: 542
Service name: t2
     Source     Destination  Input SNMP  Output SNMP   Flow  Packet    Byte
     Prefix          Prefix       Index        Index  count   count   count
10.1.1.2/20    10.225.0.1/0          24           26      0      13    9650
10.1.1.2/20   10.143.80.1/0          24           26      0      13   10061
10.1.1.2/20   10.59.176.1/0          24           26      0      13   10426
10.1.1.2/20     10.5.32.1/0          24           26      0      13   12225
10.1.1.2/20    10.36.16.1/0          24           26      0      13    9116
10.1.1.2/20     10.1.96.1/0          24           26      0      12   11050
10.1.1.2/20    10.14.48.1/0          24           26      0      13   10812
10.1.1.2/20   10.31.192.1/0          24           26      0      13   11473
10.1.1.2/20  10.129.144.1/0          24           26      0      13    7647
10.1.1.2/20  10.188.160.1/0          24           26      0      13   10056

user@router> show services accounting aggregation source-destination-prefix name
t2 extensive limit 3
Service Accounting interface: sp-2/0/0, Local interface index: 542
Service name: t2

  Source address: 10.1.1.2, Source prefix length: 20
  Destination address: 10.200.176.1, Destination prefix length: 0
  Input SNMP interface index: 24, Output SNMP interface index: 26
  Source-AS: 69, Destination-AS: 69
```

```
Start time: Fri Feb 21 14:16:57 2003, End time: Fri Feb 21 14:22:50 2003
Flow count: 0, Packet count: 6, Byte count: 5340

Source address: 10.1.1.2, Source prefix length: 20
Destination address: 10.243.160.1, Destination prefix length: 0
Input SNMP interface index: 24, Output SNMP interface index: 26
Source-AS: 69, Destination-AS: 69
Start time: Fri Feb 21 14:16:57 2003, End time: Fri Feb 21 14:22:50 2003
Flow count: 0, Packet count: 6, Byte count: 5490

Source address: 10.1.1.2, Source prefix length: 20
Destination address: 10.162.160.1, Destination prefix length: 0
Input SNMP interface index: 24, Output SNMP interface index: 26
Source-AS: 69, Destination-AS: 69
Start time: Fri Feb 21 14:16:57 2003, End time: Fri Feb 21 14:22:50 2003
Flow count: 0, Packet count: 6, Byte count: 4079
```

Example: Multiple Port Mirroring with Next-Hop Groups Configuration

When you need to analyze traffic containing more than one packet type, or you wish to perform multiple types of analysis on a single type of traffic, you can implement multiple port mirroring and next-hop groups. You can make up to 16 copies of traffic per group and send the traffic to next-hop group members. A maximum of 30 groups can be configured on a routing platform at any given time. The port-mirrored traffic can be sent to any interface, except aggregated SONET/SDH, aggregated Ethernet, loopback (lo0), or administrative (fxp0) interfaces. To send port-mirrored traffic to multiple flow servers or packet analyzers, you can use the next-hop-group statement at the [edit forwarding-options] hierarchy level.

Figure 30: Active Flow Monitoring—Multiple Port Mirroring with Next-Hop Groups Topology Diagram

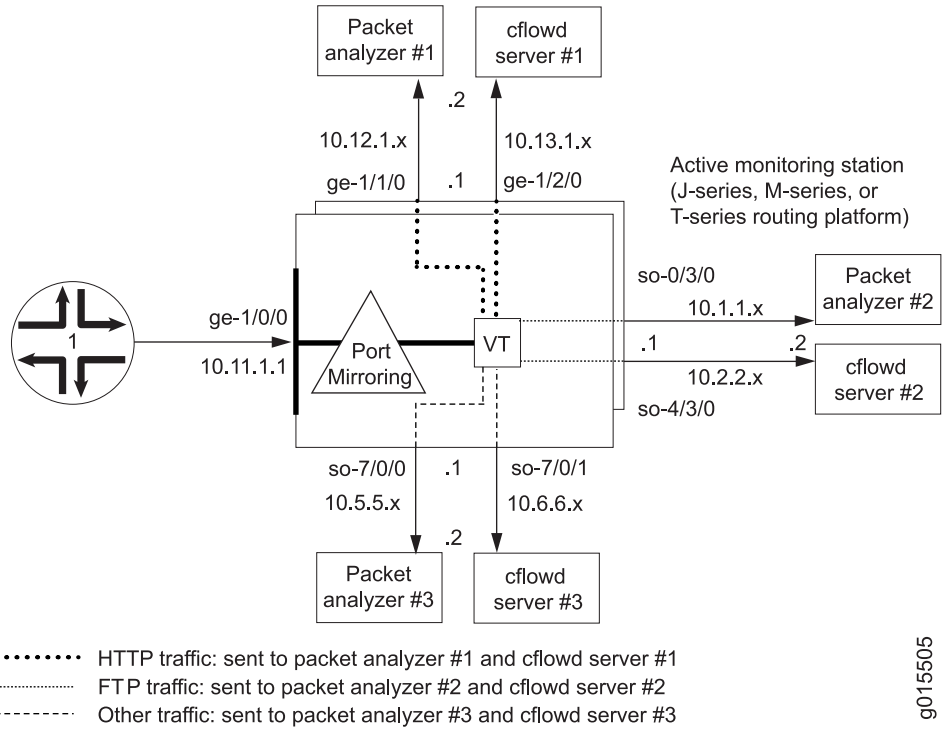


Figure 30 on page 374 shows an example of how to configure multiple port mirroring with next-hop groups. All traffic enters the monitoring router at interface ge-1/0/0. A firewall filter counts and port-mirrors all incoming packets to a Tunnel Services PIC. A second filter is applied to the tunnel interface and splits the traffic into three categories: HTTP traffic, FTP traffic, and all other traffic. The three types of traffic are assigned to three separate next-hop groups. Each next-hop group contains a unique pair of exit interfaces that lead to different groups of packet analyzers and flow servers.

```
[edit]
interfaces {
   ge-1/0/0 { # This is the input interface where packets enter the router.
      unit 0 {
         family inet {
            filter {
               input mirror_pkts; # Here is where you apply the first filter.
            }
            address 10.11.1.1/24;
         }
      }
   }
   ge-1/1/0 { # This is an exit interface for HTTP packets.
      unit 0 {
         family inet {
            address 10.12.1.1/24;
         }
      }
```

```
    }
    ge-1/2/0 { # This is an exit interface for HTTP packets.
      unit 0 {
        family inet {
          address 10.13.1.1/24;
        }
      }
    }
    so-0/3/0 { # This is an exit interface for FTP packets.
      unit 0 {
        family inet {
          address 10.1.1.1/30;
        }
      }
    }
    so-4/3/0 { # This is an exit interface for FTP packets.
      unit 0 {
        family inet {
          address 10.2.2.1/30;
        }
      }
    }
    so-7/0/0 { # This is an exit interface for all remaining packets.
      unit 0 {
        family inet {
          address 10.5.5.1/30;
        }
      }
    }
    so-7/0/1 { # This is an exit interface for all remaining packets.
      unit 0 {
        family inet {
          address 10.6.6.1/30;
        }
      }
    }
    vt-3/3/0 { # The tunnel interface is where you send the port-mirrored traffic.
      unit 0 {
        family inet;
      }
      unit 1 {
        family inet {
          filter {
            input collect_pkts;  # This is where you apply the second firewall filter.
          }
        }
      }
    }
}
forwarding-options {
    port-mirroring { # This is required when you configure next-hop groups.
      family inet {
        input {
          rate 1; # This port-mirrors all packets (one copy for every packet received).
        }
        output { # Sends traffic to a tunnel interface to enable multiport mirroring.
```

```
            interface vt-3/3/0.1;
            no-filter-check;
          }
        }
      }
      next-hop-group ftp-traffic { # Point-to-point interfaces require you to specify the
        interface so-4/3/0.0; # interface name.
        interface so-0/3/0.0;
      }
      next-hop-group http-traffic { # Configure a next hop for all multipoint interfaces.
        interface ge-1/1/0.0 {
          next-hop 10.12.1.2;
        }
        interface ge-1/2/0.0 {
          next-hop 10.13.1.2;
        }
      }
      next-hop-group default-collect {
        interface so-7/0/0.0;
        interface so-7/0/1.0;
      }
    }
    firewall {
      family inet {
        filter mirror_pkts { # Apply this filter to the input interface.
          term catch_all {
            then {
              count input_mirror_pkts;
              port-mirror; # This action sends traffic to be copied and port-mirrored.
            }
          }
        }
        filter collect_pkts { # Apply this filter to the tunnel interface.
          term ftp-term { # This term sends FTP traffic to an FTP next-hop group.
            from {
              protocol ftp;
            }
            then next-hop-group ftp-traffic;
          }
          term http-term { # This term sends HTTP traffic to an HTTP next-hop group.
            from {
              protocol http;
            }
            then next-hop-group http-traffic;
          }
          term default { # This sends all remaining traffic to a final next-hop group.
            then next-hop-group default-collectors;
          }
        }
      }
    }
```

Flow Monitoring Output Formats

When you implement passive flow monitoring and active flow monitoring, you should be familiar with flow monitoring formats and fields. Version 5 and version 8 export data into specified fields. Version 9 exports data into templates.

The flow monitoring station monitors the traffic flow and exports the data in flow format to an external server. The JUNOS software collects information about the following fields:

- Source and destination IP address
- Total number of bytes and packets sent
- Start and end times of the data flow
- Source and destination port numbers
- TCP flags
- IP protocol and IP type of service
- Originating AS of source and destination address
- Source and destination address prefix mask lengths
- Next-hop router's IP address
- MPLS label (version 9 only)
- ICMP (version 9 only)

Detailed descriptions of the formats are available as follows:

- Version 5 Formats and Fields on page 377
- Version 8 Formats and Fields on page 381
- Version 9 Formats and Fields on page 387

Version 5 Formats and Fields

A detailed explanation of version 5 packet formats and fields is shown in the following figures and tables:

- Figure 31 on page 378
- Table 24 on page 378
- Figure 32 on page 379
- Table 25 on page 379

Figure 31: Version 5 Packet Header Format

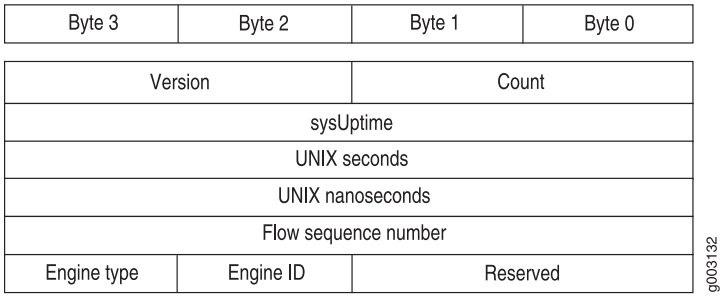


Table 24: Export Version 5 Packet Header Fields

Field	Description	Comments
Version	5	–
Count	The number of records in the Protocol Data Unit (PDU) or packet	–
sysUptime	Current time elapsed, in milliseconds, since the routing platform started	–
UNIX seconds	Current seconds since 0000 UTC 1970	NTP synchronized time; the clock on each services PIC is autonomous (200–400 msec jitter) across PICs in a chassis
UNIX nanoseconds	Residual nanoseconds since 0000 UTC 1970	See Comments above for UNIX seconds
Flow sequence number	Sequence number of total flows received	–
Engine type	User-configured 8-bit value	Also known as VIP type on other vendors' equipment
Engine ID	User-configured 8-bit value	–

Figure 32: Version 5 Flow-Export Flow Header Format

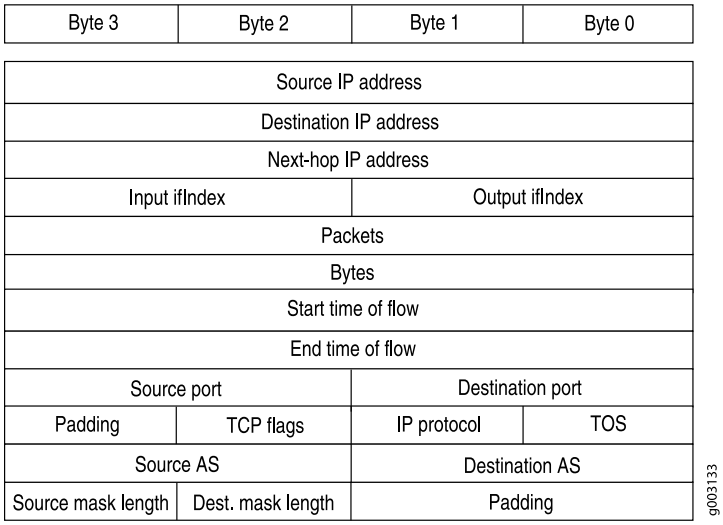


Table 25: Export Version 5 Flow-Export Flow Header Fields

Field	Description	Comments
Source IP address	Source IP address of the flow	–
Destination IP address	Destination IP address of the flow	–
Next-hop IP address	IP address of the routing platform where flows are forwarded	–
Input ifIndex	SNMP index value for the input interface where the routing platform receives flows	JUNOS Release 5.7 and later—Dynamically inserted, but overridden by manual configuration JUNOS Release 5.5—Manually set JUNOS Release 5.4—Set to zero
Output ifIndex	SNMP index value for the output interface where the routing platform forwards flows	JUNOS Release 5.7 and later—Dynamically inserted, but overridden by manual configuration JUNOS Release 5.5—Manually set JUNOS Release 5.4—Set to zero
Packets	Total number of packets received in a flow	–
Bytes	Total number of bytes received in a flow	–
Start time of flow	System up time, in seconds, at the start of the flow	System up time for the services PIC accepting flows
End time of flow	System up time, in seconds, at the end of the flow	System up time for the services PIC accepting flows

Table 25: Export Version 5 Flow-Export Flow Header Fields *(continued)*

Field	Description	Comments
Source port	Source application port	–
Destination port	Destination application port	The ICMP type is placed in the high-order byte and the ICMP type code is placed in the low-order byte of this field.
TCP flags	TCP flags set in the flow	–
IP protocol	IP protocol number	–
TOS	IP type of service	–
Source AS	AS number of the source address	JUNOS Release 5.7 and later—Dynamically inserted if AS information is available
Destination AS	AS number of the destination address	JUNOS Release 5.7 and later—Dynamically inserted if AS information is available
Source mask length	Source address network mask length	–
Dest. mask length	Destination address network mask length	–
Padding	Bytes available to ensure a minimum packet length	–

Useful formulas for flow monitoring are:

- start flow timestamp absolute = *unixTime* x 1000 – (*sysUptime* – start flow timestamp)
- end flow timestamp absolute = *unixTime* x 1000 – (*sysUptime* – end flow timestamp)

NOTE: In the 2-byte destination port field of the export version 5 flow-export flow format, the following information can be derived:

- High-order byte—ICMP type
- Low-order byte—ICMP type code

For example, if the ICMP type is 3 (00000011 in binary) and the ICMP type code is network unreachable (Type Code 0, or 00000000 in binary), the resulting destination port field value is 00000011 00000000 (768 in decimal).

For more information on ICMP type and type code, see RFC 792 at http://www.ietf.org.

Version 8 Formats and Fields

A detailed explanation of version 8 packet formats and fields is shown as follows:

Figure 33: Version 8 Template Flow Format

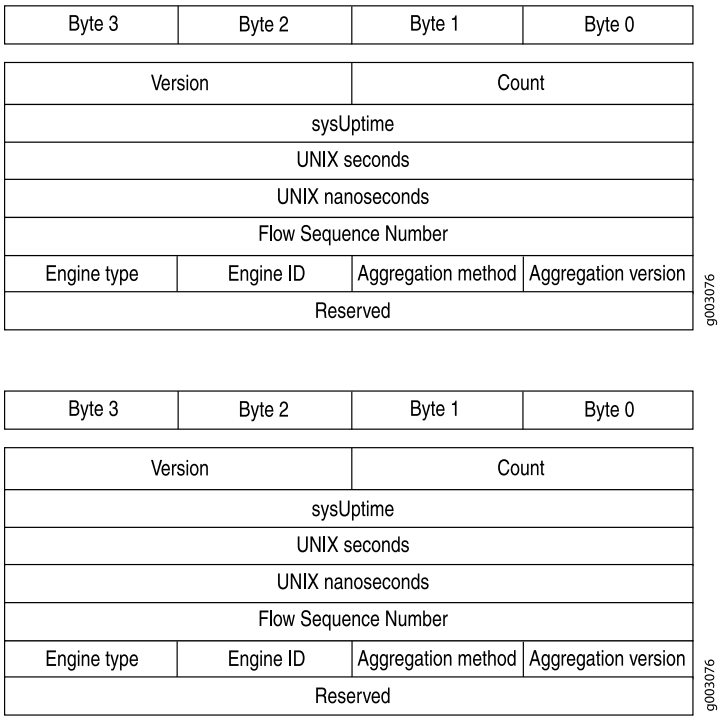


Table 26: Version 8 Flow Template Fields

Field	Description
Version	8
Count	The number of records in the protocol data unit (PDU) or packet
sysUptime	Current time elapsed, in milliseconds, since the routing platform started
UNIX seconds	Current seconds since 0000 UTC 1970
UNIX nanoseconds	Residual nanoseconds since 0000 UTC 1970
Flow sequence number	Sequence counter of total flows received
Engine type	Type of flow switching engine
Engine ID	ID number of the flow switching engine
Aggregation method	Aggregation method used
Aggregation version	Version of the aggregation export
Reserved	Empty field reserved for future usage

Figure 34: Version 8 AS Aggregation Flow Entry Format

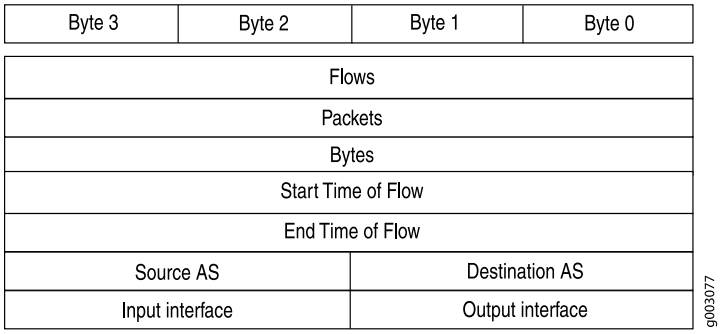


Table 27: Version 8 AS Aggregation Flow Entry Fields

Field	Description
Flows	Total number of flows
Packets	Total number of packets received in a flow
Bytes	Total number of bytes received in a flow
Start time of flow	System up time, in seconds, at the start of the flow
End time of flow	System up time, in seconds, at the end of the flow

Table 27: Version 8 AS Aggregation Flow Entry Fields *(continued)*

Field	Description
Source AS	AS number of the source address
Destination AS	AS number of the destination address
Input interface	SNMP index value for the input interface where the routing platform receives flows
Output interface	SNMP index value for the output interface where the routing platform forwards flows

Figure 35: Version 8 Protocol/Port Aggregation Flow Entry Format

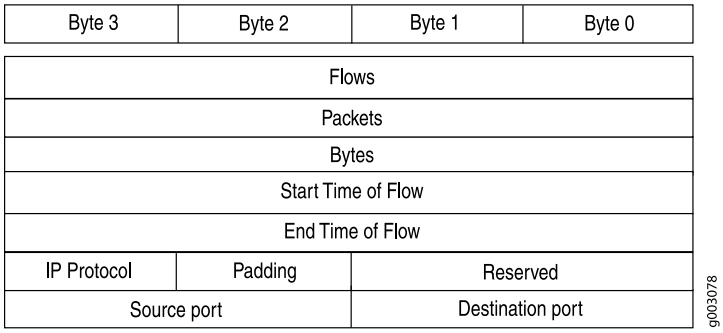


Table 28: Version 8 Protocol/Port Aggregation Flow Entry Fields

Field	Description
Flows	Total number of flows
Packets	Total number of packets received in a flow
Bytes	Total number of bytes received in a flow
Start time of flow	System up time, in seconds, at the start of the flow
End time of flow	System up time, in seconds, at the end of the flow
IP protocol	IP protocol number
Padding	Bytes available to ensure a minimum packet length
Reserved	Empty field reserved for future usage
Source port	Source application port
Destination port	Destination application port

Figure 36: Version 8 Prefix Aggregation Flow Entry Format

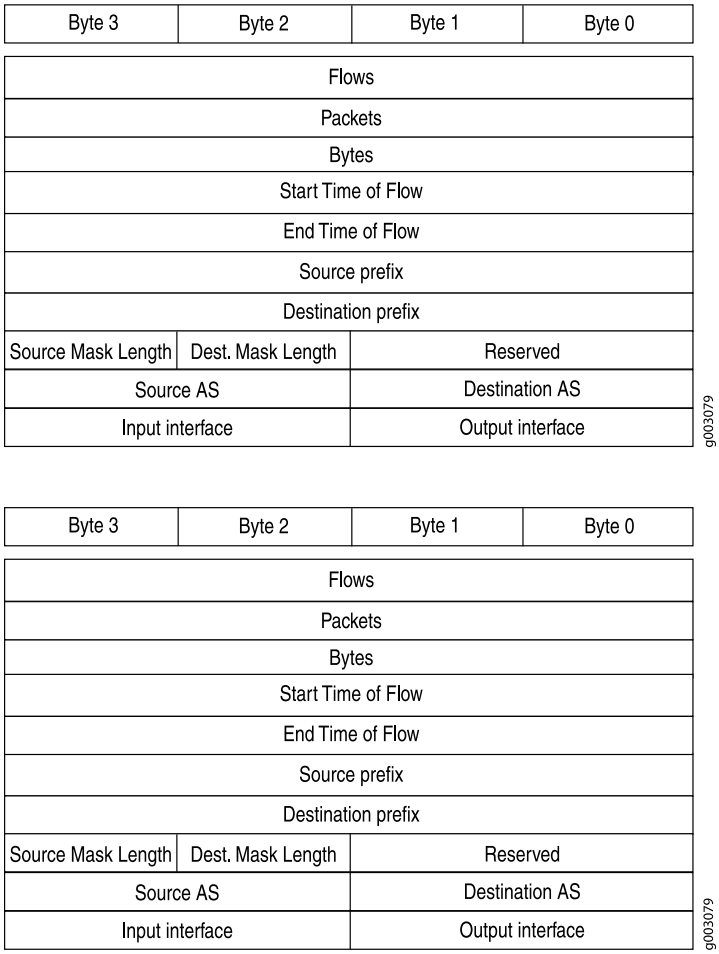


Table 29: Version 8 Prefix Aggregation Flow Entry Fields

Field	Description
Flows	Total number of flows
Packets	Total number of packets received in a flow
Bytes	Total number of bytes received in a flow
Start time of flow	System up time, in seconds, at the start of the flow
End time of flow	System up time, in seconds, at the end of the flow
Source prefix	Source IP address prefix
Destination prefix	Destination IP address prefix
Source mask length	Source address network mask length

Table 29: Version 8 Prefix Aggregation Flow Entry Fields *(continued)*

Field	Description
Dest. mask length	Destination address network mask length
Reserved	Empty field reserved for future usage
Source AS	AS number of the source address
Destination AS	AS number of the destination address
Input interface	SNMP index value for the input interface where the routing platform receives flows
Output interface	SNMP index value for the output interface where the routing platform forwards flows

Figure 37: Version 8 Source Prefix Aggregation Flow Entry Format

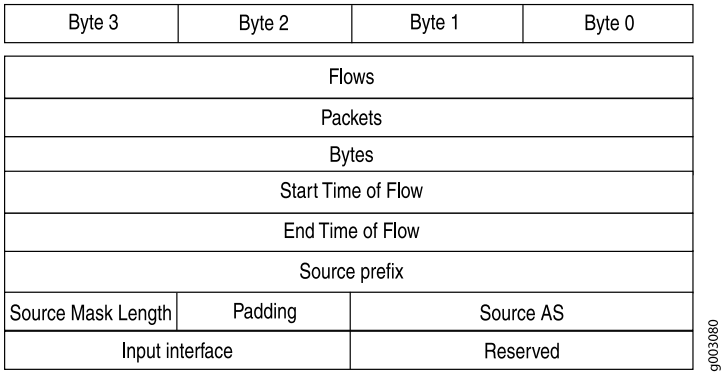


Table 30: Version 8 Source Prefix Aggregation Flow Entry Fields

Field	Description
Flows	Total number of flows
Packets	Total number of packets received in a flow
Bytes	Total number of bytes received in a flow
Start time of flow	System up time, in seconds, at the start of the flow
End time of flow	System up time, in seconds, at the end of the flow
Source prefix	Source IP address prefix
Source mask length	Source address network mask length
Padding	Bytes available to ensure a minimum packet length
Source AS	AS number of the source address

Table 30: Version 8 Source Prefix Aggregation Flow Entry Fields *(continued)*

Field	Description
Input interface	SNMP index value for the input interface where the routing platform receives flows
Reserved	Empty field reserved for future usage

Figure 38: Version 8 Destination Prefix Aggregation Flow Entry Format

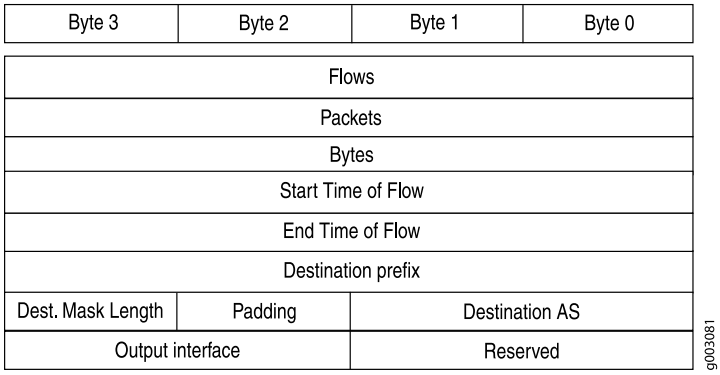


Table 31: Version 8 Destination Prefix Aggregation Flow Entry Fields

Field	Description
Flows	Total number of flows
Packets	Total number of packets received in a flow
Bytes	Total number of bytes received in a flow
Start time of flow	System up time, in seconds, at the start of the flow
End time of flow	System up time, in seconds, at the end of the flow
Destination prefix	Destination IP address prefix
Dest. mask length	Destination address network mask length
Padding	Bytes available to ensure a minimum packet length
Destination AS	AS number of the destination address
Output interface	SNMP index value for the output interface where the routing platform forwards flows
Reserved	Empty field reserved for future usage

For more information about version 5 and version 8 packet formats and fields, see http://www.caida.org.

Version 9 Formats and Fields

A detailed explanation of active flow monitoring version 9 packet formats and fields is shown as follows:

- Table 32 on page 387
- Figure 39 on page 389
- Table 33 on page 389
- Figure 41 on page 391
- Table 33 on page 389
- Figure 42 on page 392
- Table 37 on page 392
- Figure 43 on page 393
- Table 38 on page 393

The JUNOS software supports the following version 9 template formats:

Table 32: Flow Monitoring Version 9 Template Formats

Template	Fields
IPV4	Flow selectors: ■ Source and destination IP address ■ Source and destination address prefix mask lengths ■ Source and destination port numbers ■ IP protocol and IP type of service ■ ICMP type Flow nonselectors: ■ TCP flags ■ Input and output SNMP ■ Input bytes ■ Input packets ■ Start time ■ End time

Table 32: Flow Monitoring Version 9 Template Formats *(continued)*

Template	Fields
MPLS	Flow selectors: ■ MPLS label 1 ■ MPLS label 2 ■ MPLS label 3 Flow nonselectors: ■ Input and output SNMP ■ Input bytes ■ Input packets ■ Start time ■ End time
MPLS_IPV4	Flow selectors: ■ MPLS label 1 ■ MPLS label 2 ■ MPLS label 3 Flow nonselectors: ■ Input and output SNMP ■ Input bytes ■ Input packets ■ Start time ■ End time
IPv6	Flow selectors: ■ IP protocol and IP type of service ■ Source and destination port numbers ■ Input SNMP ■ Source and destination IPv6 address ■ ICMP type Flow nonselectors: ■ Input bytes ■ Input packets ■ TCP flags ■ Output SNMP ■ Source and destination autonomous system ■ Last and first switched ■ IPv6 source and destination mask ■ IP protocol version ■ IPv6 next hop

Figure 39: Version 9 Flow Header Format

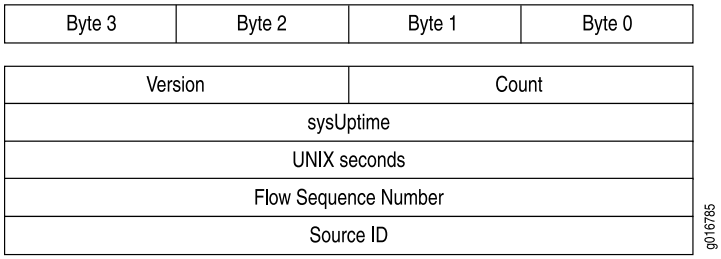


Table 33: Version 9 Flow Header Fields

Field	Description
Version	9
Count	Total number of records in the protocol data unit (PDU) or packet. This number includes all of the options FlowSet records, template FlowSet records, and data FlowSet records.
sysUptime	Current time elapsed, in milliseconds, since the routing platform started
UNIX seconds	Current seconds since 0000 UTC 1970
Flow sequence number	Sequence counter of total flows received
Source ID	32-bit value that identifies the data exporter. Version 9 uses the integrated field diagnostics (IFD) SNMP index of the PIC or device that is exporting the data flow. This field is equivalent to engine type and engine ID fields found in versions 5 and 8.

Figure 40: Version 9 Template FlowSet Format

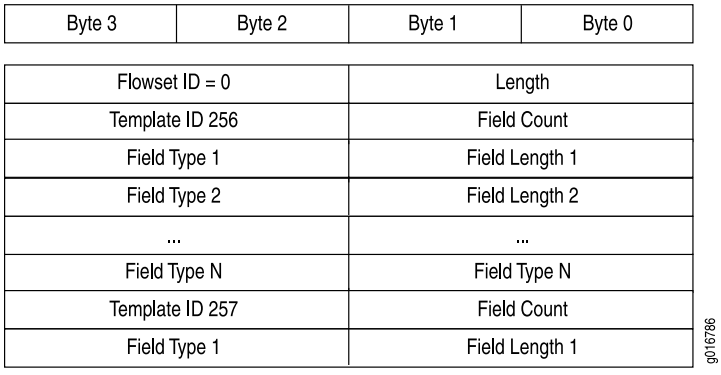


Table 34: Version 9 Template FlowSet Fields

Field	Description
FlowSet ID	FlowSet type. FlowSet ID 0 is reserved for the Template FlowSet.
Length	FlowSet length. Individual template FlowSets might contain multiple template records, which means that the length of template FlowSets varies.
Template ID	Unique template ID assigned to each newly generated template. Templates numbered 256 and higher define data formats. Templates numbered 0 through 255 define FlowSet IDs.
Field Count	Fields in the template record. This field allows the collector to determine the end of the current template record and the start of the next.
Field Type	Field type. These are defined in Table 35 on page 390.
Field Length	Length, in bytes, of the corresponding field type.

Table 35: Field Type Definitions Supported in the JUNOS Software

Field Type	Description
1	IN_BYTES: The number of bytes associated with an IP flow. By default, the length is 4 bytes.
2	IN_PKTS: The number of packets associated with an IP flow. By default, the length is 4 packets.
4	PROTOCOL: The IP protocol byte.
5	TOS: The type of service byte setting of an incoming packet.
6	TCP_FLAGS: The cumulative TCP flags associated with a flow.
7	L4_SRC_PORT: The TCP/UDP source port.
8	IPv4_SRC_ADDR: The IPv4 source address.
9	SRC_MASK: The number of contiguous bits in the source subnet mask.
10	INPUT_SNMP: The IFD SNMP input interface index. By default, the length is 2.
11	L4_DST_PORT: The TCP/UDP destination port number.
12	IPV4_DST_ADDR: The IPv4 destination address.
13	DST_MASK: The number of contiguous bits in the destination subnet mask.
14	OUTPUT_SNMP: The IFD SNMP output interface index. By default, the length is 2.

Table 35: Field Type Definitions Supported in the JUNOS Software *(continued)*

Field Type	Description
16	SRC_AS: The source autonomous system number. This is always set to zero
17	DST_AS: The destination autonomous system number. This is always set to zero
21	LAST_SWITCHED: The uptime of the device (in milliseconds) at which the last packet of the flow was switched.
22	FIRST_SWITCHED: The uptime of the device (in milliseconds) at which the first packet of the flow was switched.
29	IPV6_SRC_MASK: The length of the IPv6 source mask in contiguous bits.
30	IPV6_DST_MASK: The length of the IPv6 destination mask in contiguous bits.
32	ICMP_TYPE: The ICMP type.
34	SAMPLING_INTERVAL: The rate at which packets are sampled. As an example, a rate of 100 means that one packet is sampled for every 100 packets in the data flow.
35	SAMPLING_ALGORITHM: The type of algorithm being used. Ox01 indicates deterministic sampling and 0x02 indicates random sampling.
60	IP_PROTOCOL_VERSION: The IP protocol version being used.
62	IPV6_NEXT_HOP: The IPv6 address of the next-hop router.
70	MPLS_LABEL_1:The first MPLS label in the stack.
71	MPLS_LABEL_2: The second MPLS label in the stack.
72	MPLS_LABEL_3: The third MPLS label in the stack.

Figure 41: Version 9 Data FlowSet Format

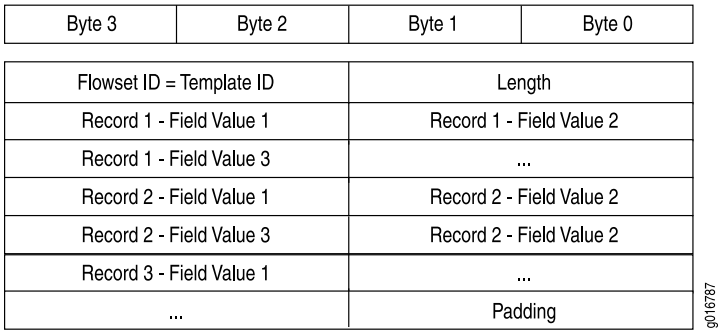


Table 36: Version 9 Data FlowSet Format

Field	Description
FlowSet ID = Template ID	Data FlowSet that associated with a FlowSet ID. The FlowSet ID maps to a previously generated template ID. The flow collector must use the FlowSet ID to find the corresponding template record and decode the flow records from the FlowSet.
Length	FlowSet length. Data FlowSets are fixed in length.
Record Number - Field Value Number	Flow data records, each containing a set of field values. The template record identified by the FlowSet ID dictates the type and length of the field values.
Padding	Bytes (in zeros) that the exporter inserts so that the subsequent FlowSet starts at a 4-byte aligned boundary.

Figure 42: Version 9 Options Template Format

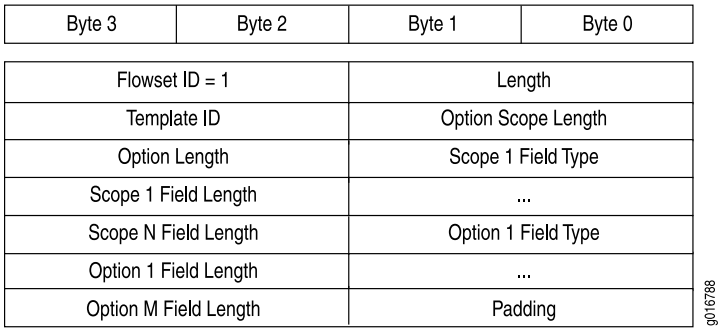


Table 37: Version 9 Options Template Format

Field	Description
FlowSet ID	FlowSet type. FlowSet ID 1 is reserved for the options template.
Length	FlowSet length. Option template FlowSets are fixed in length.
Template ID	Template ID of the options template. Options template values are greater than 255.
Option Scope Length	Length, in bytes, of any scope field definition that is part of the options template record.
Scope 1 Field Type	Relevant process. The JUNOS software supports the system process (1).
Scope 1 Field Length	Length, in bytes, of the option field.

Table 37: Version 9 Options Template Format *(continued)*

Field	Description
Padding	Bytes the exporter inserts so that the subsequent FlowSet starts at a 4-byte aligned boundary.

Figure 43: Active Flow Monitoring Version 9 Options Data Record Format

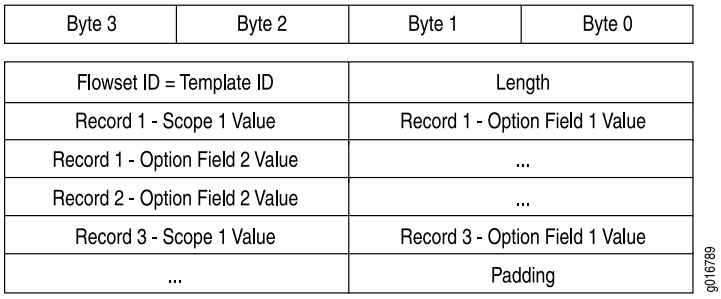


Table 38: Active Flow Monitoring Version 9 Options Data Record Format

Field	Description
FlowSet ID = Template ID	ID that precedes each options data flow record. The FlowSet ID maps to a previously generated template ID. The collector must use the FlowSet ID to find the corresponding template record and decode the options data flow records from the FlowSet.
Length	FlowSet length. Option FlowSets are fixed in length.
Number of Flow Data Records	Remainder of the options data FlowSet is a collection of flow data records, each containing a set of field values. The template record identified by the FlowSet ID dictates the type and length of the field values.
Padding	Bytes (in zeros) the exporter inserts so that the subsequent FlowSet starts at a 4-byte aligned boundary.

For More Information

To learn more about passive flow monitoring, active flow monitoring, cflowd versions 5 and 8, and flow monitoring version 9 see the following:

- Version 9: RFC 3954 at http://www.faqs.org/rfcs/rfc3954.html
- Versions 5 and 8: Cooperative Association for Internet Data Analysis (CAIDA) Web site at http://www.caida.org.
- *JUNOS Services Interfaces Configuration Guide*

- *JUNOS Policy Framework Configuration Guide*
- Internet draft draft-cavuto-dtcp-01.txt, *DTCP: Dynamic Tasking Control Protocol* (expires March 2007)

For more information on IPSec and the ES PIC, see the *JUNOS System Basics Configuration Guide*.

Revision History

13 April 2009—9.5R1 Release. Roy Spencer.

15 January 2009—9.4R1 Release. Added support for IPv6 sampling and templates for flow monitoring version 9. Fawn Damitio.

10 October 2008—9.3R1 Release. Roy Spencer.

10 July 2008—9.2R1 Release. Roy Spencer.

10 April 2008—9.1R1 Release. Roy Spencer.

1 February 2008—Version 9 flow aggregation to multiple flow servers. 9.0R1 Release. Fawn Damitio.

29 June 2007—Passive flow monitoring support on MultiServices 400 PIC (Type 2). 8.4R1 Release. Fawn Damitio.

27 March 2007—Added version 9 support for active flow monitoring. 8.3R1 Release. Fawn Damitio.

12 January 2007—Added active flow monitoring support for M120 routers. 8.2R1 Release. Fawn Damitio.

15 September 2006—Added support for the flow-tap services application, 8.1R1 Release. Richard Hendricks.

29 June 2006—8.0R1 Release. Richard Hendricks.

27 March 2006—Added support for the clear passive-monitoring statistics command, 7.6R1 Release. Richard Hendricks.

9 January 2006—Added support for the Dynamic Flow Capture MIB, 7.5R1 Release. Richard Hendricks.

14 September 2005—Added dynamic flow capture support on Monitoring Services III PICs installed in T-series and M320 routing platforms, and port mirroring for IPv6 packets, 7.4R1 Release. Richard Hendricks.

13 June 2005—Added passive flow monitoring support for selected Ethernet-based interfaces, active flow monitoring support for Adaptive Services II PICs installed in TX Matrix platforms, filter-based forwarding support for output interfaces, and a minor update to the request services flow-collector change-destination interface *interface-name* (primary | secondary) command. 7.3R1 Release. Richard Hendricks.

5 April 2005—7.2R1 Release. Richard Hendricks.

2 February 2005—Added support for passive monitoring and flow collection services on Monitoring Services II PICs installed in T-series and M320 routing platforms. Also, included information about the expanded set of flow collector name format macros, 7.1R1 Release. Richard Hendricks.

6 October 2004—Added support for active flow monitoring on Adaptive Services II PICs installed in T-series and M320 routing platforms, 7.0R1 Release. Richard Hendricks.

6 July 2004—Added support for the next-hop IP address field in version 5 records, 6.4R1 Release. Richard Hendricks.

5 April 2004—Added an explanation of how to load-balance traffic across multiple Monitoring Services I PICs for active flow monitoring and provided more information about flow collection services, 6.3R1 Release. Richard Hendricks.

21 January 2004—Added additional flow collector interface information. Richard Hendricks.

22 December 2003—Added passive flow monitoring support for ATM2 IQ interfaces, MPLS label removal, and flow collector interface configuration, 6.2R1 Release. Richard Hendricks.

22 September 2003—6.1R1 Release. Richard Hendricks.

30 June 2003—Added Monitoring Services II PIC, Adaptive Services PIC, and rearranged existing content, 6.0R1 Release. Richard Hendricks.

2 April 2003—Added new active flow monitoring content, 5.7R1 Release. Richard Hendricks.

27 December 2002—Revised the entire chapter for the 5.6R1 Release. Richard Hendricks.

22 October 2002—Added active flow monitoring section. Richard Hendricks.

30 September 2002—5.5R1 Release. Richard Hendricks.

27 August 2002—Added 5.5 show commands and expanded the packet, header, and field descriptions. Richard Hendricks.

19 July 2002—5.4R1 Release. Richard Hendricks.

28 June 2002—Reformatted the document, edited content, and added several new sections. Richard Hendricks.

6 May 2002—Initial document written. Renu Bhargava.

Chapter 11

IPSec

This feature guide covers these topics:

Overview

IP Security (IPSec) provides a secure way to authenticate senders and encrypt IP version 4 (IPv4) and version 6 (IPv6) traffic between network devices, such as routing platforms and hosts. IPSec offers network administrators and their users the benefits of data confidentiality, data integrity, sender authentication, and anti-replay services. IPSec is increasingly becoming a critical component in today's contemporary IP networks.

IPSec is a framework for ensuring secure private communication over IP networks and is based on standards developed by the International Engineering Task Force

(IETF). IPsec provides security services at the network layer of the Open Systems Interconnection (OSI) model by enabling a system to select required security protocols, determine the algorithms to use for the security services, and implement any cryptographic keys required to provide the requested services. You can use IPsec to protect one or more paths between a pair of hosts, between a pair of security gateways (such as routing platforms), or between a security gateway and a host.

The terminology and components of IPSec can be intimidating to first-time users. However, if you learn a few key concepts, you can quickly master and deploy IPSec in your network. The main concepts you need to understand are as follows:

IPSec-Enabled PICs

The first choice you need to make when implementing IPSec on a JUNOS software-based routing platform is the type of Physical Interface Card (PIC) you wish to use. There are three types of PICs available for M-series and T-series platforms:

- The ES PIC is a first-generation PIC that provides encryption services and software support for IPSec.
- The Adaptive Services (AS) PIC is a next-generation PIC that provides IPSec services and other services, such as Network Address Translation (NAT) and stateful firewall.
- The AS II Federal Information Processing Standards (FIPS) PIC is a special version of the AS PIC that communicates securely with the Routing Engine by using internal IPSec. You must configure IPSec on the AS II FIPS PIC when you enable FIPS mode on the routing platform. For more information about implementing IPSec on an AS II FIPS PIC installed in a routing platform configured in FIPS mode, see the *Secure Configuration Guide for Common Criteria and JUNOS-FIPS*.
- The MultiServices PICs supply hardware acceleration for an array of packet processing-intensive services in the M-series and T-series routers. These services include IPSec services and other services, such as stateful firewall, NAT, IPSec, anomaly detection, and tunnel services.

The J-series Services Routers also perform IPSec services in a manner similar to the AS and MultiServices PICs. However, the J-series Services Routers do this using the JUNOS software without a corresponding PIC. For more information about implementing IPSec on a J-series Services Router, see the *J-series Services Router Advanced WAN Access Configuration Guide*.

Authentication Algorithms

Authentication is the process of verifying the identity of the sender. Authentication algorithms use a shared key to verify the authenticity of the IPSec devices. The JUNOS software uses the following authentication algorithms:

- Message Digest 5 (MD5) uses a one-way hash function to convert a message of arbitrary length to a fixed-length message digest of 128 bits. Because of the conversion process, it is mathematically infeasible to calculate the original message by computing it backwards from the resulting message digest. Likewise, a change to a single character in the message will cause it to generate a very different message digest number.

  To verify that the message has not been tampered with, the JUNOS software compares the calculated message digest against a message digest that is decrypted with a shared key. The JUNOS software uses the MD5 hashed message authentication code (HMAC) variant that provides an additional level of hashing. MD5 can be used with authentication header (AH), Encapsulating Security Payload (ESP), and Internet Key Exchange (IKE).

- Secure Hash Algorithm 1 (SHA-1) uses a stronger algorithm than MD5. SHA-1 takes a message of less than 264 bits in length and produces a 160-bit message digest. The large message digest ensures that the data has not been changed and that it originates from the correct source. The JUNOS software uses the SHA-1 HMAC variant that provides an additional level of hashing. SHA-1 can be used with AH, ESP, and IKE.

- SHA-256, SHA-384, and SHA-512 (sometimes grouped under the name SHA-2) are variants of SHA-1 and use longer message digests. The JUNOS software supports the SHA-256 version of SHA-2, which can process all versions of Advanced Encryption Standard (AES), Data Encryption Standard (DES), and Triple DES (3DES) encryption.

Encryption Algorithms

Encryption encodes data into a secure format so that it cannot be deciphered by unauthorized users. Like authentication algorithms, a shared key is used with encryption algorithms to verify the authenticity of the IPSec devices. The JUNOS software uses the following encryption algorithms:

- Data Encryption Standard cipher-block chaining (DES-CBC) is a symmetric secret-key block algorithm. DES uses a key size of 64 bits, where 8 bits are used for error detection and the remaining 56 bits provide encryption. DES performs a series of simple logical operations on the shared key, including permutations and substitutions. CBC takes the first block of 64 bits of output from DES, combines this block with the second block, feeds this back into the DES algorithm, and repeats this process for all subsequent blocks.

- Triple DES-CBC (3DES-CBC) is an encryption algorithm that is similar to DES-CBC, but provides a much stronger encryption result because it uses three keys for 168-bit (3 x 56-bit) encryption. 3DES works by using the first key to encrypt the blocks, the second key to decrypt the blocks, and the third key to reencrypt the blocks.

- Advanced Encryption Standard (AES) is a next-generation encryption method based on the Rijndael algorithm developed by Belgian cryptographers Dr. Joan Daemen and Dr. Vincent Rijmen. It uses a 128-bit block and three different key sizes (128, 192, and 256 bits). Depending on the key size, the algorithm performs a series of computations (10, 12, or 14 rounds) that include byte substitution, column mixing, row shifting, and key addition. The use of AES in conjunction with IPSec is defined in RFC 3602, *The AES-CBC Cipher Algorithm and Its Use with IPsec*.

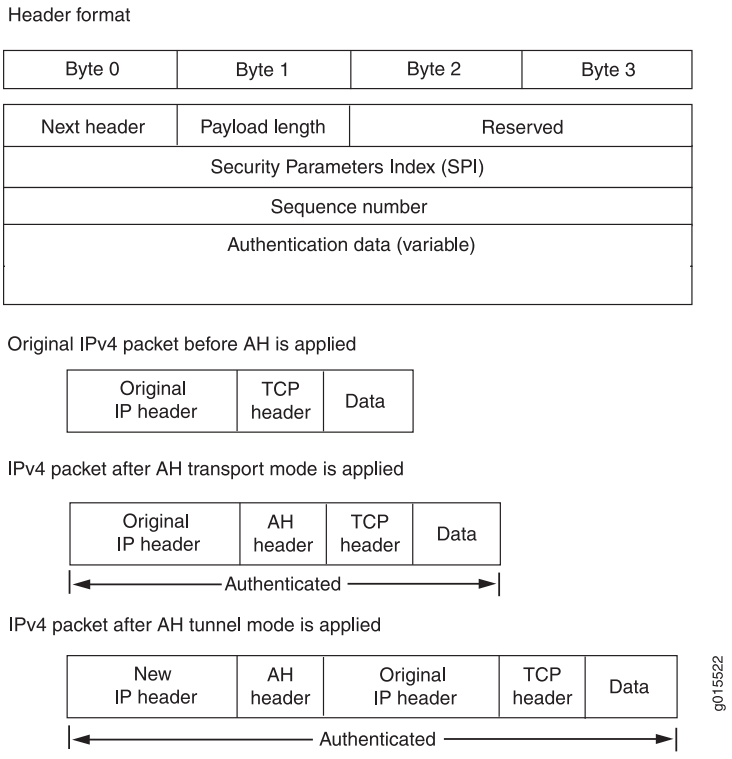
IPSec Protocols

IPSec protocols determine the type of authentication and encryption applied to packets that are secured by the routing platform. The JUNOS software supports the following IPSec protocols:

- AH—Defined in RFC 2402, AH provides connectionless integrity and data origin authentication for IPv4 and IPv6 packets. It also provides protection against replays. AH authenticates as much of the IP header as possible, as well as the upper-level protocol data. However, some IP header fields may change in transit. Because the value of these fields may not be predictable by the sender, they cannot be protected by AH. In an IP header, AH can be identified with a value of 51 in the Protocol field of an IPv4 packet and the Next Header field of an IPv6 packet. An example of the IPSec protection offered by AH is shown in Figure 44 on page 402.

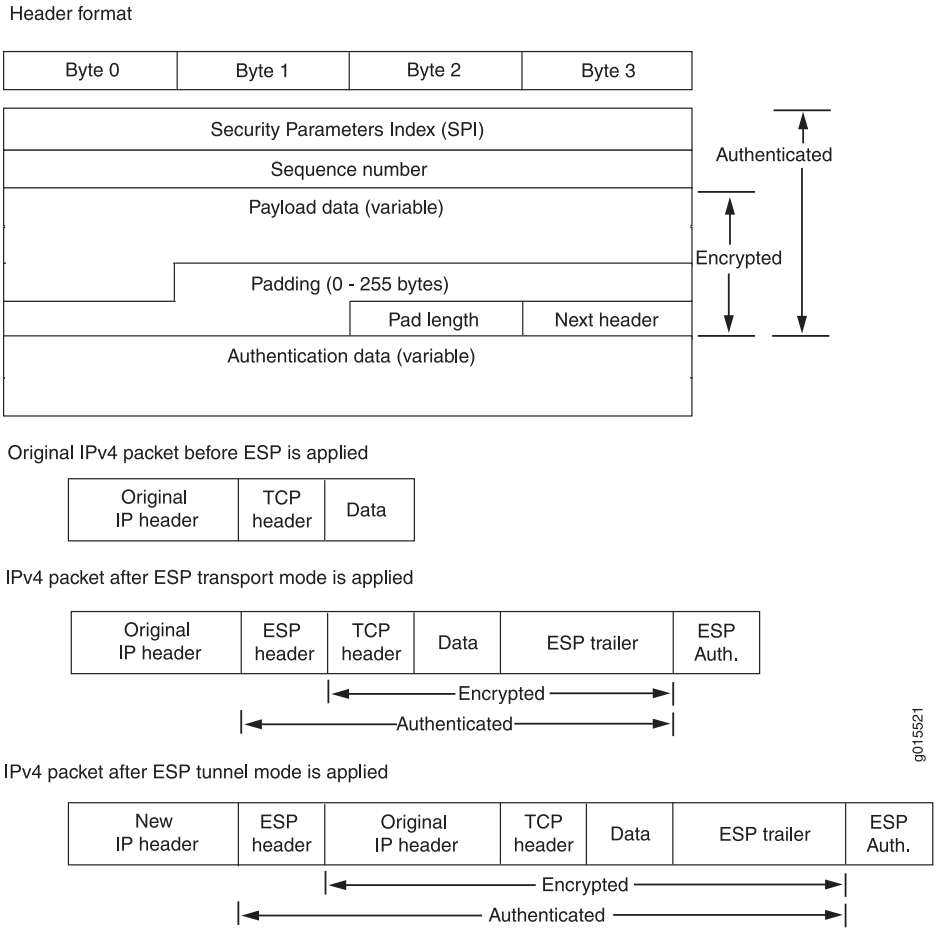
NOTE: AH is not supported on the T-series, M120, and M320 routing platforms.

Figure 44: AH Protocol



- ESP—Defined in RFC 2406, ESP can provide encryption and limited traffic flow confidentiality, or connectionless integrity, data origin authentication, and an anti-replay service. In an IP header, ESP can be identified a value of 50 in the Protocol field of an IPv4 packet and the Next Header field of an IPv6 packet. An example of the IPSec protection offered by ESP is shown in Figure 45 on page 403.

Figure 45: ESP Protocol



- Bundle—When you compare AH with ESP, there are some benefits and shortcomings in both protocols. ESP provides a decent level of authentication and encryption, but does so only for part of the IP packet. Conversely, although AH does not provide encryption, it does provide authentication for the entire IP packet. Because of this, the JUNOS software offers a third form of IPSec protocol called a protocol bundle. The bundle option offers a hybrid combination of AH authentication with ESP encryption.

Security Associations

Another IPSec consideration is the type of security association (SA) that you wish to implement. An SA is a set of IPSec specifications that are negotiated between devices that are establishing an IPSec relationship. These specifications include preferences for the type of authentication, encryption, and IPSec protocol that should be used when establishing the IPSec connection. An SA can be either unidirectional or bidirectional, depending on the choices made by the network administrator. An SA is uniquely identified by a Security Parameter Index (SPI), an IPv4 or IPv6 destination address, and a security protocol (AH or ESP) identifier.

You can configure IPSec with a preset, preshared manual SA or use IKE to establish a dynamic SA. Manual SAs require you to specify all the IPSec requirements up front. Conversely, IKE dynamic SAs typically contain configuration defaults for the highest levels of authentication and encryption.

IPSec Modes

The last major consideration is the type of IPSec mode you wish to implement in your network. The JUNOS software supports the following IPSec modes:

- Tunnel mode is supported for both AH and ESP in the JUNOS software and is the usual choice for a routing platform. In tunnel mode, the SA and associated protocols are applied to tunneled IPv4 or IPv6 packets. For a tunnel mode SA, an outer IP header specifies the IPsec processing destination, and an inner IP header specifies the ultimate destination for the packet. The security protocol header appears after the outer IP header, and before the inner IP header. In addition, there are slight differences for tunnel mode when you implement it with AH and ESP:
  - For AH, portions of the outer IP header are protected, as well as the entire tunneled IP packet.
  - For ESP, only the tunneled packet is protected, not the outer header.

  When one side of a security association is a security gateway (such as a routing platform), the SA must use tunnel mode. However, when traffic (for example, SNMP commands or BGP sessions) is destined for a routing platform, the system acts as a host. Transport mode is allowed in this case because the system does not act as a security gateway and does not send or receive transit traffic.
- Transport mode provides a security association between two hosts. In transport mode, the protocols provide protection primarily for upper layer protocols. For IPv4 and IPv6 packets, a transport mode security protocol header appears immediately after the IP header and any options, and before any higher layer

protocols (for example, TCP or UDP). There are slight differences for transport mode when you implement it with AH and ESP:

- For AH, selected portions of the IP header are protected, as well as selected portions of the extension headers and selected options within the IPv4 header.
- For ESP, only the higher layer protocols are protected, not the IP header or any extension headers preceding the ESP header.

Digital Certificates

For small networks, the use of preshared keys in an IPSec configuration is often sufficient. However, as a network grows, it can become a challenge to add new preshared keys on the local routing platform and all new and existing IPSec peers. One solution for scaling an IPSec network is to use digital certificates.

A digital certificate implementation uses the public key infrastructure (PKI), which requires you to generate a key pair consisting of a public key and a private key. The keys are created with a random number generator and are used to encrypt and decrypt data. In networks that do not use digital certificates, an IPSec-enabled device encrypts data with the private key and IPSec peers decrypt the data with the public key.

With digital certificates, the key sharing process requires an additional level of complexity. First, you and your IPSec peers request a certificate authority (CA) to send you a CA certificate that contains the public key of the CA. Next, you request the CA to enroll a local digital certificate that contains your public key and some additional information. When the CA processes your request, it signs your local certificate with the private key of the CA. Then you install the CA certificate and the local certificate in your local routing platform and load the CA certificate in the remote devices before you can establish IPSec tunnels with your peers.

When you request a peering relationship with an IPSec peer, the peer receives a copy of your local certificate. Because the peer already has the CA certificate loaded, it can use the CA's public key contained in the CA certificate to decrypt your local certificate that has been signed by the CA's private key. As a result, the peer now has a copy of your public key. The peer encrypts data with your public key before sending it to you. When your local routing platform receives the data, it decrypts the data with your private key.

In the JUNOS software, you must implement the following steps to be able to initially use digital certificates:

- Configure a CA profile to request CA and local digital certificates—The profile contains the name and URL of the CA or registration authority (RA), as well as some retry timer settings.
- Configure Certificate Revocation List support—A certificate revocation list (CRL) contains a list of certificates canceled before their expiration date. When a participating peer uses a CRL, the CA acquires the most recently issued CRL and checks the signature and validity of a peer's digital certificate. You can request and load CRLs manually, configure an LDAP server to handle CRL processing automatically, or disable CRL processing that is enabled by default.

- Request a digital certificate from the CA—The request can be made either online or manually. Online CA digital certificate requests use the Simple Certificate Enrollment Protocol (SCEP) format. If you request the CA certificate manually, you must also load the certificate manually.
- Generate a private/public key pair—The public key is included in the local digital certificate and the private key is used to decrypt data received from peers.
- Generate and enroll a local digital certificate—The local certificate can be processed online using SCEP or generated manually in the Public-Key Cryptography Standards #10 (PKCS-10) format. If you create the local certificate request manually, you must also load the certificate manually.
- Apply the digital certificate to an IPSec configuration—To activate a local digital certificate, you configure the IKE proposal to use digital certificates instead of preshared keys, reference the local certificate in the IKE policy, and identify the CA in the service set.

Optionally, you can do the following:

- Configure the digital certificate to automatically reenroll—Starting in JUNOS Release 8.5, you can configure automatic reenrollment for digital certificates.
- Monitor digital certificate events and delete certificates and requests—You can issue operational mode commands to monitor IPSec tunnels established using digital certificates and delete certificates or requests.

For more details on managing digital certificates, configuring them in an IPSec service set, and monitoring and clearing them, see “Option: Using Digital Certificates” on page 421 and “Example: AS PIC IKE Dynamic SA with Digital Certificates Configuration” on page 487.

Service Sets

The Adaptive Services PIC supports two types of service sets when you configure IPSec tunnels. Because they are used for different purposes, it is important to know the differences between these service set types.

- Next-hop service set—Supports multicast and multicast-style dynamic routing protocols (such as OSPF) over IPSec. Next-hop service sets allow you to use *inside* and *outside* logical interfaces on the Adaptive Services PIC to connect with multiple routing instances. They also allow the use of Network Address Translation (NAT) and stateful firewall capabilities. However, next-hop service sets do not monitor Routing Engine traffic by default and require configuration of multiple service sets to support traffic from multiple interfaces.
- Interface service set—Applied to a physical interface and similar to a stateless firewall filter. They are easy to configure, can support traffic from multiple interfaces, and can monitor Routing Engine traffic by default. However, they cannot support dynamic routing protocols or multicast traffic over the IPSec tunnel.

In general, we recommend that you use next-hop service sets because they support routing protocols and multicast over the IPSec tunnel, they are easier to understand, and the routing table makes forwarding decisions without administrative intervention.

System Requirements

To implement IPSec, your system must meet these minimum requirements:

- JUNOS Release 8.5 or later for automatic reenrollment of digital certificates.
- JUNOS Release 8.3 or later for IPSec support on OSPF version 2
- JUNOS Release 8.2 or later for support on M120 routers
- JUNOS Release 8.1 or later for IPSec IKE support in routing instances, and certificate revocation list support on J-Series Services Routers and AS and MultiServices PICs installed on M-series and T-series routing platforms
- JUNOS Release 7.6 or later for AES encryption and SHA-256 authentication support on J-series Services Routers and AS PICs installed in M-series routers, and IPv6-based IPSec for AS PICs installed in M-series and T-series routing platforms
- JUNOS Release 7.5 or later for digital certificate support on J-series Services Routers and AS PICs installed in M-series and T-series routing platforms, and support of the IPSec Monitoring Management Information Base (MIB)
- JUNOS Release 7.4 or later for dynamic endpoint tunneling support and configuring multiple routed tunnels in a single next-hop service set
- JUNOS Release 7.2 or later for transport mode IPSec on Routing Engines running OSPF version 3 and support for the AS II FIPS PIC
- JUNOS Release 7.1 or later for IPSec on the ES PIC for T-series and M320 routing platforms
- JUNOS Release 7.0 or later for IPSec on a J-series Services Router
- JUNOS Release 6.4 or later for IPSec on the AS PIC for T-series and M320 routing platforms
- JUNOS Release 6.2 or later for IPSec on the AS PIC for M-series routers
- JUNOS Release 5.7 or later for multicast over IPSec tunnels on M-series routers
- JUNOS Release 5.2 or later for IPSec on the ES PIC for M-series routers
- Two Juniper Networks J-series, M-series, or T-series routing platforms
- Two ES PICs or AS PICs for M-series and T-series routing platforms

Terms and Acronyms

A

Adaptive Services PIC A next-generation Physical Interface Card (PIC) that provides IPSec services and other services, such as Network Address Translation (NAT) and stateful firewall, on M-series and T-series platforms.

Advanced Encryption Standard (AES) A next-generation encryption method that is based on the Rijndael algorithm and uses a 128-bit block, three different key sizes (128, 192, and 256 bits), and multiple rounds of processing to encrypt data.

authentication header (AH) A component of the IPSec protocol used to verify that the contents of a packet have not changed (data integrity), and to validate the identity of the sender (data source authentication). For more information about AH, see RFC 2402.

C

certificate authority (CA) A trusted third-party organization that generates, enrolls, validates, and revokes digital certificates. The CA guarantees the identity of a user and issues public and private keys for message encryption and decryption.

certificate revocation list (CRL) A list of digital certificates that have been invalidated before their expiration date, including the reasons for their revocation and the names of the entities that have issued them. A CRL prevents usage of digital certificates and signatures that have been compromised.

cipher block chaining (CBC) A cryptographic method that encrypts blocks of ciphertext by using the encryption result of one block to encrypt the next block. Upon decryption, the validity of each block of ciphertext depends on the validity of all the preceding ciphertext blocks. For more information on how to use CBC with DES and ESP to provide confidentiality, see RFC 2405.

D

Data Encryption Standard (DES) An encryption algorithm that encrypts and decrypts packet data by processing the data with a single shared key. DES operates in increments of 64-bit blocks and provides 56-bit encryption.

digital certificate Electronic file that uses private and public key technology to verify the identity of a certificate creator and distribute keys to peers.

E

Encapsulating Security Payload (ESP) A component of the IPSec protocol used to encrypt data in an IPv4 or IPv6 packet, provide data integrity, and ensure data source authentication. For more information about ESP, see RFC 2406.

ES PIC A PIC that provides first-generation encryption services and software support for IPSec on M-series and T-series platforms.

H

Hashed Message Authentication Code (HMAC) A mechanism for message authentication using cryptographic hash functions. HMAC can be used with any iterative cryptographic hash function, such as MD5 or SHA-1, in combination with a secret shared key. For more information on HMAC, see RFC 2104.

I

Internet Key Exchange (IKE) Establishes shared security parameters for any hosts or routers using IPSec. IKE establishes the SAs for IPSec. For more information about IKE, see RFC 2407.

M

Message Digest 5 (MD5) An authentication algorithm that takes a data message of arbitrary length and produces a 128-bit message digest. For more information, see RFC 1321.

P

Perfect Forward Secrecy (PFS) Provides additional security by means of a Diffie-Hellman shared secret value. With PFS, if one key is compromised, previous and subsequent keys are secure because they are not derived from previous keys.

public key infrastructure (PKI) A trust hierarchy that enables users of a public network to securely and privately exchange data through the use of public and private cryptographic key pairs that are obtained and shared with peers through a trusted authority.

R

registration authority (RA) A trusted third-party organization that acts on behalf of a CA to guarantee the identity of a user.

Routing Engine A PCI-based architectural portion of a JUNOS-based routing platform that handles the routing protocol process, the interface process, some of the chassis components, system management, and user access.

S

Secure Hash Algorithm 1 (SHA-1) An authentication algorithm that takes a data message of less than 264 bits in length and produces a 160-bit message digest. For more information on SHA-1, see RFC 3174.

Secure Hash Algorithm 2 (SHA-2) A successor to the SHA-1 authentication algorithm that includes a group of SHA-1 variants (SHA-224, SHA-256, SHA-384, and SHA-512). SHA-2 algorithms use larger hash sizes and are designed to work with enhanced encryption algorithms such as AES.

security association (SA) Specifications that must be agreed upon between two network devices before IKE or IPSec are allowed to function. SAs primarily specify protocol, authentication, and encryption options.

Security Association Database (SADB) A database where all SAs are stored, monitored, and processed by IPSec.

Security Parameter Index (SPI) An identifier that is used to uniquely identify an SA at a network host or routing platform.

Security Policy Database (SPD) A database that works with the SADB to ensure maximum packet security. For inbound packets, IPSec checks the SPD to verify if the incoming packet matches the security configured for a particular policy. For outbound packets, IPSec checks the SPD to see if the packet needs to be secured.

Simple Certificate Enrollment Protocol (SCEP) A protocol that supports CA and registration authority (RA) public key distribution, certificate enrollment, certificate revocation, certificate queries, and certificate revocation list (CRL) queries.

T

Triple Data Encryption Standard (3DES) An enhanced DES algorithm that provides 168-bit encryption by processing data three times with three different keys.

Configuring IPSec

To implement IPSec, complete the following configuration procedures:

Considering General IPSec Issues

Before you configure IPSec, it is helpful to understand some general guidelines.

- IPv4 and IPv6 traffic and tunnels—You can configure IPSec tunnels to carry traffic in the following ways: IPv4 traffic traveling over IPv4 IPSec tunnels, IPv6 traffic traveling over IPv4 IPSec tunnels, IPv4 traffic traveling over IPv6 IPSec tunnels, and IPv6 traffic traveling over IPv6 IPSec tunnels.
- Configuration syntax differences between the AS and MultiServices PICs and the ES PIC—There are slight differences in the configuration statements and operational mode commands that are used with the PICs that support IPSec. As a result, the syntax for the AS and MultiServices PICs cannot be used interchangeably with the syntax for the ES PIC. However, the syntax for one type of PIC can be converted to its equivalent syntax on the other PIC for interoperability. The differences are highlighted in Table 39 on page 411.
- Configuring keys for authentication and encryption—When preshared keys are required for authentication or encryption, you must use the guidelines shown in Table 40 on page 412 to implement the correct key size.
- Rejection of weak and semiweak keys—The DES and 3DES encryption algorithms will reject weak and semiweak keys. As a result, do not create and use keys that contain the patterns listed in Table 41 on page 413.

Table 39: Comparison of IPSec Configuration Statements and Operational Mode Commands for the AS and MultiServices PICs and ES PIC

AS and MultiServices PICs Statements and Commands	ES PIC Statements and Commands
Configuration Mode Statements	
[edit service-set *name*]	–
[edit services ipsec-vpn ike] ■ policy {…} ■ proposal {…}	[edit security ike] ■ policy {…} ■ proposal {…}
[edit services ipsec-vpn ipsec] ■ policy {…} ■ proposal {…}	[edit security ipsec] ■ policy {…} ■ proposal {…}
[edit services ipsec-vpn rule *rule-name*] ■ remote-gateway *address*	[edit interface es- *fpc* / *pic* /*port*] ■ tunnel destination *address*
[edit services ipsec-vpn rule *rule-name* term *term-name*] ■ from *match-conditions* {…} then dynamic {…} ■ from *match-conditions* {…} then manual {…}	[edit security ipsec] ■ security-association *name* dynamic {…} ■ security-association *name* manual {…}

Table 39: Comparison of IPSec Configuration Statements and Operational Mode Commands for the AS and MultiServices PICs and ES PIC *(continued)*

AS and MultiServices PICs Statements and Commands	ES PIC Statements and Commands
[edit services ipsec-vpn rule-set]	–
[edit services service-set ipsec-vpn] ■ local-gateway *address*	[edit interface es- *fpc /pic /port*] ■ tunnel source *address*
Operational Mode Commands	
clear security pki ca-certificate	–
clear security pki certificate-request	–
clear security pki local-certificate	–
clear services ipsec-vpn certificates	–
request security pki ca-certificate enroll	request security certificate (unsigned)
request security pki ca-certificate load	request system certificate add
request security pki generate-certificate-request	–
request security pki generate-key-pair	request security key-pair
request security pki local-certificate enroll	request security certificate (signed)
request security pki local-certificate load	request system certificate add
show security pki ca-certificate	show system certificate
show security pki certificate-request	–
show security pki crl	–
show security pki local-certificate	show system certificate
show services ipsec-vpn certificates	show ipsec certificates
show services ipsec-vpn ike security-associations	show ike security-associations
show services ipsec-vpn ipsec security-associations	show ipsec security-associations

Table 40: Authentication and Encryption Key Lengths

	Number of Hexadecimal Characters	Number of ASCII Characters
Authentication		
HMAC-MD5-96	32	16

Table 40: Authentication and Encryption Key Lengths *(continued)*

	Number of Hexadecimal Characters	Number of ASCII Characters
Authentication		
HMAC-SHA1-96	40	20
Encryption		
AES-128-CBC	16	32
AES-192-CBC	24	48
AES-256-CBC	32	64
DES-CBC	16	8
3DES-CBC	48	24

Table 41: Weak and Semiweak Keys

Weak Keys			
0101	0101	0101	0101
1F1F	1F1F	1F1F	1F1F
E0E0	E0E0	E0E0	E0E0
FEFE	FEFE	FEFE	FEFE
Semiweak Keys			
01FE	01FE	01FE	01FE
1FE0	1FE0	0EF1	0EF1
01E0	01E0	01F1	01F1
1FFE	1FFE	0EFE	0EFE
011F	011F	010E	010E
E0FE	E0FE	F1FE	F1FE
FE01	FE01	FE01	FE01
E01F	E01F	F10E	F10E
E001	E001	F101	F101
FEF1	FEF1	FE0E	FE0E
1F01	1F01	0E01	0E01

Table 41: Weak and Semiweak Keys *(continued)*

Weak Keys			
FEE0	FEE0	FEF1	FEF1

Keep in mind the following limitations of IPSec services on the AS PIC:

- The AS PIC does not transport packets containing IPv4 options across IPSec tunnels. If you try to send packets containing IP options across an IPSec tunnel, the packets are dropped. Also, if you issue a ping command with the record-route option across an IPSec tunnel, the ping command fails.
- The AS PIC does not transport packets containing the following IPv6 options across IPSec tunnels: hop-by-hop, destination (Type 1 and 2), and routing. If you try to send packets containing these IPv6 options across an IPSec tunnel, the packets are dropped.
- Destination class usage is not supported with IPSec services on the AS PIC.

Configuring Security Associations

The first IPSec configuration step is to select a type of security association for your IPSec connection. You must statically configure all specifications for manual SAs, but you can rely on some defaults when you configure an IKE dynamic SA. To configure a security association, see the following sections.

Configuring Manual SAs

On the ES PIC, you configure a manual security association at the [edit security ipsec security-association *name*] hierarchy level. Include your choices for authentication, encryption, direction, mode, protocol, and SPI. Be sure that these choices are configured exactly the same way on the remote IPSec gateway.

```
[edit security]
ipsec {
   security-association  sa-name  {
      description  description;
      manual {
         direction (inbound | outbound | bidirectional) {
            authentication {
               algorithm (hmac-md5-96 | hmac-sha1-96);
               key (ascii-text key | hexadecimal key);
            }
            auxiliary-spi auxiliary-spi;
            encryption {
               algorithm (des-cbc | 3des-cbc);
               key (ascii-text key  | hexadecimal key);
            }
            protocol (ah | esp | bundle);
            spi  spi-value;
         }
```

```
        }
        mode (tunnel | transport);
    }
}
```

On the AS and MultiServices PICs, you configure a manual security association at the [edit services ipsec-vpn rule *rule-name*] hierarchy level. Include your choices for authentication, encryption, direction, protocol, and SPI. Be sure that these choices are configured exactly the same way on the remote IPSec gateway.

```
[edit services ipsec-vpn]
rule rule-name {
    match-direction (input | output);
    term  term-name  {
        from {
            destination-address  address;
            source-address  address;
        }
        then {
            backup-remote-gateway address;
            clear-dont-fragment-bit;
            manual {
                direction (inbound | outbound | bidirectional) {
                    authentication {
                        algorithm (hmac-md5-96 | hmac-sha1-96);
                        key (ascii-text key | hexadecimal key);
                    }
                    auxiliary-spi spi-value;
                    encryption {
                        algorithm algorithm; # This can be aes-128-cbc, aes-192-cbc,
                        # aes-256-cbc, des-cbc, or 3des-cbc.
                        key (ascii-text key | hexadecimal key);
                    }
                    protocol (ah | bundle | esp);
                    spi spi-value;
                }
            }
            no-anti-replay;
            remote-gateway  address;
            syslog;
        }
    }
}
rule-set  rule-set-name  {
    [ rule  rule-names  ];
}
```

Configuring IKE Dynamic SAs

On the ES PIC, you configure an IKE dynamic SA at the [edit security ike] and [edit security ipsec] hierarchy levels. Include your choices for IKE policies and proposals, which include options for authentication algorithms, authentication methods, Diffie-Hellman groups, encryption, IKE modes, and preshared keys. The IKE policy must use the IP address of the remote end of the IPSec tunnel as the policy name.

Also, include your choices for IPSec policies and proposals, which include options for authentication, encryption, protocols, Perfect Forward Secrecy (PFS), and IPSec modes. Be sure that these choices are configured exactly the same way on the remote IPSec gateway.

```
[edit security]
ike {
   proposal ike-proposal-name {
      authentication-algorithm (md5 | sha1);
      authentication-method (dsa-signatures | pre-shared-keys | rsa-signatures);
      description description;
      dh-group (group1 | group2);
      encryption-algorithm (3des-cbc | des-cbc);
      lifetime-seconds seconds;
   }
   policy ike-peer-address {
      description description;
      encoding (binary | pem);
      identity identity-name;
      local-certificate certificate-filename;
      local-key-pair private-public-key-file;
      mode (aggressive | main);
      pre-shared-key (ascii-text key | hexadecimal key);
      proposals [ proposal-names ];
   }
}
ipsec {
   proposal ipsec-proposal-name  {
      authentication-algorithm (hmac-md5-96 | hmac-sha1-96);
      description description;
      encryption-algorithm (3des-cbc | des-cbc);
      lifetime-seconds seconds;
      protocol (ah | esp | bundle);
   }
   policy ipsec-policy-name {
      description description;
      perfect-forward-secrecy {
         keys (group1 | group2);
      }
      proposals [ proposal-names ];
   }
   security-association sa-name {
      description description;
      dynamic {
         ipsec-policy policy-name;
         replay-window-size (32 | 64);
      }
      mode (tunnel | transport);
   }
}
```

On the AS and MultiServices PICs, you configure an IKE dynamic security association at the [edit services ipsec-vpn ike], [edit services ipsec-vpn ipsec], and [edit services ipsec-vpn rule *rule-name*] hierarchy levels. Include your choices for IKE policies and proposals, which include options for authentication algorithms, authentication

methods, Diffie-Hellman groups, encryption, IKE modes, and preshared keys. Also, include your choices for IPSec policies and proposals, which include options for authentication, encryption, protocols, PFS, and IPSec modes. Be sure that these choices are configured exactly the same way on the remote IPSec gateway.

If you choose not to explicitly configure IKE and IPSec policies and proposals on the AS and MultiServices PICs, your configuration can default to some preset values. These default values are shown in Table 42 on page 417.

Table 42: IKE and IPSec Proposal and Policy Default Values for the AS and MultiServices PICs

IKE Policy Statement	Default Value
mode	main
proposals	default
IKE Proposal Statement	Default Value
authentication-algorithm	sha1
authentication-method	pre-shared-keys
dh-group	group2
encryption-algorithm	3des-cbc
lifetime-seconds	3600 (seconds)
IPSec Policy Statement	Default Value
perfect-forward-secrecy keys	group2
proposals	default
IPSec Proposal Statement	Default Value
authentication-algorithm	hmac-sha1-96
encryption-algorithm	3des-cbc
lifetime-seconds	28800 (seconds)
protocol	esp

NOTE: If you use the default IKE and IPSec policy and proposal values preset within the AS and MultiServices PICs, you must explicitly configure an IKE policy and include a preshared key. This is because the pre-shared-keys authentication method is one of the preset values in the default IKE proposal.

If you decide to configure values manually, the following information shows the complete statement hierarchy and options for dynamic IKE SAs on the AS and MultiServices PICs:

```
[edit services ipsec-vpn]
ike {
   proposal proposal-name {
      authentication-algorithm (md5 | sha1 | sha256);
      authentication-method (pre-shared-keys | rsa-signatures);
      description description;
      dh-group (group1 | group2);
      encryption-algorithm algorithm; # This can be aes-128-cbc, aes-192-cbc,
      # aes-256-cbc, des-cbc, or 3des-cbc.
      lifetime-seconds seconds;
   }
   policy policy-name {
      description description;
      local-id {
         ipv4_addr [ values ];
         key_id [ values ];
      }
      local-certificate certificate-id-name;
      mode (aggressive | main);
      pre-shared-key (ascii-text key | hexadecimal key);
      proposals [ proposal-names ];
      remote-id {
         ipv4_addr [ values ];
         key_id [ values ];
      }
   }
}
ipsec {
   proposal proposal-name {
      authentication-algorithm (hmac-md5-96 | hmac-sha1-96);
      description description;
      encryption-algorithm algorithm; # This can be aes-128-cbc, aes-192-cbc,
      # aes-256-cbc, des-cbc, or 3des-cbc.
      lifetime-seconds seconds;
      protocol (ah | esp | bundle);
   }
   policy policy-name {
      description description;
      perfect-forward-secrecy {
         keys (group1 | group2);
      }
      proposals [ proposal-names ];
   }
}
rule rule-name {
   match-direction (input | output);
   term term-name {
      from {
         destination-address address;
         source-address address;
      }
      then {
```

```
            backup-remote-gateway address;
            clear-dont-fragment-bit;
            dynamic {
               ike-policy policy-name;
               ipsec-policy policy-name;
            }
            no-anti-replay;
            remote-gateway address;
            syslog;
         }
      }
   }
   rule-set rule-set-name {
      [ rule rule-names ];
   }
```

Using a Filter to Select Traffic to Be Secured

For the ES PIC, you need to configure a firewall filter to direct traffic into the IPSec tunnel. To apply a security association to traffic that matches a firewall filter, include the ipsec-sa *sa-name* statement at the [edit firewall filter *filter-name* term *term-name* then] hierarchy level.

```
[edit firewall filter filter-name]
term term-name {
   from {
      source-address {
         ip-address;
      }
      destination-address {
         ip-address;
      }
   }
   then {
      count counter-name;
      ipsec-sa sa-name;
   }
}
term other {
   then accept;
}
```

For the AS and MultiServices PICs, you do not need to configure a separate firewall filter. A filter is already built into the IPSec VPN rule statement at the [edit services ipsec-vpn] hierarchy level. To apply a security association to traffic that matches the IPSec VPN rule, include the dynamic or manual statement at the [edit services rule *rule-name* term *term-name* then] hierarchy level. To specify whether the rule should match input or output traffic, include the match-direction statement at the [edit services rule *rule-name*] hierarchy level.

After defining the rules for your IPSec VPNs, you must apply the rules to a service set. To do this, include the ipsec-vpn-rules *rule-name* statement at the [edit services service-set *service-set-name*] hierarchy level. Include an IPv4 or IPv6 IPSec gateway

with the local-gateway *local-ip-address* statement at the [edit services service-set *service-set-name*] hierarchy level.

Also, you must select either a single interface or a pair of interfaces that participate in IPSec. To select a single interface, include the interface-service *interface-name* statement at the [edit services service-set *service-set-name*] hierarchy level. To select a pair of interfaces and a next hop, include the next-hop-service statement at the [edit services service-set *service-set-name*] hierarchy level and specify an inside interface and an outside interface. Only next-hop service sets support IPSec within Layer 3 VPNs and use of routing protocols over the IPSec tunnel.

```
[edit services]
service-set service-set-name {
    interface-service {
        service-interface interface-name;
    }
    next-hop-service {
        inside-service-interface interface-name;
        outside-service-interface interface-name;
    }
    ipsec-vpn-options {
        local-gateway local-ip-address <routing-instance instance-name>;
        trusted-ca ca-profile-name;
    }
    ipsec-vpn-rules rule-name;
}
ipsec-vpn {
    rule rule-name {
        term term-name {
            from {
                source-address {
                    ip-address;
                }
                destination-address {
                    ip-address;
                }
            }
            then {
                remote-gateway remote-ip-address;
                (dynamic | manual);
            }
        }
        match-direction output;
    }
}
```

Applying the Filter or Service Set to the Interface Receiving Traffic to Be Secured

For the ES PIC, apply your firewall filter on the input interface receiving the traffic that you wish to send to the IPSec tunnel. To do this, include the filter statement at the [edit interfaces *interface-name* unit *unit-number* family inet] hierarchy level.

```
[edit interfaces interface-name unit unit-number family inet]
filter {
```

```
    input filter-name;
}
```

For the AS and MultiServices PICs, apply your IPSec-based interface service set to the input interface receiving the traffic that you wish to send to the IPSec tunnel. To do this, include the service-set *service-set-name* statement at the [edit interfaces *interface-name* unit *unit-number* family inet service (input | output)] hierarchy level.

```
[edit interfaces interface-name unit unit-number family inet]
service {
    input {
        service-set service-set-name;
    }
    output {
        service-set service-set-name;
    }
}
```

To configure a next-hop-based service set on the AS and MultiServices PICs, include the service-domain statement at the [edit interfaces *interface-name* unit *unit-number*] hierarchy level and specify one logical interface on the AS PIC as an inside interface and a second logical interface on the AS PIC as an outside interface.

```
[edit interfaces sp-fpc/pic/port]
unit 0 {
    family inet {
        address ip-address;
    }
}
unit 1 {
    family inet;
    service-domain inside;
}
unit 2 {
    family inet;
    service-domain outside;
}
```

Option: Using Digital Certificates

A popular way for network administrators to scale an IPSec network is to use digital certificates instead of preshared keys. To enable digital certificates in your network, you need to use a combination of operational mode commands and configuration statements. The following steps enable you to implement digital certificates on J-series Services Routers and AS and MultiServices PICs installed in M-series and T-series routing platforms:

- Configuring a CA Profile on page 422
- Configuring a Certificate Revocation List on page 422
- Requesting a CA Digital Certificate on page 423
- Generating a Private/Public Key Pair on page 423
- Generating and Enrolling a Local Digital Certificate on page 424

Configuring a CA Profile

The CA profile contains the name and URL of the CA or RA, as well as some retry timer settings. CA certificates issued by Entrust, VeriSign, and Microsoft are all compatible with J-series, M-series, and T-series routing platforms. To configure the domain name of the CA or RA, include the ca-identity statement at the [edit security pki ca-profile *ca-profile-name*] hierarchy level. To configure the URL of the CA, include the url statement at the [edit security pki ca-profile *ca-profile-name* enrollment] hierarchy level. To configure the number of enrollment attempts the routing platform should perform, include the retry statement at the [edit security pki ca-profile *ca-profile-name* enrollment] hierarchy level. To configure the amount of time the routing platform should wait between enrollment attempts, include the retry-interval statement at the [edit security pki ca-profile *ca-profile-name* enrollment] hierarchy level.

```
[edit security pki]
ca-profile ca-profile-name {
    ca-identity ca-identity;
    enrollment {
        url url-name;
        retry number-of-enrollment-attempts; # The range is 0 though 100 attempts.
        retry-interval seconds; # The range is 0 though 3600 seconds.
    }
}
```

Configuring a Certificate Revocation List

A certificate revocation list (CRL) contains a list of digital certificates that have been canceled before their expiration date. When a participating peer uses a digital certificate, it checks the certificate signature and validity. It also acquires the most recently issued CRL and checks that the certificate serial number is not on that CRL. By default, CRL verification is enabled on any CA profile running on JUNOS Release 8.1 or later. To disable CRL verification, include the disable statement at the [edit security pki ca-profile *ca-profile-name* revocation-check] hierarchy level.

To specify the URL for the Lightweight Directory Access Protocol (LDAP) server where your CA stores its current CRL, include the url statement at the [edit security pki ca-profile *ca-profile-name* revocation-check crl] hierarchy level. If the LDAP server requires a password to access the CRL, include the password statement at the [edit security pki ca-profile *ca-profile-name* revocation-check crl url] hierarchy level.

NOTE: You do not need to specify a URL for the LDAP server if the certificate includes a certificate distribution point (CDP). The CDP is a field within the certificate that contains information about how to retrieve the CRL for the certificate. The routing platform uses this information to download the CRL automatically. Any LDAP URL you configure takes precedence over the CDP included in the certificate.

If you manually downloaded the CRL, you must manually install it on the routing platform. To manually install the CRL, issue the request security pki crl load ca-profile *ca-profile-name* filename *path/filename* command.

To configure the time interval between CRL updates, include the refresh-interval statement at the [edit security ca-profile *ca-profile-name* revocation-check crl] hierarchy level.

To override the default behavior and permit IPSec peer authentication to continue when the CRL fails to download, include the disable on-download-failure statement at the [edit security ca-profile *ca-profile-name* revocation-check crl] hierarchy level.

```
[edit security pki ca-profile ca-profile-name]
revocation-check {
    disable;
    crl {
        disable on-download-failure;
        refresh-interval number-of-hours { # The range is 0 through 8784 hours.
            url {
                url-name;
                password;
            }
        }
    }
}
```

Requesting a CA Digital Certificate

You can request a CA digital certificate either online or manually. To request a digital certificate from a CA or RA online by using SCEP, issue the request security pki ca-certificate enroll ca-profile *ca-profile-name* command.

If you obtained the CA digital certificate manually through e-mail or other out-of-band mechanism, you must load it manually. To manually install a certificate in your routing platform, issue the request security pki local-certificate load command.

Generating a Private/Public Key Pair

A key pair is a critical element of a digital certificate implementation. The public key is included in the local digital certificate and the private key is used to decrypt data received from peers. To generate a private/public key pair, issue the request security pki generate-key-pair certificate-id *certificate-id-name* command.

Generating and Enrolling a Local Digital Certificate

You can generate and enroll a local digital certificate either online or manually. To generate and enroll a local certificate online by using SCEP, issue the request security pki local-certificate enroll command. To generate a local certificate request manually in the PKCS-10 format, issue the request security pki generate-certificate-request command.

If you create the local certificate request manually, you must also load the certificate manually. To manually install a certificate in your routing platform, issue the request security pki local-certificate load command.

Applying the Local Digital Certificate to an IPSec Configuration

To activate a local digital certificate, you configure the IKE proposal to use digital certificates instead of preshared keys, reference the local certificate in the IKE policy, and identify the CA or RA in the service set. To enable the IKE proposal for digital certificates, include the rsa-signatures statement at the [edit services ipsec-vpn ike proposal *proposal-name* authentication-method] hierarchy level. To reference the local certificate in the IKE policy, include the local-certificate statement at the [edit services ipsec-vpn ike policy *policy-name*] hierarchy level. To identify the CA or RA in the service set, include the trusted-ca statement at the [edit services service-set *service-set-name* ipsec-vpn-options] hierarchy level.

```
[edit services]
service-set service-set-name {
    .....
    ipsec-vpn-options {
        trusted-ca ca-profile-name;
    }
}
ipsec-vpn {
    ike {
        proposal proposal-name  {
            .....
            authentication-method [pre-shared-keys | rsa-signatures];
        }
        policy policy-name {
            ....
            local-certificate certificate-id-name;
        }
    }
}
```

Configuring Automatic Reenrollment of Digital Certificates

You can configure automatic reenrollment for digital certificates. This feature is by default not enabled. To configure automatic reenrollment for digital certificates, include the auto-re-enrollment statement at the [edit security pki] hierarchy level:

```
[edit]
```

```
security {
   pki {
      auto-re-enrollment {
         certificate-id certificate-name {
            ca-profile ca-profile-name;
            challenge-password password;
            re-enroll-trigger-time-percentage percentage; # Percentage of validity-period
               # (specified in certificate) when automatic
               # reenrollment should be initiated.
            re-generate-keypair;
            validity-period number-of-days;
         }
      }
   }
}
```

Monitoring Digital Certificates

Purpose You can issue various forms of the show security pki command to view digital certificates and certificate requests and certificate revocation lists:

Action

- To display the CA digital certificate, issue the show security pki ca-certificate ca-profile *ca-profile-name* command.
- To display the local digital certificate and the public key used to enroll the certificate, issue the show security pki local-certificate certificate-id *certificate-id-name* command.
- To display the local certificate request in PKCS-10 format, issue the show security pki certificate-request certificate-id *certificate-id-name* command.
- You can also view which digital certificates are used in IKE negotiations to establish IPSec tunnels by issuing the show services ipsec-vpn certificates command.
- To display the certificate revocation list, issue the show security pki crl ca-profile *ca-profile-name* command.
- To determine if a certificate is enabled for automatic-reenrollment, issue the show security pki command.

Clearing Digital Certificates

Purpose Variations of the clear security pki command enable you to delete certificates or requests and certificate revocation lists:

Action

- To delete the CA digital certificate, issue the clear security pki ca-certificate ca-profile *ca-profile-name* command.
- To delete the local digital certificate and the associated private/public key pair, issue the clear security pki local-certificate certificate-id *certificate-id-name* command.
- To delete the local certificate request, issue the clear security pki certificate-request certificate-id *certificate-id-name* command.

- To clear the digital certificates that were used in IKE negotiations to establish IPSec tunnels, issue the clear services ipsec-vpn certificates command.
- To delete the certificate revocation list, issue the clear security pki crl ca-profile *ca-profile-name* command.

Related Topics To see a full example showing the use of digital certificates in an IPSec topology, see Example: AS PIC IKE Dynamic SA with Digital Certificates Configuration on page 487. For more information about operational mode commands used with digital certificates, see the *JUNOS System Basics and Services Command Reference*. For more information about configuration statements used with digital certificates, see the *J-series Services Router Advanced WAN Access Configuration Guide*, the *JUNOS System Basics Configuration Guide*, and the *JUNOS Services Interfaces Configuration Guide*.

Option: Using Filter-Based Forwarding to Select Traffic to Be Secured

Instead of using a firewall filter, you can also forward traffic into an IPSec security association by using a filter-based forwarding instance. First, configure the filter-based forwarding instance. Then, configure a routing table group to advertise the routes from the filter-based forwarding instance. Next, create a firewall filter for the ES PIC and reference the filter-based forwarding instance. Lastly, apply the filter and IPSec security association to the ES PIC.

```
[edit]
routing-instances {
    forwarding {
        instance-type forwarding;
        routing-options {
            static {
                route 10.10.10.0/24 next-hop 192.168.0.5;
            }
        }
    }
}
routing-options {
    rib-groups {
        group-name  {
            import-rib [ inet.0 forwarding.inet.0 ];
        }
    }
}
firewall {
    family inet {
        filter filter-name {
            term term-name {
                then routing-instance instance-name;
            }
        }
    }
}
[edit]
interfaces {
    es-0/0/0 {
```

```
        unit 0 {
            tunnel {
                source source-ip-address;
                destination destination-ip-address;
            }
            family inet {
                ipsec-sa sa-name;
                filter {
                    input filter-name;
                }
                address ip-address;
            }
        }
    }
}
```

Option: Using IPSec with a Layer 3 VPN

Some key concepts to keep in mind when configuring IPSec within a VPN include the following:

- Add the outside services interface for a next-hop style service set into the routing instance by including the interface sp-*fpc/pic/port* statement at the [edit routing-instances *instance-name*] hierarchy level.
- For interface style service sets, add the interface on which you apply the service set and the services interface by including both interfaces at the [edit routing-instances *instance-name*] hierarchy level.
- To define a routing instance for the local gateway within the service set, include the routing-instance *instance-name* option at the [edit services service-set *service-set-name* ipsec-vpn-options local-gateway *address*] hierarchy level.

The following configuration for an AS PIC on a provider edge (PE) router demonstrates the use of next-hop service sets with an IKE dynamic SA in a VPN routing and forwarding (VRF) routing instance.

```
[edit]
interfaces {
    so-0/0/0 {
        description "Interface connected to the customer edge (CE) router";
        unit 0 {
            family inet {
                address 10.6.6.6/32;
            }
        }
    }
    so-2/2/0 {
        description "Source IPSec tunnel interface to the network core";
        unit 0 {
            family inet {
                address 10.10.1.1/30;
            }
        }
    }
```

```
    sp-3/1/0 {
        description "AS PIC interface";
        unit 0 {
            family inet {
                address 10.7.7.7/32;
            }
        }
        unit 1 {
            family inet;
            service-domain inside;
        }
        unit 2 {
            family inet;
            service-domain outside;
        }
    }
}
policy-options {
    policy-statement vpn-export-policy {
        then {
            community add community-name;
            accept;
        }
    }
    policy-statement vpn-import-policy {
        term term-name {
            from community community-name;
            then accept;
        }
    }
    community community-name members target:100:20;
}
routing-instances {
    vrf {
        instance-type vrf;
        interface sp-3/1/0.1; # Inside sp interface.
        interface so-0/0/0.0; # Interface that connects to the CE router.
        route-distinguisher route-distinguisher;
        vrf-import vpn-import-policy;
        vrf-export vpn-export-policy;
        routing-options {
            static {
                route ip-address/prefix next-hop so-0/0/0.0; # Routes for the CE router.
                route ip-address/prefix next-hop sp-3/1/0.1; # Routes for IPSec.
            }
        }
    }
}
services {
    service-set service-set-name {
        next-hop-service {
            inside-service-interface sp-3/1/0.1;
            outside-service-interface sp-3/1/0.2;
        }
        ipsec-vpn-options {
            local-gateway 10.10.1.1;
```

```
        }
        ipsec-vpn-rules rule-name;
    }
    ipsec-vpn {
        rule rule-name {
            term term-name {
                from {
                    source-address {
                        source-ip-address;
                    }
                }
                then {
                    remote-gateway 10.10.1.2;
                    dynamic {
                        ike-policy ike-policy-name;
                    }
                }
            }
            match-direction direction;
        }
        ike {
            policy  ike-policy-name  {
                pre-shared-key ascii-text preshared-key;
            }
        }
    }
}
```

For more information on VRF routing instances, see the *JUNOS VPNs Configuration Guide*. For more information on next-hop service sets, see the *JUNOS Services Interfaces Configuration Guide*.

Option: Securing BGP Sessions with Transport Mode

For the ES PIC, you can use IPSec to secure BGP sessions between Routing Engines in M-series and T-series platforms. To configure, create a transport mode security association and apply the SA to the BGP configuration by including the ipsec-sa statement at the [edit protocols bgp group *group-name*] hierarchy level.

```
[edit]
protocols {
    bgp {
        group group-name  {
            local-address ip-address;
            export export-policy;
            peer-as as-number;
            ipsec-sa sa-name;
            neighbor peer-ip-address;
        }
    }
}
```

Option: Securing OSPFv3 Networks with Transport Mode

OSPF version 3 (OSPFv3), unlike OSPF version 2, does not have a built-in authentication method and relies on IPSec to provide this functionality. Using the ES PIC syntax, you can use IPSec to secure OSPFv3 between Routing Engines in M-series and T-series platforms. You can secure specific OSPFv3 interfaces and protect OSPFv3 virtual links. To configure, create a transport mode security association and apply the SA to the OSPFv3 configuration by including the ipsec-sa statement at the [edit protocols ospf3 area *area-number* interface *interface-name*] or [edit protocols ospf3 area *area-number* virtual-link neighbor-id *neighbor-ip-address* transit-area *area-number*] hierarchy level.

```
[edit]
protocols {
   ospf3 {
      area  area-number  {
         interface  interface-name  {
            ipsec-sa  sa-name;
         }
         virtual-link neighbor-id  neighbor-ip-address  transit-area  area-number  {
            ipsec-sa  sa-name;
         }
      }
   }
}
```

Option: Securing OSPFv2 Networks with Transport Mode

By default, you can configure MD5 or simple text password-based authentication over OSPFv2 links. In addition to these basic authentications, the JUNOS software supports OSPFv2 with a security authentication header (AH), Encapsulating Security Payload (ESP), or an IPSec protocol bundle that supports both AH and ESP. You can configure IPSec over OSPFv2 using transport mode security associations on physical, sham, or virtual links.

Because the JUNOS software supports only bidirectional security associations over OSPFv2, OSPFv2 peers must be configured with the same IPSec security association. Configuring OSPFv2 peers with different security associations or with dynamic IKE will prevent adjacencies from being established. In addition, you must configure identical security associations for sham links with the same remote endpoint address, for virtual links with the same remote endpoint address, for all neighbors on OSPF nonbroadcast multiaccess (NBMA) or point-to-multipoint links, and for every subnet that is part of a broadcast link.

To create a manual bidirectional security association, include the security-association *security-association-name* statement at the [edit security ipsec] hierarchy level:

```
[edit]
security {
   ipsec {
      security-association security-association name {
         mode transport;
```

```
        manual {
            direction bidirectional {
                protocol (ah | esp | bundle);
                spi spi–value;
                authentication {
                    algorithm (hmac-md5-96 | hmac-sha1-96);
                    key (ascii-text key | hexadecimal key);
                }
            }
        }
    }
}
```

To configure IPSec on an OSPFv2 interface, create a transport mode security association and include the **ipsec-sa name** statement at the [**edit protocols ospf area** *area-id*] hierarchy level:

```
[edit]
protocols {
    ospf {
        area area-id {
            interface interface-name {
                ipsec-sa sa-name;
            }
            virtual-link neighbor-id a.b.c.d transit-area x.x.x.x {
                ipsec-sa sa-name;
            }
            sham-link-remote {
                ipsec-sa sa-name;
            }
        }
    }
}
```

To verify your configuration, enter the **show ospf interface detail** command. This command gives detailed information about the **ospfv2** interface and displays the interface's security association at the bottom of the output. In the example below, the security association configured on this router is **sa1**.

```
user@router> show ospf interface detail
Interface             State     Area            DR ID           BDR ID Nbrs
fe-0/0/1.0            BDR      0.0.0.0         192.168.37.12   10.255.245.215 1
Type LAN, address 192.168.37.11, Mask 255.255.255.248, MTU 4460, Cost 40
DR addr 192.168.37.12, BDR addr 192.168.37.11, Adj count 1, Priority 128
Hello 10, Dead 40, ReXmit 5, Not Stub
t1-0/2/1.0            PtToPt   0.0.0.0         0.0.0.0         0.0.0.0 0
Type P2P, Address 0.0.0.0, Mask 0.0.0.0, MTU 1500, Cost 2604
Adj count 0
Hello 10, Dead 40, ReXmit 5, Not Stub
Auth type: MD5, Active key ID 3, Start time 2002 Nov 19 10:00:00 PST
IPsec SA Name: sa1
```

Option: Monitoring IPSec by Using SNMP

In JUNOS Release 7.5 and later, the IPSec Monitoring MIB provides a way to monitor IPSec information on J-series Services Routers and AS PICs installed in M-series and T-series routing platforms by using Simple Network Management Protocol (SNMP). The MIB provides an IKE tunnel table to monitor IKE security associations and view related statistics, an IPSec tunnel table to view IPSec tunnel statistics, and an IPSec security associations table to view all IPSec SAs. For more information, see the *JUNOS Network Management Configuration Guide*.

Option: Configuring IPSec Dynamic Endpoints

IPSec tunnels can also be established using *dynamic peer* security gateways, in which the remote end of the tunnels do not have a statically assigned IPv4 or IPv6 address. Since the remote address is not known and is assigned from an address pool each time the remote host reboots, establishment of the tunnel relies on using IKE main mode with preshared global keys. Both policy-based and link-type tunnels are supported as follows:

- Policy-based tunnels used shared mode.
- Link-type or routed tunnels use dedicated mode. Each tunnel allocates a service interface from a pool of interfaces configured for the dynamic peers. Routing protocols can be configured to run on these service interfaces to learn routes over the IPSec tunnel that is used as a link.

This section includes the following topics:

Dynamic Endpoint Tunnel Architecture

When you configure dynamic endpoint tunnels, the following components are used:

Authentication Process

The remote dynamic peer initiates IKE and IPSec negotiations with the local (Juniper Networks) routing platform. The local router uses a default set of authentication and encryption values to match the IPSec and IKE proposals sent by the remote peer to

establish the SA. If any of the values match, the tunnel establishment process continues. The default values are shown in Table 43 on page 433.

Table 43: Default IKE and IPSec Proposals for Dynamic SA Negotiations

Statement Name	Values
Implicit IKE Proposal	
authentication-method	preshared keys
dh-group	group1, group2
authentication-algorithm	sha1, md5
encryption-algorithm	3des-cbc, des-cbc
lifetime-seconds	3600 seconds
Implicit IPSec Proposal	
protocol	esp, ah, bundle
authentication-algorithm	hmac-sha1-96, hmac-md5-96
encryption-algorithm	3des-cbc, des-cbc
lifetime-seconds	28,800 seconds (8 hours)

Phase 2 of the authentication process matches the *proxy identities* of the protected hosts and networks sent by the peer against a list of configured proxy identities. The accepted proxy identity is used to create the dynamic rules for encrypting the traffic. You can configure proxy identities by including the allowed-proxy-pair statement in an IKE access profile at the [edit access profile *profile-name* client * ike] hierarchy level. If no configured entry matches, the negotiation is rejected.

However, if you do not configure the allowed-proxy-pair statement, the default value ANY(0.0.0.0/0)-ANY is applied, and the local router accepts any proxy identities sent by the peer.

Once the phase 2 negotiation has been successfully completed, the routing platform builds dynamic rules and inserts the reverse route into the routing table using the accepted proxy identity.

Dynamic Implicit Rules

After successful negotiation with the dynamic peer, the key management process (kmd) creates a dynamic rule for the accepted phase 2 proxy and applies it on the local AS or MultiServices PIC. The source and destination addresses are specified by the accepted proxy. This rule is used to encrypt traffic directed to one of the end hosts in the phase 2 proxy identity.

NOTE: You do not configure this rule; it is created by the key management process (kmd).

The ipsec-inside-interface value is the interface name assigned to the dynamic tunnel. The source-address and destination-address values are accepted from the proxy ID. The match-direction value is input for next-hop-style service sets.

Rule lookup for static tunnels is unaffected by the presence of a dynamic rule; it is performed in the order configured. When a packet is received for a service-set, static rules are always matched first. Dynamic rules are matched only after the rule match for static rules has failed.

Reverse Route Insertion

Static routes are automatically inserted into the route table for those networks and hosts protected by a remote tunnel endpoint. These protected hosts and networks are known as remote proxy identities.

Each route is created based on the remote proxy network and prefix length sent by the peer and is inserted in the relevant route table after successful phase 1 and phase 2 negotiations.

The route preference for each of these static reverse routes is 1. This value is necessary to avoid conflict with similar routes that might be added by the routing protocol process (rpd).

No routes are added if the accepted remote proxy address is the default (0.0.0.0/0). In this case, you can run routing protocols over the IPSec tunnel to learn routes and add static routes for the traffic you want to be protected over this tunnel.

For next-hop style service sets, the reverse routes include next hops pointing to the locations specified by the inside-service-interface statements.

The selection of the routing table in which these routes are inserted depends on where you configure the inside-service-interface statement. If these interfaces are present in a VRF routing instance, then routes are added to the corresponding VRF routing table; otherwise, the routes are added to inet.0.

NOTE: Reverse route insertion takes place only for tunnels to dynamic peers. These routes are added only for next-hop style service sets.

Configuring an IKE Access Profile

You can configure only one tunnel profile per service set for all dynamic peers. The configured preshared key in the profile is used for IKE authentication of all dynamic peers terminating in that service set.

The IKE tunnel profile specifies all the information needed to complete the IKE negotiation. For more information on access profiles, see the *JUNOS System Basics Configuration Guide*.

```
[edit access]
profile profile-name {
    client * {
        ike {
            allowed-proxy-pair {
                remote remote-proxy-address local local-proxy-address;
            }
            pre-shared-key ([ ascii-text key-string ] | [hexadecimal key-string ]);
            interface-id string-value;
        }
    }
}
```

NOTE: For dynamic peers, the JUNOS software supports only IKE main mode with the preshared key method of authentication. In this mode, an IPv4 or IPv6 address is used to identify a tunnel peer to get the preshared key information. The client value * (wildcard) means that the configuration within this profile is valid for all dynamic peers terminating within the service set accessing this profile.

The following statements are the parts of the IKE profile:

- allowed-proxy-pair—During phase 2 IKE negotiation, the remote peer supplies its network address (remote) and its peer's network address (local). Since multiple dynamic tunnels are authenticated through the same mechanism, this statement must include the list of possible combinations. If the dynamic peer does not present a valid combination, the phase 2 IKE negotiation fails.

  By default, remote 0.0.0.0/0 local 0.0.0.0/0 is used if no values are configured.

- pre-shared-key—Mandatory key used to authenticate the dynamic peer during IKE phase 1 negotiation. This key must be configured on both ends of the tunnel and distributed through an out-of-band secure mechanism. You can configure the key value either in hexadecimal or ascii-text format.

- interface-id—Interface identifier, a mandatory attribute used to derive the logical service interface information for the session.

Configuring the Service Set

To complete a dynamic endpoint tunnel configuration, you need to reference the IKE access profile configured at the [edit access] hierarchy level in the service set. To do this, include the ike-access-profile statement at the [edit services service-set *name* ipsec-vpn-options] hierarchy level:

```
[edit services]
service-set name {
    next-hop-service {
        inside-service-interface interface-name;
```

```
        outside-service-interface interface-name;
    }
    ipsec-vpn-options {
        local-gateway address;
        ike-access-profile profile-name;
    }
}
```

You can reference only one access profile in each service set. This profile is used to negotiate IKE and IPSec security associations with dynamic peers only.

NOTE: If you configure an IKE access profile in a service set, no other service set can share the same local-gateway address.

Configuring the Interface Identifier

You can configure an interface identifier for a group of dynamic peers, which specifies which adaptive services logical interface(s) take part in the dynamic IPSec negotiation. By assigning the same interface identifier to multiple logical interfaces, you can create a pool of interfaces for this purpose. To configure, include the ipsec-interface-id statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces sp-fpc/pic/port]
unit logical-unit-number {
    dial-options {
        ipsec-interface-id identifier;
        (shared | dedicated);
    }
}
```

Specifying the interface identifierin the dial-options statement makes this logical interface part of the pool identified by the IPSec interface identifier.

NOTE: Only one interface identifier can be specified at a time. You can include the ipsec-interface-id statement or the l2tp-interface-id statement, but not both simultaneously.

The shared statement enables one logical interface to be shared across multiple tunnels. The dedicated statement specifies that the logical interface is associated with a single tunnel, which is necessary when you are configuring an IPSec link-type tunnel. You must include the dedicated statement when you specify an ipsec-interface-id value.

Option: Configuring Multiple Routed Tunnels in a Single Next-Hop Service Set

To save you time and simplify your configurations, an enhancement to the JUNOS software enables you to configure several routed IPSec tunnels within a single next-hop service set. To configure, establish multiple services interfaces as inside

interfaces by including the service-domain inside statement at the [edit interfaces sp-*fpc/pic/port* unit *logical-unit-number*] hierarchy level. Then, include the ipsec-inside-interface statement at the [edit services ipsec-vpn rule *rule-name* term *term-name* from] hierarchy level.

NOTE: The full IPSec and IKE proposals and policies are not shown in the following example for the sake of brevity. For more information on proposals and policies, see "Configuring IKE Dynamic SAs" on page 415.

```
[edit]
interfaces {
   sp-3/3/0 {
      unit 3 {
         family inet;
         service-domain inside;
      }
      unit 4 {
         family inet;
         service-domain outside;
      }
      unit 5 {
         family inet;
         service-domain inside;
      }
   }
}
services {
   service-set link_type_ss_1 {
      next-hop-service {
         inside-service-interface sp-3/3/0.3;
         outside-service-interface sp-3/3/0.4;
      }
      ipsec-vpn-options {
         local-gateway 10.8.7.2;
      }
      ipsec-vpn-rules link_rule_1;
   }
   ipsec-vpn {
      rule link_rule_1 {
         term 1 {
            from {
               ipsec-inside-interface sp-3/3/0.3;
            }
            then {
               remote-gateway 10.10.7.3;
               backup-remote-gateway 10.8.7.1;
               dynamic {
                  ike-policy main_mode_ike_policy;
                  ipsec-policy dynamic_ipsec_policy;
               }
            }
         }
         term 2 {
```

```
            from {
                ipsec-inside-interface sp-3/3/0.5;
            }
            then {
                remote-gateway 10.12.7.5;
                dynamic {
                    ike-policy main_mode_ike_policy;
                    ipsec-policy dynamic_ipsec_policy;
                }
            }
        }
        match-direction input;
    }
  }
}
```

To confirm that your configuration is working, issue the show services ipsec-vpn ipsec security-associations command. Notice that each IPSec inside interface that you assigned to each IPSec tunnel is included in the output of this command.

```
user@router> show services ipsec-vpn ipsec security-associations
Service set: link_type_ss_1

  Rule: link_rule_1, Term: 1, Tunnel index: 1
  Local gateway: 10.8.7.2, Remote gateway: 10.8.7.1
  IPSec inside interface: sp-3/3/0.3
    Direction SPI         AUX-SPI     Mode       Type     Protocol
    inbound   3216392497  0           tunnel     dynamic  ESP
    outbound  398917249   0           tunnel     dynamic  ESP

  Rule: link_rule_1, Term: 2, Tunnel index: 2
  Local gateway: 10.8.7.2, Remote gateway: 10.12.7.5
  IPSec inside interface: sp-3/3/0.5
    Direction SPI         AUX-SPI     Mode       Type     Protocol
    inbound   762146783   0           tunnel     dynamic  ESP
    outbound  319191515   0           tunnel     dynamic  ESP
```

IPSec Configuration Examples

This section contains configuration examples and commands you can use to verify your IPSec configuration:

- Example: ES PIC Manual SA Configuration on page 439
- Example: AS PIC Manual SA Configuration on page 448
- Example: ES PIC IKE Dynamic SA Configuration on page 456
- Example: AS PIC IKE Dynamic SA Configuration on page 467
- Example: IKE Dynamic SA Between an AS PIC and an ES PIC Configuration on page 476
- Example: AS PIC IKE Dynamic SA with Digital Certificates Configuration on page 487
- Example: Dynamic Endpoint Tunneling Configuration on page 505

Example: ES PIC Manual SA Configuration

Figure 46: ES PIC Manual SA Topology Diagram

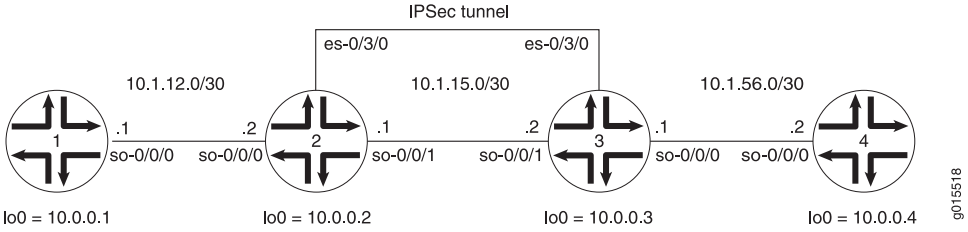


Figure 46 on page 439 shows an IPSec topology containing a group of four routers. Routers 2 and 3 establish an IPSec tunnel using an ES PIC and manual SA settings. Routers 1 and 4 provide basic connectivity and are used to verify that the IPSec tunnel is operational.

On Router 1, provide basic OSPF connectivity to Router 2.

Router 1

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R2 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.12.2/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.1/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.1;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface lo0.0;
        }
    }
}
```

On Router 2, enable OSPF as the underlying routing protocol to connect to Routers 1 and 3. Configure a bidirectional manual SA called sa-manual at the [edit security ipsec security-association] hierarchy level. Use AH for the protocol, 400 for the SPI, HMAC-MD5-96 for authentication, and a 32-bit hexadecimal authentication key for

the MD5 authentication key. (For more information about key length, see Table 40 on page 412.) Because you are using AH, there is no need to configure encryption.

To direct traffic into the ES PIC and the IPSec tunnel, create two firewall filters. The es-traffic filter matches inbound traffic from Router 1 destined for Router 4, whereas the es-return filter matches the return path from Router 4 to Router 1. Apply the es-traffic filter to the so-0/0/0 interface; then apply both the es-return filter and the sa-manual SA to the es-0/3/0 interface.

Router 2

```
[edit]
interfaces {
   so-0/0/0 {
      description "To R1 so-0/0/0";
      unit 0 {
         family inet {
            filter {
               input es-traffic; # Apply a filter that sends traffic to the IPSec tunnel here.
            }
            address 10.1.12.1/30;
         }
      }
   }
   so-0/0/1 {
      description "To R3 so-0/0/1";
      unit 0 {
         family inet {
            address 10.1.15.1/30;
         }
      }
   }
   es-0/3/0 {
      unit 0 {
         tunnel { # Specify the IPSec tunnel endpoints here.
            source 10.1.15.1;
            destination 10.1.15.2;
         }
         family inet {
            ipsec-sa sa-manual; # Apply the manual SA here.
            filter {
               input es-return; # Apply the filter that matches return IPSec traffic here.
            }
         }
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.0.0.2/32;
         }
      }
   }
}
routing-options {
   router-id 10.0.0.2;
}
```

```
protocols {
   ospf {
      area 0.0.0.0 {
         interface so-0/0/0.0;
         interface so-0/0/1.0;
         interface lo0.0;
      }
   }
}
security {
   ipsec {
      security-association sa-manual { # Define the manual SA specifications here.
         mode tunnel;
         manual {
            direction bidirectional {
               protocol ah;
               spi 400;
               authentication {
                  algorithm hmac-md5-96;
                  key hexadecimal "$9$r0/eK8x7VY2ahSvL7-2gfTQF9Apu1EhrmfF/CtI
                     RlKMW7-VwYg4ZhSeW8XbwoJGjHmP5QF69wY4Zjif5369ApBSyKv8XRE";
               }
            }
         }
      }
   }
}

# The 32-bit unencrypted hexadecimal key is abcdef01abcdef01abcdef01abcdef01.
firewall {
   filter es-traffic { # Define a filter that sends traffic to the IPSec tunnel here.
      term to-es {
         from {
            source-address {
               10.1.12.0/24;
            }
            destination-address {
               10.1.56.0/24;
            }
         }
         then {
            count ipsec-tunnel;
            ipsec-sa sa-manual;
         }
      }
      term other {
         then accept;
      }
   }
   filter es-return { # Define a filter that matches return IPSec traffic here.
      term return {
         from {
            source-address {
               10.1.56.0/24;
            }
            destination-address {
```

```
                10.1.12.0/24;
            }
          }
          then accept;
        }
      }
    }
```

On Router 3, enable OSPF as the underlying routing protocol to connect to Routers 2 and 4. Configure a bidirectional manual SA called sa-manual at the [edit security ipsec security-association] hierarchy level. Use the exact same specifications that you used for the SA on Router 2: AH for the protocol, 400 for the SPI, HMAC-MD5-96 for authentication, and a 32-bit hexadecimal authentication key of abcdef01abcdef01abcdef01abcdef01 for the MD5 authentication key. (For more information about authentication key length, see Table 40 on page 412.) Because you are using AH, there is no need to configure an encryption algorithm.

To direct traffic into the ES PIC and the IPSec tunnel, create two firewall filters. The es-traffic filter matches inbound traffic from Router 4 destined for Router 1, whereas the es-return filter matches the return path from Router 1 to Router 4. Apply the es-traffic filter to the so-0/0/0 interface; then apply both the es-return filter and the sa-manual SA to the es-0/3/0 interface.

Router 3

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R4 so-0/0/0";
        unit 0 {
            family inet {
                filter {
                    input es-traffic; # Apply a filter that sends traffic to the IPSec tunnel here.
                }
                address 10.1.56.1/30;
            }
        }
    }
    so-0/0/1 {
        description "To R2 so-0/0/1";
        unit 0 {
            family inet {
                address 10.1.15.2/30;
            }
        }
    }
    es-0/3/0 {
        unit 0 {
            tunnel { # Specify the IPSec tunnel endpoints here.
                source 10.1.15.2;
                destination 10.1.15.1;
            }
            family inet {
                ipsec-sa sa-manual; # Apply the manual SA here.
                filter {
                    input es-return; # Apply the filter that matches return IPSec traffic here.
                }
```

```
        }
      }
    }
    lo0 {
      unit 0 {
        family inet {
          address 10.0.0.3/32;
        }
      }
    }
}
routing-options {
    router-id 10.0.0.3;
}
protocols {
    ospf {
      area 0.0.0.0 {
        interface so-0/0/0.0;
        interface so-0/0/1.0;
        interface lo0.0;
      }
    }
}
security {
    ipsec {
      security-association sa-manual { # Define the manual SA specifications here.
        mode tunnel;
        manual {
          direction bidirectional {
            protocol ah;
            spi 400;
            authentication {
              algorithm hmac-md5-96;
              key hexadecimal "$9$KMfMWx-ds4oGyl87dboaQF36tu0BESyK5Q6
                Ap0hcvWLXdbs24aJDylMXxNY2ZUjk.5Tz36Ct24JDkqQz/CtuORleW8xNcS";
            }
          }
        }
      }
    }
}


## The 32-bit unencrypted hexadecimal key is abcdef01abcdef01abcdef01abcdef01.
firewall {
    filter es-traffic { # Define a filter that sends traffic to the IPSec tunnel here.
      term to-es {
        from {
          source-address {
            10.1.56.0/24;
          }
          destination-address {
            10.1.12.0/24;
          }
        }
```

```
            then {
                count ipsec-tunnel;
                ipsec-sa sa-manual;
            }
        }
        term other {
            then accept;
        }
    }
    filter es-return { # Define a filter that matches return IPSec traffic here.
        term return {
            from {
                source-address {
                    10.1.12.0/24;
                }
                destination-address {
                    10.1.56.0/24;
                }
            }
            then accept;
        }
    }
}
```

On Router 4, provide basic OSPF connectivity to Router 3.

Router 4

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R3 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.56.2/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.4/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.4;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface lo0.ping
        }
    }
}
```

Verifying Your Work

To verify proper operation of a manual IPSec SA on the ES PIC, use the following commands:

- ping
- show ipsec security-associations (detail)
- traceroute

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

- Router 1 on page 445
- Router 2 on page 445
- Router 3 on page 446
- Router 4 on page 447

Router 1

On Router 1, issue a ping command to the so-0/0/0 interface of Router 4 to send traffic across the IPSec tunnel.

```
user@R1> ping 10.1.56.2
PING 10.1.56.2 (10.1.56.2): 56 data bytes
64 bytes from 10.1.56.2: icmp_seq=0 ttl=253 time=0.939 ms
64 bytes from 10.1.56.2: icmp_seq=1 ttl=253 time=0.886 ms
64 bytes from 10.1.56.2: icmp_seq=2 ttl=253 time=0.826 ms
^C
--- 10.1.56.2 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.826/0.884/0.939/0.046 ms
```

You can also issue the traceroute command to verify that traffic to 10.1.56.2 travels over the IPSec tunnel between Router 2 and Router 3. Notice that the second hop does not reference 10.1.15.2—the physical interface on Router 3. Instead, the loopback address of 10.0.0.3 on Router 3 appears as the second hop. This indicates that the IPSec tunnel is operating correctly.

```
user@R1> traceroute 10.1.56.2
traceroute to 10.1.56.2 (10.1.56.2), 30 hops max, 40 byte packets
 1  10.1.12.1 (10.1.12.1)  0.655 ms  0.549 ms  0.508 ms
 2  10.0.0.3 (10.0.0.3)  0.833 ms  0.786 ms  0.757 ms
 3  10.1.56.2 (10.1.56.2)  0.808 ms  0.741 ms  0.716 ms
```

Router 2

Another way to verify that matched traffic is being diverted to the bidirectional IPSec tunnel is to view the firewall filter counter. After you issue the ping command from Router 1 (three packets), the es-traffic firewall filter counter looks like this:

```
user@R2> show firewall filter es-traffic
Filter: es-traffic
Counters:
Name                                                Bytes              Packets
ipsec-tunnel                                          252                    3
```

After you issue the ping command from both Router 1 (three packets) and Router 4 (two packets), the es-traffic firewall filter counter looks like this:

```
user@R2> show firewall filter es-traffic
Filter: es-traffic
Counters:
Name                                                Bytes              Packets
ipsec-tunnel                                          420                    5
```

To verify that the IPSec security association is active, issue the show ipsec security-associations detail command. Notice that the SA contains the settings you specified, such as AH for the protocol and HMAC-MD5-96 for the authentication algorithm.

```
user@R2> show ipsec security-associations detail
Security association: sa-manual, Interface family: Up

  Local gateway: 10.1.15.1, Remote gateway: 10.1.15.2
  Local identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)

   Direction: inbound, SPI: 400, AUX-SPI: 0
    Mode: tunnel, Type: manual, State: Installed
    Protocol: AH, Authentication: hmac-md5-96, Encryption: None
    Anti-replay service: Disabled

    Direction: outbound, SPI: 400, AUX-SPI: 0
    Mode: tunnel, Type: manual, State: Installed
    Protocol: AH, Authentication: hmac-md5-96, Encryption: None
    Anti-replay service: Disabled
```

Router 3

View the firewall filter counter to continue verifying that matched traffic is being diverted to the bidirectional IPSec tunnel. After you issue the ping command from Router 1 (three packets), the es-traffic firewall filter counter looks like this:

```
user@R3> show firewall filter es-traffic
Filter: es-traffic
Counters:
Name                                                Bytes              Packets
ipsec-tunnel                                          252                    3
```

After you issue the ping command from both Router 1 (three packets) and Router 4 (two packets), the es-traffic firewall filter counter looks like this:

```
user@R3> show firewall filter es-traffic
Filter: es-traffic
Counters:
```

```
Name                                                    Bytes              Packets
ipsec-tunnel                                              420                    5
```

To verify that the IPSec security association is active, issue the show ipsec security-associations detail command. Notice that the SA on Router 3 contains the same settings you specified on Router 2.

```
user@R3> show ipsec security-associations detail
Security association: sa-manual, Interface family: Up

  Local gateway: 10.1.15.2, Remote gateway: 10.1.15.1
  Local identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)

    Direction: inbound, SPI: 400, AUX-SPI: 0
    Mode: tunnel, Type: manual, State: Installed
    Protocol: AH, Authentication: hmac-md5-96, Encryption: None
    Anti-replay service: Disabled

    Direction: outbound, SPI: 400, AUX-SPI: 0
    Mode: tunnel, Type: manual, State: Installed
    Protocol: AH, Authentication: hmac-md5-96, Encryption: None
    Anti-replay service: Disabled
```

Router 4

On Router 4, issue a ping command to the so-0/0/0 interface of Router 1 to send traffic across the IPSec tunnel.

```
user@R4> ping 10.1.12.2
PING 10.1.12.2 (10.1.12.2): 56 data bytes
64 bytes from 10.1.12.2: icmp_seq=0 ttl=253 time=0.937 ms
64 bytes from 10.1.12.2: icmp_seq=1 ttl=253 time=0.872 ms
^C
--- 10.1.12.2 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.872/0.905/0.937/0.032 ms
```

You can also issue the traceroute command to verify that traffic to 10.1.12.2 travels over the IPSec tunnel between Router 3 and Router 2. Notice that the second hop does not reference 10.1.15.1—the physical interface on Router 2. Instead, the loopback address of 10.0.0.2 on Router 2 appears as the second hop. This indicates that the IPSec tunnel is operating correctly.

```
user@R4> traceroute 10.1.12.2
traceroute to 10.1.12.2 (10.1.12.2), 30 hops max, 40 byte packets
 1  10.1.56.1 (10.1.56.1)  0.670 ms  0.589 ms  0.548 ms
 2  10.0.0.2 (10.0.0.2)  0.815 ms  0.791 ms  0.763 ms
 3  10.1.12.2 (10.1.12.2)  0.798 ms  0.741 ms  0.714 ms
```

Example: AS PIC Manual SA Configuration

Figure 47: AS PIC Manual SA Topology Diagram

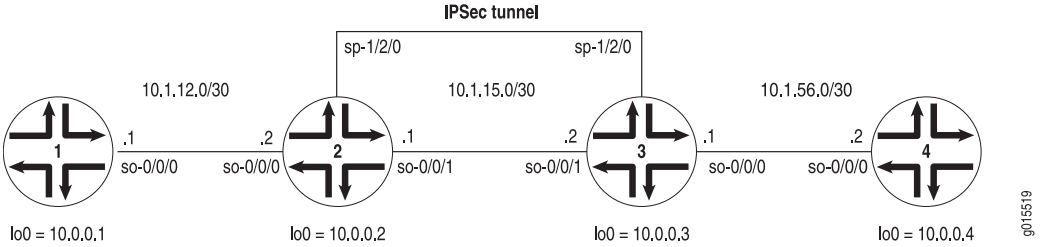


Figure 47 on page 448 shows a similar IPSec topology to the one used in the ES PIC manual SA example. The difference is that Routers 2 and 3 establish an IPSec tunnel using an AS PIC and use slightly modified manual SA settings. Routers 1 and 4 again provide basic connectivity and are used to verify that the IPSec tunnel is operational.

On Router 1, provide basic OSPF connectivity to Router 2.

Router 1

```
[edit]
interfaces {
   so-0/0/0 {
      description "To R2 so-0/0/0";
      unit 0 {
         family inet {
            address 10.1.12.2/30;
         }
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.0.0.1/32;
         }
      }
   }
}
routing-options {
   router-id 10.0.0.1;
}
protocols {
   ospf {
      area 0.0.0.0 {
         interface so-0/0/0.0;
         interface lo0.0;
      }
   }
}
```

On Router 2, enable OSPF as the underlying routing protocol to connect to Routers 1 and 3. Configure a bidirectional manual SA in a rule called rule-manual-SA-BiEspshades at the [edit ipsec-vpn rule] hierarchy level. Reference this

rule in a service set called service-set-manual-BiEspshades at the [edit services service-set] hierarchy level.

Configure all specifications for your manual SA. Use ESP for the protocol, 261 for the SPI, HMAC-SHA1-96 for authentication, DES-CBC for encryption, a 20-bit ASCII authentication key for the SHA-1 authentication key, and an 8-bit ASCII encryption key for the DES-CBC authentication key. (For more information about key lengths, see Table 40 on page 412.)

To direct traffic into the AS PIC and the IPSec tunnel, configure a next-hop style service set and add the adaptive services logical interface used as the IPSec inside interface into the OSPF configuration.

Router 2

```
[edit]
interfaces {
   so-0/0/0 {
      description "To R1 so-0/0/0";
      unit 0 {
         family inet {
            address 10.1.12.1/30;
         }
      }
   }
   so-0/0/1 {
      description "To R3 so-0/0/1";
      unit 0 {
         family inet {
            address 10.1.15.1/30;
         }
      }
   }
   sp-1/2/0 {
      services-options {
         syslog {
            host local {
               services info;
            }
         }
      }
      unit 0 {
         family inet {
         }
         unit 1 { # sp-1/2/0.1 is the IPSec inside interface.
            family inet;
            service-domain inside;
         }
         unit 2 { # sp-1/2/0.2 is the IPSec outside interface.
            family inet;
            service-domain outside;
         }
      }
      lo0 {
         unit 0 {
            family inet {
               address 10.0.0.2/32;
```

```
        }
      }
    }
}
routing-options {
    router-id 10.0.0.2;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface lo0.0;
            interface sp-1/2/0.1; # This sends OSPF traffic over the IPSec tunnel.
        }
    }
}
services {
    service-set service-set-manual-BiEspshades { # Define your service set here.
        next-hop-service { # Required for dynamic routing protocols such as OSPF.
            inside-service-interface sp-1/2/0.1;
            outside-service-interface sp-1/2/0.2;
        }
        ipsec-vpn-options {
            local-gateway 10.1.15.1; # Specify the local IP address of the IPSec tunnel.
        }
        ipsec-vpn-rules rule-manual-SA-BiEspshades; # Reference the IPSec rule here.
    }
    ipsec-vpn {
        rule rule-manual-SA-BiEspshades { # Define your IPSec VPN rule here.
            term term-manual-SA-BiEspshades {
                then {
                    remote-gateway 10.1.15.2; # The remote IP address of the IPSec tunnel.
                    manual { # Define the manual SA specifications here.
                        direction bidirectional {
                            protocol esp;
                            spi 261;
                            authentication {
                                algorithm hmac-sha1-96;
                                key ascii-text "$9$v.s8xd24Zk.5bs.5QFAtM8XNVYJGifT3goT369
                                    OBxNdw2ajHmFnCZUnCtuEh";
                                ## The unencrypted key is juniperjuniperjunipe (20 characters
                                    for HMAC-SHA-1-96).
                            }
                            encryption {
                                algorithm des-cbc;
                                key ascii-text "$9$3LJW/A0EclLxdBIxdbsJZn/CpOR";
                                ## The unencrypted key is juniperj (8 characters for DES-CBC).
                            }
                        }
                    }
                }
            }
            match-direction input; # Correct match direction for next-hop service sets.
        }
    }
}
```

```
    }
  security {
    pki {
      auto-re-enrollment {
        certificate-id certificate-name {
          ca-profile ca-profile-name;
          challenge-password password;
          re-enroll-trigger-time-percentage percentage; #Percentage of validity-period
            # (specified in certificate) when automatic
            # reenrollment should be initiated.
          re-generate-keypair;
          validity-period number-of-days;
        }
      }
    }
```

On Router 3, enable OSPF as the underlying routing protocol to connect to Routers 2 and 4. Configure a bidirectional manual SA in a rule called rule-manual-SA-BiEspshades at the [edit ipsec-vpn rule] hierarchy level. Reference this rule in a service set called service-set-manual-BiEspshades at the [edit services service-set] hierarchy level.

Configure the same specifications for your manual SA that you specified on Router 2. Use ESP for the protocol, 261 for the SPI, HMAC-SHA1-96 for authentication, DES-CBC for encryption, a 20-bit ASCII authentication key for the SHA-1 authentication key, and an 8-bit ASCII encryption key for the DES-CBC authentication key. (For more information about key lengths, see Table 40 on page 412.)

To direct traffic into the AS PIC and the IPSec tunnel, configure a next-hop style service set and add the adaptive services logical interface used as the IPSec inside interface into the OSPF configuration.

Router 3

```
[edit]
interfaces {
  so-0/0/0 {
    description "To R4 so-0/0/0";
    unit 0 {
      family inet {
        address 10.1.56.1/30;
      }
    }
  }
  so-0/0/1 {
    description "To R2 so-0/0/1";
    unit 0 {
      family inet {
        address 10.1.15.2/30;
      }
    }
  }
  sp-1/2/0 {
    services-options {
      syslog {
        host local {
          services info;
        }
```

```
        }
    }
    unit 0 {
        family inet {
        }
        unit 1 { # sp-1/2/0.1 is the IPSec inside interface.
            family inet;
            service-domain inside;
        }
        unit 2 { # sp-1/2/0.2 is the IPSec outside interface.
            family inet;
            service-domain outside;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.3/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.3;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface lo0.0;
            interface sp-1/2/0.1; # This sends OSPF traffic over the IPSec tunnel.
        }
    }
}
services {
    service-set service-set-manual-BiEspshades { # Define your service set here.
        next-hop-service { # Required for dynamic routing protocols such as OSPF.
            inside-service-interface sp-1/2/0.1;
            outside-service-interface sp-1/2/0.2;
        }
        ipsec-vpn-options {
            local-gateway 10.1.15.2; # Specify the local IP address of the IPSec tunnel.
        }
        ipsec-vpn-rules rule-manual-SA-BiEspshades; # Reference the IPSec rule here.
    }
    ipsec-vpn {
        rule rule-manual-SA-BiEspshades { # Define your IPSec VPN rule here.
            term term-manual-SA-BiEspshades {
                then {
                    remote-gateway 10.1.15.1; # The remote IP address of the IPSec tunnel.
                    manual { # Define the manual SA specifications here.
                        direction bidirectional {
                            protocol esp;
                            spi 261;
                            authentication {
                                algorithm hmac-sha1-96;
```

```
                        key ascii-text "$9$v.s8xd24Zk.5bs.5QFAtM8XNVYJGifT3goT369
                          OBxNdw2ajHmFnCZUnCtuEh";
                        ## The unencrypted key is juniperjuniperjunipe (20 characters
                          for HMAC-SHA-1-96).
                    }
                    encryption {
                        algorithm des-cbc;
                        key ascii-text "$9$3LJW/A0EclLxdBIxdbsJZn/CpOR";
                        ## The unencrypted key is juniperj (8 characters for DES-CBC).
                    }
                }
            }
        }
    }
    match-direction input; # Specify in which direction the rule should match.
  }
 }
}
```

On Router 4, provide basic OSPF connectivity to Router 3.

Router 4

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R3 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.56.2/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.4/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.4;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface lo0.0;
        }
    }
}
```

Verifying Your Work

To verify proper operation of a manual IPSec SA on the AS PIC, use the following commands:

- ping
- show services ipsec-vpn ipsec security-associations (detail)
- show services ipsec-vpn ipsec statistics

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

- Router 1 on page 454
- Router 2 on page 454
- Router 3 on page 455

Router 1

On Router 1, issue a ping command to the lo0 interface on Router 4 to send traffic across the IPSec tunnel.

```
user@R1> ping 10.0.0.4
PING 10.0.0.4 (10.0.0.4): 56 data bytes
64 bytes from 10.0.0.4: icmp_seq=0 ttl=254 time=1.375 ms
64 bytes from 10.0.0.4: icmp_seq=1 ttl=254 time=18.375 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=254 time=1.120 ms
^C
--- 10.0.0.4 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.120/6.957/18.375/8.075 ms
```

Router 2

To verify that the IPSec security association is active, issue the show services ipsec-vpn ipsec security-associations detail command. Notice that the SA contains the settings you specified, such as ESP for the protocol and HMAC-SHA1-96 for the authentication algorithm.

```
user@R2> show services ipsec-vpn ipsec security-associations detail
Service set: service-set-manual-BiEspshades
  Rule: rule-manual-SA-BiEspshades, Term: term-manual-SA-BiEspshades,
  Tunnel index: 1
  Local gateway: 10.1.15.1, Remote gateway: 10.1.15.2
  Local identity: ipv4_subnet(any:0,[0..7]=10.0.0.0/8)
  Remote identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)

    Direction: inbound, SPI: 261, AUX-SPI: 0
    Mode: tunnel, Type: manual, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: des-cbc
    Anti-replay service: Disabled

    Direction: outbound, SPI: 261, AUX-SPI: 0
```

```
    Mode: tunnel, Type: manual, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: des-cbc
    Anti-replay service: Disabled
```

To verify that traffic is traveling over the bidirectional IPSec tunnel, issue the show services ipsec-vpn statistics command:

```
user@R2> show services ipsec-vpn ipsec statistics

PIC: sp-1/2/0, Service set: service-set-manual-BiEspshades

ESP Statistics:
  Encrypted bytes:             1616
  Decrypted bytes:             1560
  Encrypted packets:             20
  Decrypted packets:             19
AH Statistics:
  Input bytes:                    0
  Output bytes:                   0
  Input packets:                  0
  Output packets:                 0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
```

Router 3

To verify that the IPSec security association is active, issue the show services ipsec-vpn ipsec security-associations detail command. To be successful, the SA on Router 3 must contain the same settings you specified on Router 2.

```
user@R3> show services ipsec-vpn ipsec security-associations detail
Service set: service-set-manual-BiEspshades
  Rule: rule-manual-SA-BiEspshades, Term: term-manual-SA-BiEspshades,
  Tunnel index: 1
  Local gateway: 10.1.15.2, Remote gateway: 10.1.15.1
  Local identity: ipv4_subnet(any:0,[0..7]=10.0.0.0/8)
  Remote identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)

    Direction: inbound, SPI: 261, AUX-SPI: 0
    Mode: tunnel, Type: manual, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: des-cbc
    Anti-replay service: Disabled

    Direction: outbound, SPI: 261, AUX-SPI: 0
    Mode: tunnel, Type: manual, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: des-cbc
    Anti-replay service: Disabled
```

To verify that traffic is traveling over the bidirectional IPSec tunnel, issue the show services ipsec-vpn statistics command:

```
user@R3> show services ipsec-vpn ipsec statistics
PIC: sp-1/2/0, Service set: service-set-manual-BiEspshades
ESP Statistics:
  Encrypted bytes:             1560
```

```
  Decrypted bytes:              1616
  Encrypted packets:              19
  Decrypted packets:              20
AH Statistics:
  Input bytes:                     0
  Output bytes:                    0
  Input packets:                   0
  Output packets:                  0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
```

Example: ES PIC IKE Dynamic SA Configuration

Figure 48: ES PIC IKE Dynamic SA Topology Diagram

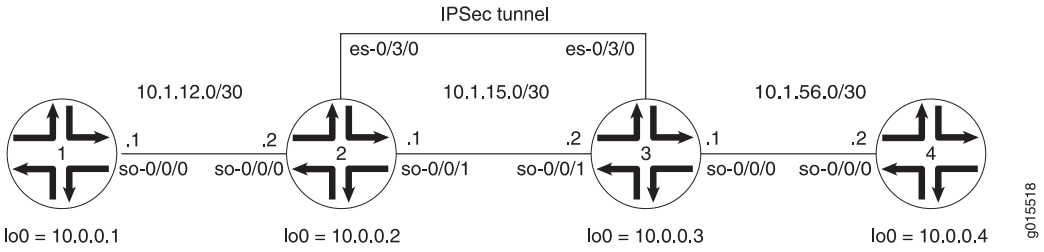


Figure 48 on page 456 shows the same IPSec topology as seen in the ES PIC manual SA example. However, this time the configuration requires Routers 2 and 3 to establish an IPSec tunnel using an IKE dynamic SA, enhanced authentication, and stronger encryption. Routers 1 and 4 continue to provide basic connectivity and are used to verify that the IPSec tunnel is operational.

On Router 1, provide basic OSPF connectivity to Router 2.

Router 1

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R2 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.12.2/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.1/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.1;
}
```

```
protocols {
   ospf {
      area 0.0.0.0 {
         interface so-0/0/0.0;
         interface lo0.0;
      }
   }
}
```

On Router 2, enable OSPF as the underlying routing protocol to connect to Routers 1 and 3. Configure a bidirectional IKE dynamic SA called sa-dynamic at the [edit security ipsec security-association] hierarchy level. For your IKE policy and proposal, use preshared keys for the authentication method, SHA-1 for the authentication algorithm, 3DES-CBC for encryption, group 2 for the Diffie-Hellman group, main mode, 3600 seconds for the lifetime, and a preshared key of juniper for the initial IKE negotiation. For your IPSec policy and proposal, use ESP for the protocol, HMAC-SHA1-96 for authentication, 3DES-CBC for encryption, 28800 seconds for the lifetime, and group 2 for the PFS group.

To direct traffic into the ES PIC and the IPSec tunnel, create two firewall filters. The es-traffic filter matches inbound traffic from Router 1 destined for Router 4, whereas the es-return filter matches the return path from Router 4 to Router 1. Apply the es-traffic filter to the so-0/0/0 interface, and then apply both the es-return filter and the sa-dynamic SA to the es-0/3/0 interface.

Router 2

```
[edit]
interfaces {
   so-0/0/0 {
      description "To R1 so-0/0/0";
      unit 0 {
         family inet {
            filter {
               input es-traffic; # Apply a filter that sends traffic to the IPSec tunnel here.
            }
            address 10.1.12.1/30;
         }
      }
   }
   so-0/0/1 {
      description "To R3 so-0/0/1";
      unit 0 {
         family inet {
            address 10.1.15.1/30;
         }
      }
   }
   es-0/3/0 {
      unit 0 {
         tunnel { # Specify the IPSec tunnel endpoints here.
            source 10.1.15.1;
            destination 10.1.15.2;
         }
         family inet {
            ipsec-sa sa-dynamic; # Apply the dynamic SA here.
            filter {
```

```
                input es-return; # Apply the filter that matches return IPSec traffic here.
            }
          }
        }
      }
      lo0 {
        unit 0 {
          family inet {
            address 10.0.0.2/32;
          }
        }
      }
}
routing-options {
    router-id 10.0.0.2;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface so-0/0/1.0;
            interface lo0.0;
        }
    }
}
security {
    ipsec {
        proposal es-ipsec-proposal { # Define your IPSec proposal specifications here.
            protocol esp;
            authentication-algorithm hmac-sha1-96;
            encryption-algorithm 3des-cbc;
            lifetime-seconds 28800;
        }
        policy es-ipsec-policy { # Define your IPSec policy specifications here.
            perfect-forward-secrecy {
                keys group2;
            }
            proposals es-ipsec-proposal; # Reference the IPSec proposal here.
        }
        security-association sa-dynamic { # Define your dynamic SA here.
            mode tunnel;
            dynamic {
                ipsec-policy es-ipsec-policy; # Reference the IPSec policy here.
            }
        }
    }
    ike {
        proposal es-ike-proposal { # Define your IKE proposal specifications here.
            authentication-method pre-shared-keys;
            dh-group group2;
            authentication-algorithm sha1;
            encryption-algorithm 3des-cbc;
            lifetime-seconds 3600;
        }
        policy 10.1.15.2 { # Define your IKE policy specifications here.
            mode main;
```

```
            proposals es-ike-proposal; # Reference the IKE proposal here.
            pre-shared-key ascii-text "$9$TF6ABIcvWxp0WxNdg4QFn";
            ## The unencrypted preshared key for this example is juniper.
        }
    }
}
firewall {
    filter es-traffic { # Define a filter that sends traffic to the IPSec tunnel here.
        term to-es {
            from {
                source-address {
                    10.1.12.0/24;
                }
                destination-address {
                    10.1.56.0/24;
                }
            }
            then {
                count ipsec-tunnel;
                ipsec-sa sa-dynamic;
            }
        }
        term other {
            then accept;
        }
    }
    filter es-return { # Define a filter that matches return IPSec traffic here.
        term return {
            from {
                source-address {
                    10.1.56.0/24;
                }
                destination-address {
                    10.1.12.0/24;
                }
            }
            then accept;
        }
    }
}
```

On Router 3, enable OSPF as the underlying routing protocol to connect to Routers 2 and 4. Configure a bidirectional IKE dynamic SA called sa-dynamic at the [edit security ipsec security-association] hierarchy level. Use the same policies and proposals that you used on Router 2.

For your IKE policy and proposal, use preshared keys for the authentication method, SHA-1 for the authentication algorithm, 3DES-CBC for encryption, group 2 for the Diffie-Hellman group, main mode, 3600 seconds for the lifetime, and a preshared key of juniper for the initial IKE negotiation. For your IPSec policy and proposal, use ESP for the protocol, HMAC-SHA1-96 for authentication, 3DES-CBC for encryption, 28800 seconds for the lifetime, and group 2 for the PFS group.

To direct traffic into the ES PIC and the IPSec tunnel, create two firewall filters. The es-traffic filter matches inbound traffic from Router 4 destined for Router 1, whereas

the es-return filter matches the return path from Router 1 to Router 4. Apply the es-traffic filter to the so-0/0/0 interface; then apply both the es-return filter and the sa-dynamic SA to the es-0/3/0 interface.

Router 3

```
[edit]
interfaces {
   so-0/0/0 {
      description "To R4 so-0/0/0";
      unit 0 {
         family inet {
            filter {
               input es-traffic; # Apply a filter that sends traffic to the IPSec tunnel here.
            }
            address 10.1.56.1/30;
         }
      }
   }
   so-0/0/1 {
      description "To R2 so-0/0/1";
      unit 0 {
         family inet {
            address 10.1.15.2/30;
         }
      }
   }
   es-0/3/0 {
      unit 0 {
         tunnel { # Specify the IPSec tunnel endpoints here.
            source 10.1.15.2;
            destination 10.1.15.1;
         }
         family inet {
            ipsec-sa sa-dynamic; # Apply the dynamic SA here.
            filter {
               input es-return; # Apply the filter that matches return IPSec traffic here.
            }
         }
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.0.0.3/32;
         }
      }
   }
}
routing-options {
   router-id 10.0.0.3;
}
protocols {
   ospf {
      area 0.0.0.0 {
         interface so-0/0/0.0;
         interface so-0/0/1.0;
```

```
        interface lo0.0;
      }
    }
}
security {
    ipsec {
        proposal es-ipsec-proposal { # Define your IPSec proposal specifications here.
            protocol esp;
            authentication-algorithm hmac-sha1-96;
            encryption-algorithm 3des-cbc;
            lifetime-seconds 28800;
        }
        policy es-ipsec-policy { # Define your IPSec policy specifications here.
            perfect-forward-secrecy {
                keys group2;
            }
            proposals es-ipsec-proposal; # Reference the IPSec proposal here.
        }
        security-association sa-dynamic { # Define your dynamic SA here.
            mode tunnel;
            dynamic {
                ipsec-policy es-ipsec-policy; # Reference the IPSec policy here.
            }
        }
    }
    ike {
        proposal es-ike-proposal { # Define your IKE proposal specifications here.
            authentication-method pre-shared-keys;
            dh-group group2;
            authentication-algorithm sha1;
            encryption-algorithm 3des-cbc;
            lifetime-seconds 3600;
        }
        policy 10.1.15.1 { # Define your IKE policy specifications here.
            mode main;
            proposals es-ike-proposal; # Reference the IKE proposal here.
            pre-shared-key ascii-text "$9$TF6ABIcvWxp0WxNdg4QFn";
            ## The unencrypted preshared key for this example is juniper.
        }
    }
}
firewall {
    filter es-traffic { # Define a filter that sends traffic to the IPSec tunnel here.
        term to-es {
            from {
                source-address {
                    10.1.56.0/24;
                }
                destination-address {
                    10.1.12.0/24;
                }
            }
            then {
                count ipsec-tunnel;
                ipsec-sa sa-dynamic;
            }
```

```
        }
        term other {
            then accept;
        }
    }
    filter es-return { # Define a filter that matches return IPSec traffic here.
        term return {
            from {
                source-address {
                    10.1.12.0/24;
                }
                destination-address {
                    10.1.56.0/24;
                }
            }
            then accept;
        }
    }
}
```

On Router 4, provide basic OSPF connectivity to Router 3.

Router 4

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R3 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.56.2/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.4/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.4;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface lo0.0;
        }
    }
}
```

Verifying Your Work

To verify proper operation of an IKE-based dynamic SA on the ES PIC, use the following commands:

- ping
- show ike security-associations (detail)
- show ipsec security-associations (detail)
- traceroute

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

- Router 1 on page 463
- Router 2 on page 464
- Router 3 on page 465
- Router 4 on page 466

Router 1

On Router 1, issue a ping command to the so-0/0/0 interface of Router 4 to send traffic across the IPSec tunnel.

```
user@R1> ping 10.1.56.2
PING 10.1.56.2 (10.1.56.2): 56 data bytes
64 bytes from 10.1.56.2: icmp_seq=0 ttl=253 time=0.917 ms
64 bytes from 10.1.56.2: icmp_seq=1 ttl=253 time=0.881 ms
64 bytes from 10.1.56.2: icmp_seq=2 ttl=253 time=0.897 ms
64 bytes from 10.1.56.2: icmp_seq=3 ttl=253 time=0.871 ms
64 bytes from 10.1.56.2: icmp_seq=4 ttl=253 time=0.890 ms
64 bytes from 10.1.56.2: icmp_seq=5 ttl=253 time=0.858 ms
64 bytes from 10.1.56.2: icmp_seq=6 ttl=253 time=0.904 ms
^C
--- 10.1.56.2 ping statistics ---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.858/0.888/0.917/0.019 ms
```

You can also issue the traceroute command to verify that traffic to 10.1.56.2 travels over the IPSec tunnel between Router 2 and Router 3. Notice that the second hop does not reference 10.1.15.2—the physical interface on Router 3. Instead, the loopback address of 10.0.0.3 on Router 3 appears as the second hop. This indicates that the IPSec tunnel is operating correctly.

```
user@R1> traceroute 10.1.56.2
traceroute to 10.1.56.2 (10.1.56.2), 30 hops max, 40 byte packets
 1  10.1.12.1 (10.1.12.1)  0.655 ms  0.549 ms  0.508 ms
 2  10.0.0.3 (10.0.0.3)  0.833 ms  0.786 ms  0.757 ms
```

3 10.1.56.2 (10.1.56.2) 0.808 ms 0.741 ms 0.716 ms

Router 2

Another way to verify that matched traffic is being diverted to the bidirectional IPSec tunnel is to view the firewall filter counter. After you issue the ping command from Router 1 (seven packets), the es-traffic firewall filter counter looks like this:

```
user@R2> show firewall filter es-traffic
Filter: es-traffic
Counters:
Name                                                Bytes              Packets
ipsec-tunnel                                          588                    7
```

After you issue the ping command from both Router 1 (seven packets) and Router 4 (five packets), the es-traffic firewall filter counter looks like this:

```
user@R2> show firewall filter es-traffic
Filter: es-traffic
Counters:
Name                                                Bytes              Packets
ipsec-tunnel                                         1008                   12
```

To verify that the IKE SA negotiation between Routers 2 and 3 is successful, issue the show ike security-associations detail command. Notice that the SA contains the settings you specified, such as SHA-1 for the authentication algorithm and 3DES-CBC for the encryption algorithm.

```
user@R2> show ike security-associations detail
IKE peer 10.1.15.2
  Role: Initiator, State: Matured
  Initiator cookie: b5dbdfe2f9000000, Responder cookie: a24c868410000041
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 10.1.15.1:500, Remote: 10.1.15.2:500
  Lifetime: Expires in 401 seconds
  Algorithms:
   Authentication        : sha1
   Encryption            : 3des-cbc
   Pseudo random function: hmac-sha1
  Traffic statistics:
   Input  bytes  :                 1736
   Output bytes  :                 2652
   Input  packets:                    9
   Output packets:                   15
  Flags: Caller notification sent
  IPSec security associations: 3 created, 0 deleted
  Phase 2 negotiations in progress: 0
```

To verify that the IPSec security association is active, issue the show ipsec security-associations detail command. Notice that the SA contains the settings you specified, such as ESP for the protocol, HMAC-SHA1-96 for the authentication algorithm, and 3DES-CBC for the encryption algorithm.

```
user@R2> show ipsec security-associations detail
Security association: sa-dynamic, Interface family: Up
  Local gateway: 10.1.15.1, Remote gateway: 10.1.15.2
```

```
  Local identity: ipv4_subnet(any:0,[0..7]=10.1.12.0/24)
  Remote identity: ipv4_subnet(any:0,[0..7]=10.1.56.0/24)
    Direction: inbound, SPI: 2133029543, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 26212 seconds
    Hard lifetime: Expires in 26347 seconds
    Anti-replay service: Disabled
    Direction: outbound, SPI: 1759450863, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 26212 seconds
    Hard lifetime: Expires in 26347 seconds
    Anti-replay service: Disabled
```

Router 3

View the firewall filter counter to continue verifying that matched traffic is being diverted to the bidirectional IPSec tunnel. After you issue the ping command from Router 1 (seven packets), the es-traffic firewall filter counter looks like this:

```
user@R3> show firewall filter es-traffic
Filter: es-traffic
Counters:
Name                                                Bytes              Packets
ipsec-tunnel                                          588                    7
```

After you issue the ping command from both Router 1 (seven packets) and Router 4 (five packets), the es-traffic firewall filter counter looks like this:

```
user@R3> show firewall filter es-traffic
Filter: es-traffic
Counters:
Name                                                Bytes              Packets
ipsec-tunnel                                         1008                   12
```

To verify the success of the IKE security association, issue the show ike security-associations detail command. Notice that the SA on Router 3 contains the same settings you specified on Router 2.

```
user@R3> show ike security-associations detail
IKE peer 10.1.15.1
  Role: Responder, State: Matured
  Initiator cookie: b5dbdfe2f9000000, Responder cookie: a24c868410000041
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 10.1.15.2:500, Remote: 10.1.15.1:500
  Lifetime: Expires in 564 seconds
  Algorithms:
   Authentication        : sha1
   Encryption            : 3des-cbc
   Pseudo random function: hmac-sha1
  Traffic statistics:
   Input  bytes  :                 2652
   Output bytes  :                 1856
   Input  packets:                   15
   Output packets:                   10
```

```
  Flags: Caller notification sent
  IPSec security associations: 3 created, 4 deleted
  Phase 2 negotiations in progress: 0
```

To verify that the IPSec security association is active, issue the show ipsec security-associations detail command. Notice that the SA on Router 3 contains the same settings you specified on Router 2.

```
user@R3> show ipsec security-associations detail
Security association: sa-dynamic, Interface family: Up
  Local gateway: 10.1.15.2, Remote gateway: 10.1.15.1
  Local identity: ipv4_subnet(any:0,[0..7]=10.1.56.0/24)
  Remote identity: ipv4_subnet(any:0,[0..7]=10.1.12.0/24)
    Direction: inbound, SPI: 1759450863, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 26427 seconds
    Hard lifetime: Expires in 26517 seconds
    Anti-replay service: Disabled
    Direction: outbound, SPI: 2133029543, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 26427 seconds
    Hard lifetime: Expires in 26517 seconds
    Anti-replay service: Disabled
```

Router 4

On Router 4, issue a ping command to the so-0/0/0 interface of Router 1 to send traffic across the IPSec tunnel.

```
user@R4> ping 10.1.12.2
user@R4> ping 10.1.12.2
PING 10.1.12.2 (10.1.12.2): 56 data bytes
64 bytes from 10.1.12.2: icmp_seq=0 ttl=253 time=13.528 ms
64 bytes from 10.1.12.2: icmp_seq=1 ttl=253 time=0.873 ms
64 bytes from 10.1.12.2: icmp_seq=2 ttl=253 time=32.145 ms
64 bytes from 10.1.12.2: icmp_seq=3 ttl=253 time=0.921 ms
64 bytes from 10.1.12.2: icmp_seq=4 ttl=253 time=0.899 ms
^C
--- 10.1.12.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.873/9.673/32.145/12.255 ms
```

You can also issue the traceroute command to verify that traffic to 10.1.12.2 travels over the IPSec tunnel between Router 3 and Router 2. Notice that the second hop does not reference 10.1.15.1—the physical interface on Router 2. Instead, the loopback address of 10.0.0.2 on Router 2 appears as the second hop. This indicates that the IPSec tunnel is operating correctly.

```
user@R4> traceroute 10.1.12.2
traceroute to 10.1.12.2 (10.1.12.2), 30 hops max, 40 byte packets
 1  10.1.56.1 (10.1.56.1)  0.681 ms  0.624 ms  0.547 ms
 2  10.0.0.2 (10.0.0.2)  0.800 ms  0.770 ms  0.737 ms
 3  10.1.12.2 (10.1.12.2)  0.793 ms  0.742 ms  0.716 ms
```

Example: AS PIC IKE Dynamic SA Configuration

Figure 49: AS PIC IKE Dynamic SA Topology Diagram

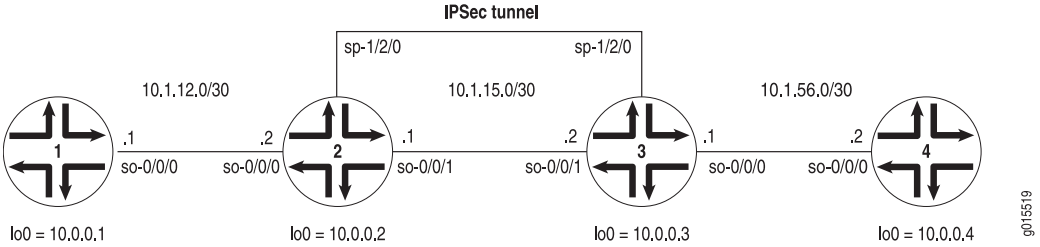


Figure 49 on page 467 shows the same IPSec topology as seen in the AS PIC manual SA example. However, this configuration requires Routers 2 and 3 to establish an IPSec tunnel using an IKE dynamic SA, enhanced authentication, and stronger encryption. Routers 1 and 4 continue to provide basic connectivity and are used to verify that the IPSec tunnel is operational.

NOTE: When you do not specify an IKE proposal, an IPSec proposal, and an IPSec policy on an AS PIC, the JUNOS software defaults to the highest level of encryption and authentication. As a result, the default authentication protocol is ESP, the default authentication mode is HMAC-SHA1-96, and the default encryption mode is 3DES-CBC. For more information about default IKE and IPSec policies and proposals on the AS PIC, see Table 42 on page 417.

On Router 1, provide basic OSPF connectivity to Router 2.

Router 1

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R2 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.12.2/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.1/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.1;
}
protocols {
    ospf {
```

```
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface lo0.0;
        }
    }
}
```

On Router 2, enable OSPF as the underlying routing protocol to connect to Routers 1 and 3. Configure a bidirectional IKE dynamic SA in a rule called rule-ike at the [edit ipsec-vpn rule] hierarchy level. Reference this rule in a service set called service-set-dynamic-BiEspsha3des at the [edit services service-set] hierarchy level.

Using default values in the AS PIC, you do not need to specify an IPSec proposal, IPSec policy, or IKE proposal. However, you do need to configure a preshared key in an IKE policy with the pre-shared-key statement at the [edit services ipsec-vpn ike policy *policy-name*] hierarchy level. (For more information about default IKE and IPSec policies and proposals on the AS PIC, see Table 42 on page 417.)

To direct traffic into the AS PIC and the IPSec tunnel, configure a next-hop style service set and add the adaptive services logical interface used as the IPSec inside interface into the OSPF configuration.

Router 2

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R1 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.12.1/30;
            }
        }
    }
    so-0/0/1 {
        description "To R3 so-0/0/1";
        unit 0 {
            family inet {
                address 10.1.15.1/30;
            }
        }
    }
    sp-1/2/0 {
        services-options {
            syslog {
                host local {
                    services info;
                }
            }
        }
        unit 0 {
            family inet {
            }
            unit 1 { # sp-1/2/0.1 is the IPSec inside interface.
                family inet;
                service-domain inside;
            }
```

```
        unit 2 { # sp-1/2/0.2 is the IPSec outside interface.
          family inet;
          service-domain outside;
        }
      }
      lo0 {
        unit 0 {
          family inet {
            address 10.0.0.2/32;
          }
        }
      }
}
routing-options {
    router-id 10.0.0.2;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface lo0.0;
            interface sp-1/2/0.1; # This sends OSPF traffic over the IPSec tunnel.
        }
    }
}
services {
    service-set service-set-dynamic-BiEspsha3des { # Define your service set here.
        next-hop-service { # Required for dynamic routing protocols such as OSPF.
            inside-service-interface sp-1/2/0.1;
            outside-service-interface sp-1/2/0.2;
        }
        ipsec-vpn-options {
            local-gateway 10.1.15.1; # Specify the local IP address of the IPSec tunnel.
        }
        ipsec-vpn-rules rule-ike; # Reference your IPSec VPN rule here.
    }
    ipsec-vpn {
        rule rule-ike { # Define your IPSec VPN rule here.
            term term-ike {
                then {
                    remote-gateway 10.1.15.2; # The remote IP address of the IPSec tunnel.
                    dynamic { # This creates a dynamic SA.
                        ike-policy ike-policy-preshared; # Reference your IKE policy here.
                    }
                }
            }
            match-direction input; # Specify in which direction the rule should match.
        }
        ike {
            policy ike-policy-preshared { # Define your IKE policy specifications here.
                pre-shared-key ascii-text "$9$KtKWX-YgJHqfVwqfTzCAvWL";
                ## The unencrypted preshared key for this example is juniper.
            }
        }
    }
}
```

```
}
```

On Router 3, enable OSPF as the underlying routing protocol to connect to Routers 2 and 4. Configure a bidirectional IKE dynamic SA in a rule called rule-ike at the [edit ipsec-vpn rule] hierarchy level. Reference this rule in a service set called service-set-dynamic-BiEspsha3des at the [edit services service-set] hierarchy level.

Again, use the same default policies and proposals that you used on Router 2. However, remember to configure a preshared key in an IKE policy with the pre-shared-key statement at the [edit services ipsec-vpn ike policy *policy-name*] hierarchy level. The key must match the one you specified on Router 2. (For more information about default IKE and IPSec policies and proposals on the AS PIC, see Table 42 on page 417.)

To direct traffic into the AS PIC and the IPSec tunnel, configure a next-hop style service set and add the adaptive services logical interface used as the IPSec inside interface into the OSPF configuration.

Router 3

```
[edit]
interfaces {
   so-0/0/0 {
      description "To R4 so-0/0/0";
      unit 0 {
         family inet {
            address 10.1.56.1/30;
         }
      }
   }
   so-0/0/1 {
      description "To R2 so-0/0/1";
      unit 0 {
         family inet {
            address 10.1.15.2/30;
         }
      }
   }
   sp-1/2/0 {
      services-options {
         syslog {
            host local {
               services info;
            }
         }
      }
      unit 0 {
         family inet {
         }
         unit 1 { # sp-1/2/0.1 is the IPSec inside interface.
            family inet;
            service-domain inside;
         }
         unit 2 { # sp-1/2/0.2 is the IPSec outside interface.
            family inet;
            service-domain outside;
         }
```

```
        }
        lo0 {
            unit 0 {
                family inet {
                    address 10.0.0.3/32;
                }
            }
        }
    }
    routing-options {
        router-id 10.0.0.3;
    }
    protocols {
        ospf {
            area 0.0.0.0 {
                interface so-0/0/0.0;
                interface lo0.0;
                interface sp-1/2/0.1; # This sends OSPF traffic over the IPSec tunnel.
            }
        }
    }
    services {
        service-set service-set-dynamic-BiEspsha3des { # Define your service set here.
            next-hop-service { # Required for dynamic routing protocols such as OSPF.
                inside-service-interface sp-1/2/0.1;
                outside-service-interface sp-1/2/0.2;
            }
            ipsec-vpn-options {
                local-gateway 10.1.15.2; # Specify the local IP address of the IPSec tunnel.
            }
            ipsec-vpn-rules rule-ike; # Reference your IPSec VPN rule here.
        }
        ipsec-vpn {
            rule rule-ike { # Define your IPSec VPN rule here.
                term term-ike {
                    then {
                        remote-gateway 10.1.15.1; # The remote IP address of the IPSec tunnel.
                        dynamic { # This creates a dynamic SA.
                            ike-policy ike-policy-preshared; # Reference your IKE policy here.
                        }
                    }
                }
                match-direction input; # Specify in which direction the rule should match.
            }
            ike {
                policy ike-policy-preshared { # Define your IKE policy specifications here.
                    pre-shared-key ascii-text "$9$KtKWX-YgJHqfVwqfTzCAvWL";
                    ## The unencrypted preshared key for this example is juniper.
                }
            }
        }
    }
}
```

On Router 4, provide basic OSPF connectivity to Router 3.

Router 4

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R3 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.56.2/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.4/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.4;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface lo0.0;
        }
    }
}
```

Verifying Your Work

To verify proper operation of an IKE-based dynamic SA on the AS PIC, use the following commands:

- ping
- show services ipsec-vpn ike security-associations (detail)
- show services ipsec-vpn ipsec security-associations (detail)
- show services ipsec-vpn ipsec statistics
- traceroute

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

- Router 1 on page 473
- Router 2 on page 473
- Router 3 on page 474
- Router 4 on page 475

Router 1

On Router 1, issue a ping command to the so-0/0/0 interface on Router 4 to send traffic across the IPSec tunnel.

```
user@R1> ping 10.1.56.2
PING 10.1.56.2 (10.1.56.2): 56 data bytes
64 bytes from 10.1.56.2: icmp_seq=0 ttl=254 time=1.351 ms
64 bytes from 10.1.56.2: icmp_seq=1 ttl=254 time=1.187 ms
64 bytes from 10.1.56.2: icmp_seq=2 ttl=254 time=1.172 ms
64 bytes from 10.1.56.2: icmp_seq=3 ttl=254 time=1.154 ms
64 bytes from 10.1.56.2: icmp_seq=4 ttl=254 time=1.156 ms
^C
--- 10.1.56.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.154/1.204/1.351/0.074 ms
```

Router 2

To verify that the IKE SA negotiation is successful, issue the show services ipsec-vpn ike security-associations command.

```
user@R2> show services ipsec-vpn ike security-associations
Remote Address  State         Initiator cookie  Responder cookie  Exchange type
10.1.15.2       Matured       03075bd3a0000003  4bff26a5c7000003  Main
```

To verify that the IPSec security association is active, issue the show services ipsec-vpn ipsec security-associations detail command. Notice that the SA contains the default settings inherent in the AS PIC, such as ESP for the protocol and HMAC-SHA1-96 for the authentication algorithm.

```
user@R2> show services ipsec-vpn ipsec security-associations detail
Service set: service-set-dynamic-BiEspsha3des
  Rule: rule-ike, Term: term-ike, Tunnel index: 1
  Local gateway: 10.1.15.1, Remote gateway: 10.1.15.2
  Local identity: ipv4_subnet(any:0,[0..7]=10.1.12.0/24)
  Remote identity: ipv4_subnet(any:0,[0..7]=10.1.56.0/24)
    Direction: inbound, SPI: 2666326758, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 26863 seconds
    Hard lifetime: Expires in 26998 seconds
    Anti-replay service: Enabled, Replay window size: 64
    Direction: outbound, SPI: 684772754, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 26863 seconds
    Hard lifetime: Expires in 26998 seconds
    Anti-replay service: Enabled, Replay window size: 64
```

To verify that traffic is traveling over the bidirectional IPSec tunnel, issue the show services ipsec-vpn statistics command:

```
user@R2> show services ipsec-vpn ipsec statistics
```

```
PIC: sp-1/2/0, Service set: service-set-dynamic-BiEspsha3des
ESP Statistics:
  Encrypted bytes:             2248
  Decrypted bytes:             2120
  Encrypted packets:             27
  Decrypted packets:             25
AH Statistics:
  Input bytes:                    0
  Output bytes:                   0
  Input packets:                  0
  Output packets:                 0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
```

Bad headers: 0, Bad trailers: 0

Router 3

To verify that the IKE SA negotiation is successful, issue the show services ipsec-vpn ike security-associations command. To be successful, the SA on Router 3 must contain the same settings you specified on Router 2.

```
user@R3> show services ipsec-vpn ike security-associations
Remote Address  State         Initiator cookie  Responder cookie  Exchange type
10.1.15.1       Matured       03075bd3a0000003  4bff26a5c7000003  Main
```

To verify that the IPSec SA is active, issue the show services ipsec-vpn ipsec security-associations detail command. To be successful, the SA on Router 3 must contain the same settings you specified on Router 2.

```
user@R3> show services ipsec-vpn ipsec security-associations detail
Service set: service-set-dynamic-BiEspsha3des
  Rule: rule-ike, Term: term-ike, Tunnel index: 1
  Local gateway: 10.1.15.2, Remote gateway: 10.1.15.1
  Local identity: ipv4_subnet(any:0,[0..7]=10.1.56.0/24)
  Remote identity: ipv4_subnet(any:0,[0..7]=10.1.12.0/24)
    Direction: inbound, SPI: 684772754, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 26598 seconds
    Hard lifetime: Expires in 26688 seconds
    Anti-replay service: Enabled, Replay window size: 64
    Direction: outbound, SPI: 2666326758, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 26598 seconds
    Hard lifetime: Expires in 26688 seconds
    Anti-replay service: Enabled, Replay window size: 64
```

To verify that traffic is traveling over the bidirectional IPSec tunnel, issue the show services ipsec-vpn statistics command:

```
user@R3> show services ipsec-vpn ipsec statistics
PIC: sp-1/2/0, Service set: service-set-dynamic-BiEspsha3des
ESP Statistics:
```

```
  Encrypted bytes:             2120
  Decrypted bytes:             2248
  Encrypted packets:             25
  Decrypted packets:             27
AH Statistics:
  Input bytes:                    0
  Output bytes:                   0
  Input packets:                  0
  Output packets:                 0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
```

Bad headers: 0, Bad trailers: 0

Router 4

On Router 4, issue a ping command to the so-0/0/0 interface on Router 1 to send traffic across the IPSec tunnel.

```
user@R4> ping 10.1.12.2
PING 10.1.12.2 (10.1.12.2): 56 data bytes
64 bytes from 10.1.12.2: icmp_seq=0 ttl=254 time=1.350 ms
64 bytes from 10.1.12.2: icmp_seq=1 ttl=254 time=1.161 ms
64 bytes from 10.1.12.2: icmp_seq=2 ttl=254 time=1.124 ms
64 bytes from 10.1.12.2: icmp_seq=3 ttl=254 time=1.142 ms
64 bytes from 10.1.12.2: icmp_seq=4 ttl=254 time=1.139 ms
64 bytes from 10.1.12.2: icmp_seq=5 ttl=254 time=1.116 ms
^C
--- 10.1.12.2 ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.116/1.172/1.350/0.081 ms
```

The final way you can confirm that traffic travels over the IPSec tunnel is by issuing the traceroute command to the so-0/0/0 interface on Router 1. Notice that the physical interface between Routers 2 and 3 is not referenced in the path; traffic enters the IPSec tunnel through the adaptive services IPSec inside interface on Router 3, passes through the loopback interface on Router 2, and ends at the so-0/0/0 interface on Router 1.

```
user@R4> traceroute 10.1.12.2
traceroute to 10.1.12.2 (10.1.12.2), 30 hops max, 40 byte packets
 1  10.1.15.2 (10.1.15.2)  0.987 ms  0.630 ms  0.563 ms
 2  10.0.0.2 (10.0.0.2)  1.194 ms  1.058 ms  1.033 ms
 3  10.1.12.2 (10.1.12.2)  1.073 ms  0.949 ms  0.932 ms
```

Example: IKE Dynamic SA Between an AS PIC and an ES PIC Configuration

Figure 50: AS PIC to ES PIC IKE Dynamic SA Topology Diagram

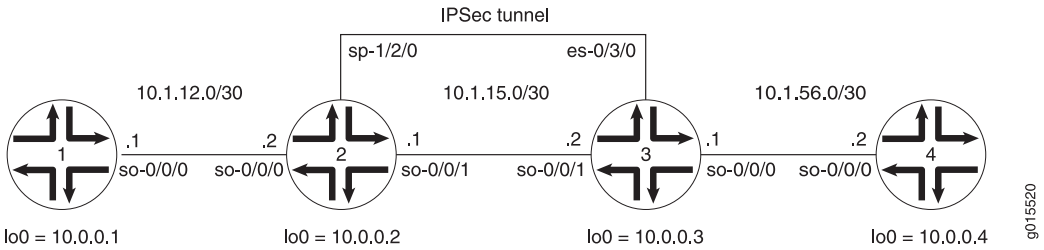


Figure 50 on page 476 shows a hybrid configuration that allows you to create an IPSec tunnel between the AS PIC and the ES PIC. Router 2 contains an AS PIC at sp-1/2/0 and Router 3 has an ES PIC at es-0/3/0. To establish an IPSec tunnel using an IKE dynamic SA, the key is to learn the default IKE SA and IPSec SA settings built into the AS PIC and configure them explicitly on the ES PIC. Routers 1 and 4 again provide basic connectivity and are used to verify that the IPSec tunnel is operational.

On Router 1, provide basic OSPF connectivity to Router 2.

Router 1

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R2 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.12.2/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.1/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.1;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface lo0.0;
        }
    }
}
```

On Router 2, enable OSPF as the underlying routing protocol to connect to Routers 1 and 3. Configure a bidirectional IKE dynamic SA in a rule called rule-ike at the [edit ipsec-vpn rule] hierarchy level. Reference this rule in a service set called service-set-dynamic-BiEspsha3des at the [edit services service-set] hierarchy level.

Using default values in the AS PIC, you do not need to specify an IPSec proposal, IPSec policy, or IKE proposal. However, you do need to configure a preshared key in an IKE policy with the pre-shared-key statement at the [edit services ipsec-vpn ike policy *policy-name*] hierarchy level. (For more information about default IKE and IPSec policies and proposals on the AS PIC, see Table 42 on page 417.)

To direct traffic into the AS PIC and the IPSec tunnel, include match conditions in the rule-ike IPSec VPN rule to match inbound traffic from Router 1 that is destined for Router 4. Because the rule is already referenced by the service set, apply the service set to the so-0/0/1 interface. To count the amount of traffic that enters the IPSec tunnel, configure a firewall filter called ipsec-tunnel and apply it to the sp-1/2/0 interface.

Router 2

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R1 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.12.1/30;
            }
        }
    }
    so-0/0/1 {
        description "To R3 so-0/0/1";
        unit 0 {
            family inet {
                service { # Apply the service set here.
                    input {
                        service-set service-set-dynamic-BiEspsha3des;
                    }
                    output {
                        service-set service-set-dynamic-BiEspsha3des;
                    }
                }
                address 10.1.15.1/30;
            }
        }
    }
    sp-1/2/0 {
        services-options {
            syslog {
                host local {
                    services info;
                }
            }
        }
        unit 0 {
            family inet {
                filter {
```

```
                input ipsec-tunnel; # Apply the firewall filter with the counter here.
            }
        }
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.0.0.2/32;
      }
    }
  }
}
routing-options {
  router-id 10.0.0.2;
}
protocols {
  ospf {
    area 0.0.0.0 {
      interface so-0/0/0.0;
      interface so-0/0/1.0;
      interface lo0.0;
    }
  }
}
firewall {
  filter ipsec-tunnel { # Configure a firewall filter to count IPSec traffic here.
    term 1 {
      then {
        count ipsec-tunnel;
        accept;
      }
    }
  }
}
services {
  service-set service-set-dynamic-BiEspsha3des { # Define your service set here.
    interface-service {
      service-interface sp-1/2/0; # Specify an interface to process IPSec.
    }
    ipsec-vpn-options {
      local-gateway 10.1.15.1; # Specify the local IP address of the IPSec tunnel.
    }
    ipsec-vpn-rules rule-ike; # Reference your IPSec VPN rule here.
  }
  ipsec-vpn {
    rule rule-ike { # Define your IPSec VPN rule here.
      term term-ike {
        from {
          source-address {
            10.1.12.0/24;
          }
          destination-address {
            10.1.56.0/24;
          }
        }
```

```
            then {
               remote-gateway 10.1.15.2; # The remote IP address of the IPSec tunnel.
               dynamic { # This creates a dynamic SA.
                  ike-policy ike-policy-preshared; # Reference your IKE proposal here.
               }
            }
         }
         match-direction output; # Specify in which direction the rule should match.
      }
      ike {
         policy ike-policy-preshared { # Define your IKE policy specifications here.
            pre-shared-key ascii-text "$9$KtKWX-YgJHqfVwqfTzCAvWL";
            ## The unencrypted preshared key for this example is juniper.
         }
      }
   }
}
```

On Router 3, enable OSPF as the underlying routing protocol to connect to Routers 2 and 4. Configure a bidirectional IKE dynamic SA called sa-dynamic at the [edit security ipsec security-association] hierarchy level. To allow the ES PIC to communicate with the IKE dynamic SA established on Router 2, you must explicitly configure the same policies and proposals on the ES PIC that are available by default on the AS PIC. (For more information about default IKE and IPSec policies and proposals on the AS PIC, see Table 42 on page 417.)

For your IKE policy and proposal, use preshared keys for the authentication method, SHA-1 for the authentication algorithm, 3DES-CBC for encryption, group 2 for the Diffie-Hellman group, main mode, 3600 seconds for the lifetime, and a preshared key of juniper for the initial IKE negotiation. For your IPSec policy and proposal, use ESP for the protocol, HMAC-SHA1-96 for authentication, 3DES-CBC for encryption, 28800 seconds for the lifetime, and group 2 for the PFS group.

To direct traffic into the ES PIC and the IPSec tunnel, create two firewall filters. The es-traffic filter matches inbound traffic from Router 4 destined for Router 1, whereas the es-return filter matches the return path from Router 1 to Router 4. Apply the es-traffic filter to the so-0/0/0 interface; then apply both the es-return filter and the sa-dynamic SA to the es-0/3/0 interface.

Router 3

```
[edit]
interfaces {
   so-0/0/0 {
      description "To R4 so-0/0/0";
      unit 0 {
         family inet {
            filter {
               input es-traffic; # Apply a filter that sends traffic to the IPSec tunnel here.
            }
            address 10.1.56.1/30;
         }
      }
   }
   so-0/0/1 {
      description "To R2 so-0/0/1";
```

```
        unit 0 {
            family inet {
                address 10.1.15.2/30;
            }
        }
    }
    es-0/3/0 {
        unit 0 {
            tunnel { # Specify the IPSec tunnel endpoints here.
                source 10.1.15.2;
                destination 10.1.15.1;
            }
            family inet {
                ipsec-sa sa-dynamic; # Apply the dynamic SA here.
                filter {
                    input es-return; # Apply the filter that matches return IPSec traffic here.
                }
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.3/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.3;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface so-0/0/1.0;
            interface lo0.0;
        }
    }
}
security {
    ipsec {
        proposal es-ipsec-proposal { # Define your IPSec proposal specifications here.
            protocol esp;
            authentication-algorithm hmac-sha1-96;
            encryption-algorithm 3des-cbc;
            lifetime-seconds 28800;
        }
        policy es-ipsec-policy { # Define your IPSec policy specifications here.
            perfect-forward-secrecy {
                keys group2;
            }
            proposals es-ipsec-proposal; # Reference the IPSec proposal here.
        }
        security-association sa-dynamic { # Define your dynamic SA here.
            mode tunnel;
```

```
        dynamic {
            ipsec-policy es-ipsec-policy; # Reference the IPSec policy here.
        }
    }
  }
  ike {
    proposal es-ike-proposal { # Define your IKE proposal specifications here.
      authentication-method pre-shared-keys;
      dh-group group2;
      authentication-algorithm sha1;
      encryption-algorithm 3des-cbc;
      lifetime-seconds 3600;
    }
    policy 10.1.15.1 { # Define your IKE policy specifications here.
      mode main;
      proposals es-ike-proposal; # Reference the IKE proposal here.
      pre-shared-key ascii-text "$9$TF6ABIcvWxp0WxNdg4QFn";
      ## The unencrypted preshared key for this example is juniper.
    }
  }
}
firewall {
  filter es-traffic { # Define a filter that sends traffic to the IPSec tunnel here.
    term to-es {
      from {
        source-address {
          10.1.56.0/24;
        }
        destination-address {
          10.1.12.0/24;
        }
      }
      then {
        count ipsec-tunnel;
        ipsec-sa sa-dynamic;
      }
    }
    term other {
      then accept;
    }
  }
  filter es-return { # Define a filter that matches return IPSec traffic here.
    term return {
      from {
        source-address {
          10.1.12.0/24;
        }
        destination-address {
          10.1.56.0/24;
        }
      }
      then accept;
    }
  }
}
```

On Router 4, provide basic OSPF connectivity to Router 3.

Router 4

```
[edit]
interfaces {
   so-0/0/0 {
      description "To R3 so-0/0/0";
      unit 0 {
         family inet {
            address 10.1.56.2/30;
         }
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.0.0.4/32;
         }
      }
   }
}
routing-options {
   router-id 10.0.0.4;
}
protocols {
   ospf {
      area 0.0.0.0 {
         interface so-0/0/0.0;
         interface lo0.0;
      }
   }
}
```

Verifying Your Work

To verify proper operation of an IKE-based dynamic SA on the AS PIC, use the following commands:

- ping
- show services ipsec-vpn ike security-associations (detail)
- show services ipsec-vpn ipsec security-associations (detail)
- traceroute

To verify proper operation of an IKE-based dynamic SA on the ES PIC, use the following commands:

- ping
- show ike security-associations (detail)
- show ipsec security-associations (detail)
- traceroute

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

Router 1

On Router 1, issue a ping command to the so-0/0/0 interface of Router 4 to send traffic across the IPSec tunnel.

```
user@R1> ping 10.1.56.2
PING 10.1.56.2 (10.1.56.2): 56 data bytes
64 bytes from 10.1.56.2: icmp_seq=0 ttl=253 time=1.172 ms
64 bytes from 10.1.56.2: icmp_seq=1 ttl=253 time=1.020 ms
64 bytes from 10.1.56.2: icmp_seq=2 ttl=253 time=0.998 ms
64 bytes from 10.1.56.2: icmp_seq=3 ttl=253 time=1.037 ms
^C
--- 10.1.56.2 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.998/1.057/1.172/0.068 ms
```

You can also issue the traceroute command to verify that traffic to 10.1.56.2 travels over the IPSec tunnel between Router 2 and Router 3. Notice that the traced path does not reference 10.1.15.2—the physical interface on Router 3. Instead, traffic arriving at Router 2 is immediately filtered into the IPSec tunnel and the path is listed as unknown with the *** notation. This indicates that the IPSec tunnel is operating correctly.

```
user@R1> traceroute 10.1.56.2
traceroute to 10.1.56.2 (10.1.56.2), 30 hops max, 40 byte packets
 1  * * *
 2  10.1.56.2 (10.1.56.2)  1.045 ms  0.915 ms  0.850 ms
```

Router 2

One way to verify that matched traffic is being diverted to the bidirectional IPSec tunnel is to view the firewall filter counter. Before any traffic flows, the ipsec-tunnel firewall filter counter looks like this:

```
user@R2> show firewall filter ipsec-tunnel
Filter: ipsec-tunnel
Counters:
Name                                                Bytes              Packets
ipsec-tunnel                                            0                    0
```

After you issue the ping command from Router 1 (four packets) to 10.1.56.2, the ipsec-tunnel firewall filter counter looks like this:

```
user@R2> show firewall filter ipsec-tunnel
```

```
Filter: ipsec-tunnel
Counters:
Name                                                Bytes              Packets
ipsec-tunnel                                          336                    4
```

After you issue the ping command from both Router 1 to 10.1.56.2 (four packets) and from Router 4 to 10.1.12.2 (six packets), the ipsec-tunnel firewall filter counter looks like this:

```
user@R2> show firewall filter ipsec-tunnel
Filter: es-traffic
Counters:
Name                                                Bytes              Packets
ipsec-tunnel                                          840                   10
```

To verify that the IKE SA negotiation is successful, issue the show services ipsec-vpn ike security-associations detail command. Notice that the SA contains the default IKE settings inherent in the AS PIC, such as SHA-1 for the authentication algorithm and 3DES-CBC for the encryption algorithm.

```
user@R2> show services ipsec-vpn ike security-associations detail
IKE peer 10.1.15.2
  Role: Responder, State: Matured
  Initiator cookie: c8e1e4c0da000040, Responder cookie: 4fbaa5184e000044
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 10.1.15.1:500, Remote: 10.1.15.2:500
  Lifetime: Expires in 3535 seconds
  Algorithms:
   Authentication        : sha1
   Encryption            : 3des-cbc
   Pseudo random function: hmac-sha1
  Traffic statistics:
   Input  bytes  :                  840
   Output bytes  :                  756
   Input  packets:                    5
   Output packets:                    4
  Flags: Caller notification sent
  IPSec security associations: 1 created, 0 deleted
  Phase 2 negotiations in progress: 0
```

To verify that the IPSec security association is active, issue the show services ipsec-vpn ipsec security-associations detail command. Notice that the SA contains the default settings inherent in the AS PIC, such as ESP for the protocol and HMAC-SHA1-96 for the authentication algorithm.

```
user@R2> show services ipsec-vpn ipsec security-associations detail
Service set: service-set-dynamic-BiEspsha3des
  Rule: rule-ike, Term: term-ike, Tunnel index: 1
  Local gateway: 10.1.15.1, Remote gateway: 10.1.15.2
  Local identity: ipv4_subnet(any:0,[0..7]=10.1.12.0/24)
  Remote identity: ipv4_subnet(any:0,[0..7]=10.1.56.0/24)
    Direction: inbound, SPI: 407204513, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 24546 seconds
    Hard lifetime: Expires in 24636 seconds
    Anti-replay service: Disabled
```

```
    Direction: outbound, SPI: 2957235894, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 24546 seconds
    Hard lifetime: Expires in 24636 seconds
    Anti-replay service: Disabled
```

Router 3

View the firewall filter counter to continue verifying that matched traffic is being diverted to the bidirectional IPSec tunnel. After you issue the ping command from Router 1 (four packets), the es-traffic firewall filter counter looks like this:

```
user@R3> show firewall filter es-traffic
Filter: es-traffic
Counters:
Name                                                Bytes              Packets
ipsec-tunnel                                          336                    4
```

After you issue the ping command from both Router 1 (four packets) and Router 4 (six packets), the es-traffic firewall filter counter looks like this:

```
user@R3> show firewall filter es-traffic
Filter: es-traffic
Counters:
Name                                                Bytes              Packets
ipsec-tunnel                                          840                   10
```

To verify the success of the IKE security association on the ES PIC, issue the show ike security-associations detail command. Notice that the IKE SA on Router 3 contains the same settings you specified on Router 2.

```
user@R3> show ike security-associations detail
IKE peer 10.1.15.1
  Role: Initiator, State: Matured
  Initiator cookie: c8e1e4c0da000040, Responder cookie: 4fbaa5184e000044
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 10.1.15.2:500, Remote: 10.1.15.1:500
  Lifetime: Expires in 3441 seconds
  Algorithms:
   Authentication        : sha1
   Encryption            : 3des-cbc
   Pseudo random function: hmac-sha1
  Traffic statistics:
   Input  bytes  :                  756
   Output bytes  :                  840
   Input  packets:                    4
   Output packets:                    5
  Flags: Caller notification sent
  IPSec security associations: 1 created, 0 deleted
  Phase 2 negotiations in progress: 0
```

To verify that the IPSec security association is active, issue the show ipsec security-associations detail command. Notice that the IPSec SA on Router 3 contains the same settings you specified on Router 2.

```
user@R3> show ipsec security-associations detail
Security association: sa-dynamic, Interface family: Up
  Local gateway: 10.1.15.2, Remote gateway: 10.1.15.1
  Local identity: ipv4_subnet(any:0,[0..7]=10.1.56.0/24)
  Remote identity: ipv4_subnet(any:0,[0..7]=10.1.12.0/24)
    Direction: inbound, SPI: 2957235894, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 28555 seconds
    Hard lifetime: Expires in 28690 seconds
    Anti-replay service: Disabled
    Direction: outbound, SPI: 407204513, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 28555 seconds
    Hard lifetime: Expires in 28690 seconds
    Anti-replay service: Disabled
```

Router 4

On Router 4, issue a ping command to the so-0/0/0 interface on Router 1 to send traffic across the IPSec tunnel.

```
user@R4> ping 10.1.12.2
PING 10.1.12.2 (10.1.12.2): 56 data bytes
64 bytes from 10.1.12.2: icmp_seq=0 ttl=254 time=1.350 ms
64 bytes from 10.1.12.2: icmp_seq=1 ttl=254 time=1.161 ms
64 bytes from 10.1.12.2: icmp_seq=2 ttl=254 time=1.124 ms
64 bytes from 10.1.12.2: icmp_seq=3 ttl=254 time=1.142 ms
64 bytes from 10.1.12.2: icmp_seq=4 ttl=254 time=1.139 ms
64 bytes from 10.1.12.2: icmp_seq=5 ttl=254 time=1.116 ms
^C
--- 10.1.12.2 ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.116/1.172/1.350/0.081 ms
```

Again, the traceroute command verifies that traffic to 10.1.12.2 travels over the IPSec tunnel between Router 3 and Router 2. Notice that the second hop does not reference 10.1.15.1—the physical interface on Router 2. Instead, the second hop is listed as unknown with the *** notation. This indicates that the IPSec tunnel is operating correctly.

```
user@R4> traceroute 10.1.12.2
traceroute to 10.1.12.2 (10.1.12.2), 30 hops max, 40 byte packets
 1  10.1.56.1 (10.1.56.1)  3.561 ms  0.613 ms  0.558 ms
 2  * * *
 3  10.1.12.2 (10.1.12.2)  1.073 ms  0.862 ms  0.818 ms
```

Example: AS PIC IKE Dynamic SA with Digital Certificates Configuration

Figure 51: AS PIC IKE Dynamic SA Topology Diagram

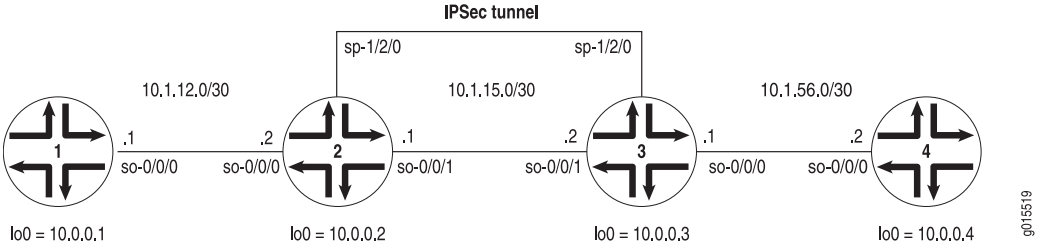


Figure 51 on page 487 shows the same IPSec topology as the AS PIC dynamic SA example on "Example: AS PIC IKE Dynamic SA Configuration" on page 467. However, this configuration requires Routers 2 and 3 to establish an IKE-based IPSec tunnel by using digital certificates in place of preshared keys. Routers 1 and 4 continue to provide basic connectivity and are used to verify that the IPSec tunnel is operational.

On Router 1, provide basic OSPF connectivity to Router 2.

Router 1

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R2 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.12.2/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.1/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.1;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface lo0.0;
        }
    }
}
```

On Router 2, you must request a CA certificate, create a local certificate, and load these digital certificates into the router before you can reference them in your IPSec

configuration. To begin, configure an IPSec profile by specifying the trusted CA and URL of the CA server that handles CA certificate processing:

```
[edit]
security {
   pki {
      ca-profile entrust {
         ca-identity entrust;
         enrollment {
            url http://ca-1.jnpr.net/cgi-bin/pkiclient.exe;
         }
      }
   }
}
```

Certificate revocation list (CRL) verification is enabled by default. You can optionally specify the Lightweight Access Directory (LDAP) server where the CA stores the CRL. The certificate typically includes a certificate distribution point (CDP), which contains information about how to retrieve the CRL for the certificate. The router uses this information to download the CRL automatically. In this example, the LDAP URL is specified, which overrides the location provided in the certificate:

```
[edit]
security pki ca-profile entrust {
   revocation-check {
      crl {
         url ldap://10.157.90.185/o=juniper,c=uscertificateRevocationListbase;
      }
   }
}
```

After you configure the CA profile, you can request a CA certificate from the trusted CA. In this example, the certificate is enrolled online and installed into the routing platform automatically.

```
user@R2> request security pki ca-certificate enroll ca-profile entrust
Received following certificates:
  Certificate: C=us, O=juniper
    Fingerprint: 00:8e:6f:58:dd:68:bf:25:0a:e3:f9:17:70:d6:61:f3:53:a7:79:10
  Certificate: C=us, O=juniper, CN=First Officer
    Fingerprint: bc:78:87:9b:a7:91:13:20:71:db:ac:b5:56:71:42:ad:1a:b6:46:17
  Certificate: C=us, O=juniper, CN=First Officer
    Fingerprint: 46:71:15:34:f0:a6:41:76:65:81:33:4f:68:47:c4:df:78:b8:e3:3f
Do you want to load the above CA certificate ? [yes,no] (no) yes
```

NOTE: If you obtain the CA certificate directly from the CA (for example, as an e-mail attachment or Web site download), you can install it with the request security pki ca-certificate load command.

Next, you must generate a private/public key pair before you can create a local certificate.

```
user@R2> request security pki generate-key-pair certificate-id local-entrust2
```

```
Generated key pair local-entrust2, key size 1024 bits
```

When the key pair is available, generate a local certificate request and send it to the CA for processing.

```
user@R2> request security pki generate-certificate-request
certificate-id local-entrust2 domain-name router2.juniper.net
filename entrust-req2 subject cn=router2.juniper.net
Generated certificate request
-----BEGIN CERTIFICATE REQUEST-----
MIIBoTCCAQoCAQAwGjEYMBYGA1UEAxMPdHAxLmp1bmlwZXIubmV0MIGfMA0GCSqG
SIb3DQEBAQUAA4GNADCBiQKBgQCiUFklQws1Ud+AqN5DDxRs2kVyKEhh9qoVFnz+
Hz4c9vsy3B8ElwTJlkmIt2cB3yifB6zePd+6WYpf57Crwre7YqPkiXM31F6z3YjX
H+1BPNbCxNWYvyrnSyVYDbFj8oOXyqog8ACDfVL2JBWrPNBYy7imq/K9soDBbAs6
5hZqqwIDAQABoEcwRQYJKoZIhvcNAQkOMTgwNjAOBgNVHQ8BAf8EBAMCB4AwJAYD
VR0RAQH/BBowGIIWdHAxLmVuZ2xhYi5qdW5pcGVyLm5ldDANBgkqhkiG9w0BAQQF
AAOBgQBc2rq1v5SOQXH7LCb/FdqAL8ZM6GoaN5d6cGwq4bB6a7UQFgtoH4O6gQ3G
3iHOZfz4xMIBpJYuGd1dkqgvcDoH3AgTsLkfn7Wi3x5H2qeQVs9bvL4P5nvEZLND
EIMUHwteolZCiZ7OfO9Fer9cXWHSQs1UtXtgPqQJy2xIeImLgw==
-----END CERTIFICATE REQUEST-----
Fingerprint:
0d:90:b8:d2:56:74:fc:84:59:62:b9:78:71:9c:e4:9c:54:ba:16:97 (sha1)
1b:08:d4:f7:90:f1:c4:39:08:c9:de:76:00:86:62:b8 (md5)
```

NOTE: You can request the creation and installation of a local certificate online with the request security pki local-certificate enroll command. For more information, see "Generating and Enrolling a Local Digital Certificate" on page 424 or the *JUNOS System Basics and Services Command Reference*.

The trusted CA digitally signs the local certificate and returns it to you. Copy the certificate file into the routing platform and load the certificate.

```
user@R2> request security pki local-certificate load filename /tmp/router2-cert
certificate-id local-entrust2
Local certificate local-entrust2 loaded successfully
```

NOTE: The name of the file sent to you by the CA might not match the name of the certificate identifier. However, the certificate-id name must always match the name of the key pair you generated for the routing platform.

After the local and CA certificates have been loaded, you can reference them in your IPSec configuration.

Using default values in the AS PIC, you do not need to configure an IPSec proposal or IPSec policy. However, you must configure an IKE proposal that specifies the use of digital certificates, reference the IKE proposal and local certificate in an IKE policy, and apply the CA profile to the service set. To enable an IKE proposal for digital certificates, include the rsa-signatures statement at the [edit services ipsec-vpn ike proposal *proposal-name* authentication-method] hierarchy level. To reference the local

certificate in the IKE policy, include the local-certificate statement at the [edit services ipsec-vpn ike policy *policy-name*] hierarchy level. To identify the CA or RA in the service set, include the trusted-ca statement at the [edit services service-set *service-set-name* ipsec-vpn-options] hierarchy level.

NOTE: For more information about default IKE and IPSec policies and proposals on the AS PIC, see Table 42 on page 417.

Optionally, you can configure automatic reenrollment of the certificate with the auto-re-enrollment statement at the [edit security pki] hierarchy level.

The remaining configuration components of your IKE-based IPSec tunnel are the same as when you use preshared keys. Enable OSPF as the underlying routing protocol to connect to Routers 1 and 3. Configure a bidirectional IKE dynamic SA in a rule called rule-ike at the [edit ipsec-vpn rule] hierarchy level. Reference this rule in a service set called service-set-dynamic-BiEspsha3des at the [edit services service-set] hierarchy level.

To direct traffic into the AS PIC and the IPSec tunnel, configure a next-hop style service set and add the adaptive services logical interface used as the IPSec inside interface into the OSPF configuration.

Router 2

```
[edit]
interfaces {
   so-0/0/0 {
      description "To R1 so-0/0/0";
      unit 0 {
         family inet {
            address 10.1.12.1/30;
         }
      }
   }
   so-0/0/1 {
      description "To R3 so-0/0/1";
      unit 0 {
         family inet {
            address 10.1.15.1/30;
         }
      }
   }
   sp-1/2/0 {
      unit 0 {
         family inet;
      }
      unit 1 { # sp-1/2/0.1 is the IPSec inside interface.
         family inet;
         service-domain inside;
      }
      unit 2 { # sp-1/2/0.2 is the IPSec outside interface.
         family inet;
         service-domain outside;
      }
   }
```

```
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.2/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.2;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface sp-1/2/0.1; # This sends OSPF traffic over the IPSec tunnel.
            interface lo0.0;
        }
    }
}
security { # Configure CA profiles here, including the URLs used to reach the CAs.
    pki {
        ca-profile entrust {
            ca-identity entrust;
            enrollment {
                url http://ca-1.jnpr.net/cgi-bin/pkiclient.exe;
            }
            revocation-check {
                crl {
                    url ldap://10.157.90.185/o=juniper,c=uscertificateRevocationListbase;
                    # Specify the URL of the LDAP server where the CA stores the CRL.
                }
            }
        }
        ca-profile microsoft {
            ca-identity microsoft;
            enrollment {
                url http://192.168.11.78:80/certsrv/mscep/mscep.dll;
            }
        }
        ca-profile verisign {
            ca-identity verisign;
            enrollment {
                url http://pilotonsiteipsec.verisign.com/cgi-bin/pkiclient.exe;
            }
        }
    }
}
services {
    service-set service-set-dynamic-BiEspsha3des { # Define your service set here.
        next-hop-service { # Required for dynamic routing protocols such as OSPF.
            inside-service-interface sp-1/2/0.1;
            outside-service-interface sp-1/2/0.2;
        }
        ipsec-vpn-options {
            trusted-ca entrust; # Reference the CA profile here.
```

```
            local-gateway 10.1.15.1; # Specify the local IP address of the IPSec tunnel.
        }
        ipsec-vpn-rules rule-ike; # Reference your IPSec VPN rule here.
    }
    ipsec-vpn {
        rule rule-ike { # Define your IPSec VPN rule here.
            term term-ike {
                then {
                    remote-gateway 10.1.15.2; # The remote IP address of the IPSec tunnel.
                    dynamic { # This creates a dynamic SA.
                        ike-policy ike-digital-certificates; # Reference your IKE policy here.
                    }
                }
            }
            match-direction input; # Specify in which direction the rule should match.
        }
        ike {
            proposal ike-proposal {
                authentication-method rsa-signatures; # Uses digital certificates
            }
            policy ike-digital-certificates {
                proposals ike-proposal; # Apply the IKE proposal here.
                local-id fqdn router2.juniper.net; # Provide an identifier for the local router.
                local-certificate local-entrust2; # Reference the local certificate here.
                remote-id fqdn router3.juniper.net; # Provide an ID for the remote router.
            }
        }
        establish-tunnels immediately;
    }
}
```

On Router 3, you must repeat the digital certificate procedures you performed on Router 2. If the IPSec peers do not have a symmetrical configuration containing all the necessary components, they cannot establish a peering relationship.

You need to request a CA certificate, create a local certificate, load these digital certificates into the router, and reference them in your IPSec configuration. Begin by configuring an IPSec CA profile. Include the ca-profile statement at the [edit security pki] hierarchy level and specify the trusted CA and URL of the CA server that handles CA certificate processing. Include the CRL statements found on Router 2 to complete your CA profile on Router 3.

After you configure the CA profile, request a CA certificate from the trusted CA. In this example, the certificate is enrolled online and installed into the routing platform automatically.

```
user@R3> request security pki ca-certificate enroll ca-profile entrust
Received following certificates:
  Certificate: C=us, O=juniper
    Fingerprint: 00:8e:6f:58:dd:68:bf:25:0a:e3:f9:17:70:d6:61:f3:53:a7:79:10
  Certificate: C=us, O=juniper, CN=First Officer
    Fingerprint: bc:78:87:9b:a7:91:13:20:71:db:ac:b5:56:71:42:ad:1a:b6:46:17
  Certificate: C=us, O=juniper, CN=First Officer
```

```
    Fingerprint: 46:71:15:34:f0:a6:41:76:65:81:33:4f:68:47:c4:df:78:b8:e3:3f
Do you want to load the above CA certificate ? [yes,no] (no) yes
```

NOTE: If you obtain the CA certificate directly from the CA (for example, as an e-mail attachment or Web site download), you can install it with the request security pki ca-certificate load command.

Next, generate a private/public key pair.

```
user@R3> request security pki generate-key-pair certificate-id local-entrust3
Generated key pair local-entrust3, key size 1024 bits
```

When the key pair is available, you can generate a local certificate request and send it to the CA for processing.

```
user@R3> request security pki generate-certificate-request
certificate-id local-entrust3 domain-name router3.juniper.net
filename entrust-req3 subject cn=router3.juniper.net
Generated certificate request
-----BEGIN CERTIFICATE REQUEST-----
MIIB8jCCAVsCAQAwZTEYMBYGA1UEAxMPdHA1Lmp1bmlwZXIubmV0MRQwEgYDVQQL
EwtFbmdpbmVlcmluZzEQMA4GA1UEChMHSnVuaXBlcjETMBEGA1UECBMKQ2FsaWZv
cm5pYTEMMAoGA1UEBhMDVVNBMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCg
Wjo5Ow8jrnphsOsRFvqQMwC6PlYa65thrJ8nHZ2qgYgRbSrO8hdODhvU6/5VuD2/
zBtgV5ZSAOlyV6DXqlbVj/2XirQAJMRCr1eYu6DhYRBMNq/UaQv4Z8Sse1EJv+uR
HTNbD7xlwpw2zwzltRuGFtFr/FrGBOhF7IE+Xm5e2wIDAQABoEOwSwYJKoZIhvcN
AQkOMT4wPDAOBgNVHQ8BAf8EBAMCB4AwKgYDVR0RAQH/BCAwHocEwKhGk4IWdHA1
LmVuZ2xhYi5qdW5pcGVyLm5ldDANBgkqhkiG9w0BAQQFAAOBgQBbiJ+ZCeQ59/eY
4Rd6awIpJFTzOsvRZLxxjFWogusVTmaD2dsqFBqftS1eJBdeiueRcYMF9vOnOGKm
FNfouegwei5+vzdNmNo55eIb3rs4pP62qOW5CUgmbHrjtp3lyJsvuOxTTcPNY8zw
b6GyM2Hdkk3Vh2ReX11tQUSqYujTjw==
-----END CERTIFICATE REQUEST-----
Fingerprint:
7c:e8:f9:45:93:8d:a3:92:7f:18:29:02:f1:c8:e2:85:3d:ad:df:1f (sha1)
00:4e:df:a0:6b:ad:8c:50:da:7c:a1:cf:5d:37:b0:ea (md5)
```

The trusted CA digitally signs the local certificate and returns it to you. Copy the certificate file into the routing platform and load the certificate.

```
user@R3> request security pki local-certificate load filename /tmp/router3-cert
certificate-id local-entrust3
Local certificate local-entrust3 loaded successfully
```

After the local and CA certificates have been loaded, you can reference them in your IPSec configuration. Using default values in the AS PIC, you do not need to configure an IPSec proposal or IPSec policy. However, you must configure an IKE proposal that uses digital certificates, reference the IKE proposal and local certificate in an IKE policy, and apply the CA profile to the service set. To enable the IKE proposal for digital certificates, include the rsa-signatures statement at the [edit services ipsec-vpn ike proposal *proposal-name* authentication-method] hierarchy level. To reference the local certificate in the IKE policy, include the local-certificate statement at the [edit services ipsec-vpn ike policy *policy-name*] hierarchy level. To identify the CA or RA in

the service set, include the trusted-ca statement at the [edit services service-set *service-set-name* ipsec-vpn-options] hierarchy level.

The remaining configuration components of your IKE-based IPSec tunnel are the same as when you use preshared keys. Enable OSPF as the underlying routing protocol to connect to Routers 2 and 4. Configure a bidirectional IKE dynamic SA in a rule called rule-ike at the [edit ipsec-vpn rule] hierarchy level. Reference this rule in a service set called service-set-dynamic-BiEspsha3des at the [edit services service-set] hierarchy level.

To direct traffic into the AS PIC and the IPSec tunnel, configure a next-hop style service set and add the adaptive services logical interface used as the IPSec inside interface into the OSPF configuration.

Router 3

```
[edit]
interfaces {
   so-0/0/0 {
      description "To R4 so-0/0/0";
      unit 0 {
         family inet {
            address 10.1.56.1/30;
         }
      }
   }
   so-0/0/1 {
      description "To R2 so-0/0/1";
      unit 0 {
         family inet {
            address 10.1.15.2/30;
         }
      }
   }
   sp-1/2/0 {
      unit 0 {
         family inet;
      }
      unit 1 { # sp-1/2/0.1 is the IPSec inside interface.
         family inet;
         service-domain inside;
      }
      unit 2 { # sp-1/2/0.2 is the IPSec outside interface.
         family inet;
         service-domain outside;
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.0.0.3/32;
         }
      }
   }
}
routing-options {
   router-id 10.0.0.3;
```

```
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface sp-1/2/0.1; # This sends OSPF traffic over the IPSec tunnel.
            interface lo0.0;
        }
    }
}
security { # Configure CA profiles here, including the URLs used to reach the CAs.
    pki {
        ca-profile entrust {
            ca-identity entrust;
            enrollment {
                url http://ca-1.jnpr.net/cgi-bin/pkiclient.exe;
            }
            revocation-check {
                crl {
                    url ldap://10.157.90.185/o=juniper,c=uscertificateRevocationListbase;
                    # Specify the URL of the LDAP server where the CA stores the CRL.
                }
            }
        }
        ca-profile microsoft {
            ca-identity microsoft;
            enrollment {
                url http://192.168.11.78:80/certsrv/mscep/mscep.dll;
            }
        }
        ca-profile verisign {
            ca-identity verisign;
            enrollment {
                url http://pilotonsiteipsec.verisign.com/cgi-bin/pkiclient.exe;
            }
        }
    }
}
services {
    service-set service-set-dynamic-BiEspsha3des { # Define your service set here.
        next-hop-service { # Required for dynamic routing protocols such as OSPF.
            inside-service-interface sp-1/2/0.1;
            outside-service-interface sp-1/2/0.2;
        }
        ipsec-vpn-options {
            trusted-ca entrust; # Reference the CA profile here.
            local-gateway 10.1.15.2; # Specify the local IP address of the IPSec tunnel.
        }
        ipsec-vpn-rules rule-ike; # Reference your IPSec VPN rule here.
    }
    ipsec-vpn {
        rule rule-ike { # Define your IPSec VPN rule here.
            term term-ike {
                then {
                    remote-gateway 10.1.15.1; # The remote IP address of the IPSec tunnel.
                    dynamic { # This creates a dynamic SA.
```

```
                    ike-policy ike-digital-certificates; # Reference your IKE policy here.
                }
            }
        }
        match-direction input; # Specify in which direction the rule should match.
    }
    ike {
        proposal ike-proposal {
            authentication-method rsa-signatures; # Uses digital certificates
        }
        policy ike-digital-certificates {
            proposals ike-proposal; # Apply the IKE proposal here.
            local-id fqdn router3.juniper.net; # Provide an identifier for the local router.
            local-certificate local-entrust3; # Reference the local certificate here.
            remote-id fqdn router2.juniper.net; # Provide an ID for the remote router.
        }
    }
    establish-tunnels immediately;
  }
}
```

On Router 4, provide basic OSPF connectivity to Router 3.

Router 4

```
[edit]
interfaces {
  so-0/0/0 {
    description "To R3 so-0/0/0";
    unit 0 {
      family inet {
        address 10.1.56.2/30;
      }
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.0.0.4/32;
      }
    }
  }
}
routing-options {
  router-id 10.0.0.4;
}
protocols {
  ospf {
    area 0.0.0.0 {
      interface so-0/0/0.0;
      interface lo0.0;
    }
  }
}
```

Verifying Your Work

To verify proper operation of an IKE-based dynamic SA on the AS PIC, use the following commands:

- ping
- show services ipsec-vpn certificates (detail)
- show services ipsec-vpn ike security-associations (detail)
- show services ipsec-vpn ipsec security-associations (detail)
- show services ipsec-vpn ipsec statistics
- traceroute

To verify and manage digital certificates in your routing platform, use the following commands:

- show security pki ca-certificate (detail)
- show security pki certificate-request (detail)
- show security pki local-certificate (detail)

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

- Router 1 on page 497
- Router 2 on page 498
- Router 3 on page 501
- Router 4 on page 504

Router 1

On Router 1, issue a ping command to the so-0/0/0 interface on Router 4 to send traffic across the IPSec tunnel.

```
user@R1> ping 10.1.56.2
PING 10.1.56.2 (10.1.56.2): 56 data bytes
64 bytes from 10.1.56.2: icmp_seq=0 ttl=254 time=1.351 ms
64 bytes from 10.1.56.2: icmp_seq=1 ttl=254 time=1.187 ms
64 bytes from 10.1.56.2: icmp_seq=2 ttl=254 time=1.172 ms
64 bytes from 10.1.56.2: icmp_seq=3 ttl=254 time=1.154 ms
64 bytes from 10.1.56.2: icmp_seq=4 ttl=254 time=1.156 ms
^C
--- 10.1.56.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.154/1.204/1.351/0.074 ms
```

If you ping the loopback address of Router 4, the operation succeeds because the address is part of the OSPF network configured on Router 4.

```
user@R1> ping 10.0.0.4
PING 10.0.0.4 (10.0.0.4): 56 data bytes
64 bytes from 10.0.0.4: icmp_seq=0 ttl=62 time=1.318 ms
64 bytes from 10.0.0.4: icmp_seq=1 ttl=62 time=1.084 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=62 time=3.260 ms
^C
--- 10.0.0.4 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.084/1.887/3.260/0.975 ms
```

Router 2

To verify that matched traffic is being diverted to the bidirectional IPSec tunnel, view the IPSec statistics:

```
user@R2> show services ipsec-vpn ipsec statistics

PIC: sp-1/2/0, Service set: service-set-dynamic-BiEspsha3des

ESP Statistics:
  Encrypted bytes:           162056
  Decrypted bytes:           161896
  Encrypted packets:           2215
  Decrypted packets:           2216
AH Statistics:
  Input bytes:                    0
  Output bytes:                   0
  Input packets:                  0
  Output packets:                 0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
```

To verify that the IKE SA negotiation is successful, issue the show services ipsec-vpn ike security-associations command:

```
user@R2> show services ipsec-vpn ike security-associations
Remote Address  State         Initiator cookie  Responder cookie  Exchange type
10.1.15.2       Matured       d82610c59114fd37  ec4391f76783ef28  Main
```

To verify that the IPSec security association is active, issue the show services ipsec-vpn ipsec security-associations detail command. Notice that the SA contains the default settings inherent in the AS PIC, such as ESP for the protocol and HMAC-SHA1-96 for the authentication algorithm.

```
user@R2> show services ipsec-vpn ipsec security-associations detail
Service set: service-set-dynamic-BiEspsha3des

  Rule: rule-ike, Term: term-ike, Tunnel index: 1
  Local gateway: 10.1.15.1, Remote gateway: 10.1.15.2
  IPSec inside interface: sp-1/2/0.1
  Local identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)

    Direction: inbound, SPI: 857451461, AUX-SPI: 0
```

```
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 9052 seconds
    Hard lifetime: Expires in 9187 seconds
    Anti-replay service: Enabled, Replay window size: 64

    Direction: outbound, SPI: 1272330309, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 9052 seconds
    Hard lifetime: Expires in 9187 seconds
    Anti-replay service: Enabled, Replay window size: 64
```

To display the digital certificates that are used to establish the IPSec tunnel, issue the show services ipsec-vpn certificates command:

```
user@R2> show services ipsec-vpn certificates
Service set: service-set-dynamic-BiEspsha3des, Total entries: 3
  Certificate cache entry: 3
    Flags: Non-root Trusted
    Issued to: router3.juniper.net, Issued by: juniper
    Alternate subject: router3.juniper.net
    Validity:
      Not before: 2005 Nov 21st, 23:33:58 GMT
      Not after: 2008 Nov 22nd, 00:03:58 GMT

  Certificate cache entry: 2
    Flags: Non-root Trusted
    Issued to: router2.juniper.net, Issued by: juniper
    Alternate subject: router2.juniper.net
    Validity:
      Not before: 2005 Nov 21st, 23:28:22 GMT
      Not after: 2008 Nov 21st, 23:58:22 GMT

  Certificate cache entry: 1
    Flags: Root Trusted
    Issued to: juniper, Issued by: juniper
    Validity:
      Not before: 2005 Oct 18th, 23:54:22 GMT
      Not after: 2025 Oct 19th, 00:24:22 GMT
```

To display the CA certificate, issue the show security pki ca-certificate detail command. Notice that there are three separate certificates: one for certificate signing, one for key encipherment, and one for the CA's digital signature.

```
user@R2> show security pki ca-certificate detail
Certificate identifier: entrust
  Certificate version: 3
  Serial number: 4355 9235
  Issuer:
    Organization: juniper, Country: us
  Subject:
    Organization: juniper, Country: us
  Validity:
    Not before: 2005 Oct 18th, 23:54:22 GMT
    Not after: 2025 Oct 19th, 00:24:22 GMT
  Public key algorithm: rsaEncryption(1024 bits)
    cb:9e:2d:c0:70:f8:ea:3c:f2:b5:f0:02:48:87:dc:68:99:a3:57:4f
    0e:b9:98:0b:95:47:0d:1f:97:7c:53:17:dd:1a:f8:da:e5:08:d1:1c
```

```
    78:68:1f:2f:72:9f:a2:cf:81:e3:ce:c5:56:89:ce:f0:97:93:fa:36
    19:3e:18:7d:8c:9d:21:fe:1f:c3:87:8d:b3:5d:f3:03:66:9d:16:a7
    bf:18:3f:f0:7a:80:f0:62:50:43:83:4f:0e:d7:c6:42:48:c0:8a:b2
    c7:46:30:38:df:9b:dc:bc:b5:08:7a:f3:cd:64:db:2b:71:67:fe:d8
    04:47:08:07:de:17:23:13
  Signature algorithm: sha1WithRSAEncryption
  Fingerprint:
    00:8e:6f:58:dd:68:bf:25:0a:e3:f9:17:70:d6:61:f3:53:a7:79:10 (sha1)
    71:6f:6a:76:17:9b:d6:2a:e7:5a:72:97:82:6d:26:86 (md5)
  Distribution CRL:
    C=us, O=juniper, CN=CRL1
    http://CA-1/CRL/juniper_us_crlfile.crl
  Use for key: CRL signing, Certificate signing

Certificate identifier: entrust
  Certificate version: 3
  Serial number: 4355 925c
  Issuer:
    Organization: juniper, Country: us
  Subject:
    Organization: juniper, Country: us, Common name: First Officer
  Validity:
    Not before: 2005 Oct 18th, 23:55:59 GMT
    Not after: 2008 Oct 19th, 00:25:59 GMT
  Public key algorithm: rsaEncryption(1024 bits)
    c0:a4:21:32:95:0a:cd:ec:12:03:d1:a2:89:71:8e:ce:4e:a6:f9:2f
    1a:9a:13:8c:f6:a0:3d:c9:bd:9d:c2:a0:41:77:99:1b:1e:ed:5b:80
    34:46:f8:5b:28:34:38:2e:91:7d:4e:ad:14:86:78:67:e7:02:1d:2e
    19:11:b7:fa:0d:ba:64:20:e1:28:4e:3e:bb:6e:64:dc:cd:b1:b4:7a
    ca:8f:47:dd:40:69:c2:35:95:ce:b8:85:56:d7:0f:2d:04:4d:5d:d8
    42:e1:4f:6b:bf:38:c0:45:1e:9e:f0:b4:7f:74:6f:e9:70:fd:4a:78
    da:eb:10:27:bd:46:34:33
  Signature algorithm: sha1WithRSAEncryption
  Fingerprint:
    bc:78:87:9b:a7:91:13:20:71:db:ac:b5:56:71:42:ad:1a:b6:46:17 (sha1)
    23:79:40:c9:6d:a6:f0:ca:e0:13:30:d4:29:6f:86:79 (md5)
  Distribution CRL:
    C=us, O=juniper, CN=CRL1
    http://CA-1/CRL/juniper_us_crlfile.crl
  Use for key: Key encipherment
Certificate identifier: entrust
  Certificate version: 3
  Serial number: 4355 925b
  Issuer:
    Organization: juniper, Country: us
  Subject:
    Organization: juniper, Country: us, Common name: First Officer
  Validity:
    Not before: 2005 Oct 18th, 23:55:59 GMT
    Not after: 2008 Oct 19th, 00:25:59 GMT
  Public key algorithm: rsaEncryption(1024 bits)
    ea:75:c4:f3:58:08:ea:65:5c:7e:b3:de:63:0a:cf:cf:ec:9a:82:e2
    d7:e8:b9:2f:bd:4b:cd:86:2f:f1:dd:d8:a2:95:af:ab:51:a5:49:4e
    00:10:c6:25:ff:b5:49:6a:99:64:74:69:e5:8c:23:5b:b4:70:62:8e
    e4:f9:a2:28:d4:54:e2:0b:1f:50:a2:92:cf:6c:8f:ae:10:d4:69:3c
    90:e2:1f:04:ea:ac:05:9b:3a:93:74:d0:59:24:e9:d2:9d:c2:ef:22
    b9:32:c7:2c:29:4f:91:cb:5a:26:fe:1d:c0:36:dc:f4:9c:8b:f5:26
    af:44:bf:53:aa:d4:5f:67
  Signature algorithm: sha1WithRSAEncryption
  Fingerprint:
    46:71:15:34:f0:a6:41:76:65:81:33:4f:68:47:c4:df:78:b8:e3:3f (sha1)
```

```
    ee:cc:c7:f4:5d:ac:65:33:0a:55:db:59:72:2c:dd:16 (md5)
  Distribution CRL:
    C=us, O=juniper, CN=CRL1
    http://CA-1/CRL/juniper_us_crlfile.crl
  Use for key: Digital signature
```

To display the local certificate request, issue the show security pki certificate-request command:

```
user@R2> show security pki certificate-request
Certificate identifier: local-entrust2
  Issued to: router2.juniper.net
  Public key algorithm: rsaEncryption(1024 bits)
  Public key verification status: Passed
```

To display the local certificate, issue the show security pki local-certificate command:

```
user@R2> show security pki local-certificate
Certificate identifier: local-entrust2
  Issued to: router2.juniper.net, Issued by: juniper
  Validity:
    Not before: 2005 Nov 21st, 23:28:22 GMT
    Not after: 2008 Nov 21st, 23:58:22 GMT
  Public key algorithm: rsaEncryption(1024 bits)
  Public key verification status: Passed
```

Router 3

To verify that matched traffic is being diverted to the bidirectional IPSec tunnel, view the IPSec statistics:

```
user@R3> show services ipsec-vpn ipsec statistics

PIC: sp-1/2/0, Service set: service-set-dynamic-BiEspsha3des

ESP Statistics:
  Encrypted bytes:           161896
  Decrypted bytes:           162056
  Encrypted packets:           2216
  Decrypted packets:           2215
AH Statistics:
  Input bytes:                    0
  Output bytes:                   0
  Input packets:                  0
  Output packets:                 0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
```

To verify that the IKE SA negotiation is successful, issue the show services ipsec-vpn ike security-associations command. To be successful, the SA on Router 3 must contain the same settings you specified on Router 2.

```
user@R3> show services ipsec-vpn ike security-associations
```

```
Remote Address  State         Initiator cookie  Responder cookie  Exchange type
10.1.15.1       Matured       d82610c59114fd37  ec4391f76783ef28  Main
```

To verify that the IPSec SA is active, issue the show services ipsec-vpn ipsec security-associations detail command. To be successful, the SA on Router 3 must contain the same settings you specified on Router 2.

```
user@R3> show services ipsec-vpn ipsec security-associations detail
Service set: service-set-dynamic-BiEspsha3des

  Rule: rule-ike, Term: term-ike, Tunnel index: 1
  Local gateway: 10.1.15.2, Remote gateway: 10.1.15.1
  IPSec inside interface: sp-1/2/0.1
  Local identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)

    Direction: inbound, SPI: 1272330309, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 7219 seconds
    Hard lifetime: Expires in 7309 seconds
    Anti-replay service: Enabled, Replay window size: 64

    Direction: outbound, SPI: 857451461, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 7219 seconds
    Hard lifetime: Expires in 7309 seconds
    Anti-replay service: Enabled, Replay window size: 64
```

To display the digital certificates that are used to establish the IPSec tunnel, issue the show services ipsec-vpn certificates command:

```
user@R3> show services ipsec-vpn certificates
Service set: service-set-dynamic-BiEspsha3des, Total entries: 3
  Certificate cache entry: 3
    Flags: Non-root Trusted
    Issued to: router3.juniper.net, Issued by: juniper
    Alternate subject: router3.juniper.net
    Validity:
      Not before: 2005 Nov 21st, 23:33:58 GMT
      Not after: 2008 Nov 22nd, 00:03:58 GMT

  Certificate cache entry: 2
    Flags: Non-root Trusted
    Issued to: router2.juniper.net, Issued by: juniper
    Alternate subject: router2.juniper.net
    Validity:
      Not before: 2005 Nov 21st, 23:28:22 GMT
      Not after: 2008 Nov 21st, 23:58:22 GMT

  Certificate cache entry: 1
    Flags: Root Trusted
    Issued to: juniper, Issued by: juniper
    Validity:
      Not before: 2005 Oct 18th, 23:54:22 GMT
      Not after: 2025 Oct 19th, 00:24:22 GMT
```

To display the CA certificate, issue the show security pki ca-certificate detail command. Notice that there are three separate certificates: one for certificate signing, one for key encipherment, and one for the CA's digital signature.

```
user@R3> show security pki ca-certificate detail
Certificate identifier: entrust
  Certificate version: 3
  Serial number: 4355 9235
  Issuer:
    Organization: juniper, Country: us
  Subject:
    Organization: juniper, Country: us
  Validity:
    Not before: 2005 Oct 18th, 23:54:22 GMT
    Not after: 2025 Oct 19th, 00:24:22 GMT
  Public key algorithm: rsaEncryption(1024 bits)
    cb:9e:2d:c0:70:f8:ea:3c:f2:b5:f0:02:48:87:dc:68:99:a3:57:4f
    0e:b9:98:0b:95:47:0d:1f:97:7c:53:17:dd:1a:f8:da:e5:08:d1:1c
    78:68:1f:2f:72:9f:a2:cf:81:e3:ce:c5:56:89:ce:f0:97:93:fa:36
    19:3e:18:7d:8c:9d:21:fe:1f:c3:87:8d:b3:5d:f3:03:66:9d:16:a7
    bf:18:3f:f0:7a:80:f0:62:50:43:83:4f:0e:d7:c6:42:48:c0:8a:b2
    c7:46:30:38:df:9b:dc:bc:b5:08:7a:f3:cd:64:db:2b:71:67:fe:d8
    04:47:08:07:de:17:23:13
  Signature algorithm: sha1WithRSAEncryption
  Fingerprint:
    00:8e:6f:58:dd:68:bf:25:0a:e3:f9:17:70:d6:61:f3:53:a7:79:10 (sha1)
    71:6f:6a:76:17:9b:d6:2a:e7:5a:72:97:82:6d:26:86 (md5)
  Distribution CRL:
    C=us, O=juniper, CN=CRL1
    http://CA-1/CRL/juniper_us_crlfile.crl
  Use for key: CRL signing, Certificate signing
Certificate identifier: entrust
  Certificate version: 3
  Serial number: 4355 925c
  Issuer:
    Organization: juniper, Country: us
  Subject:
    Organization: juniper, Country: us, Common name: First Officer
  Validity:
    Not before: 2005 Oct 18th, 23:55:59 GMT
    Not after: 2008 Oct 19th, 00:25:59 GMT
  Public key algorithm: rsaEncryption(1024 bits)
    c0:a4:21:32:95:0a:cd:ec:12:03:d1:a2:89:71:8e:ce:4e:a6:f9:2f
    1a:9a:13:8c:f6:a0:3d:c9:bd:9d:c2:a0:41:77:99:1b:1e:ed:5b:80
    34:46:f8:5b:28:34:38:2e:91:7d:4e:ad:14:86:78:67:e7:02:1d:2e
    19:11:b7:fa:0d:ba:64:20:e1:28:4e:3e:bb:6e:64:dc:cd:b1:b4:7a
    ca:8f:47:dd:40:69:c2:35:95:ce:b8:85:56:d7:0f:2d:04:4d:5d:d8
    42:e1:4f:6b:bf:38:c0:45:1e:9e:f0:b4:7f:74:6f:e9:70:fd:4a:78
    da:eb:10:27:bd:46:34:33
  Signature algorithm: sha1WithRSAEncryption
  Fingerprint:
    bc:78:87:9b:a7:91:13:20:71:db:ac:b5:56:71:42:ad:1a:b6:46:17 (sha1)
    23:79:40:c9:6d:a6:f0:ca:e0:13:30:d4:29:6f:86:79 (md5)
  Distribution CRL:
    C=us, O=juniper, CN=CRL1
    http://CA-1/CRL/juniper_us_crlfile.crl
  Use for key: Key encipherment
Certificate identifier: entrust
  Certificate version: 3
  Serial number: 4355 925b
```

```
  Issuer:
    Organization: juniper, Country: us
  Subject:
    Organization: juniper, Country: us, Common name: First Officer
  Validity:
    Not before: 2005 Oct 18th, 23:55:59 GMT
    Not after: 2008 Oct 19th, 00:25:59 GMT
  Public key algorithm: rsaEncryption(1024 bits)
    ea:75:c4:f3:58:08:ea:65:5c:7e:b3:de:63:0a:cf:cf:ec:9a:82:e2
    d7:e8:b9:2f:bd:4b:cd:86:2f:f1:dd:d8:a2:95:af:ab:51:a5:49:4e
    00:10:c6:25:ff:b5:49:6a:99:64:74:69:e5:8c:23:5b:b4:70:62:8e
    e4:f9:a2:28:d4:54:e2:0b:1f:50:a2:92:cf:6c:8f:ae:10:d4:69:3c
    90:e2:1f:04:ea:ac:05:9b:3a:93:74:d0:59:24:e9:d2:9d:c2:ef:22
    b9:32:c7:2c:29:4f:91:cb:5a:26:fe:1d:c0:36:dc:f4:9c:8b:f5:26
    af:44:bf:53:aa:d4:5f:67
  Signature algorithm: sha1WithRSAEncryption
  Fingerprint:
    46:71:15:34:f0:a6:41:76:65:81:33:4f:68:47:c4:df:78:b8:e3:3f (sha1)
    ee:cc:c7:f4:5d:ac:65:33:0a:55:db:59:72:2c:dd:16 (md5)
  Distribution CRL:
    C=us, O=juniper, CN=CRL1
    http://CA-1/CRL/juniper_us_crlfile.crl
  Use for key: Digital signature
```

To display the local certificate request, issue the show security pki certificate-request command:

```
user@R3> show security pki certificate-request
Certificate identifier: local-entrust3
  Issued to: router3.juniper.net
  Public key algorithm: rsaEncryption(1024 bits)
  Public key verification status: Passed
```

To display the local certificate, issue the show security pki local-certificate command:

```
user@R3> show security pki local-certificate
Certificate identifier: local-entrust3
  Issued to: router3.juniper.net, Issued by: juniper
  Validity:
    Not before: 2005 Nov 21st, 23:33:58 GMT
    Not after: 2008 Nov 22nd, 00:03:58 GMT
  Public key algorithm: rsaEncryption(1024 bits)
  Public key verification status: Passed
```

Router 4

On Router 4, issue a ping command to the so-0/0/0 interface on Router 1 to send traffic across the IPSec tunnel.

```
user@R4> ping 10.1.12.2
PING 10.1.12.2 (10.1.12.2): 56 data bytes
64 bytes from 10.1.12.2: icmp_seq=0 ttl=254 time=1.350 ms
64 bytes from 10.1.12.2: icmp_seq=1 ttl=254 time=1.161 ms
64 bytes from 10.1.12.2: icmp_seq=2 ttl=254 time=1.124 ms
64 bytes from 10.1.12.2: icmp_seq=5 ttl=254 time=1.116 ms
^C
--- 10.1.12.2 ping statistics ---
```

```
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.116/1.172/1.350/0.081 ms
```

The final way you can confirm that traffic travels over the IPSec tunnel is by issuing the traceroute command to the so-0/0/0 interface on Router 1. Notice that the physical interface between Routers 2 and 3 is not referenced in the path; traffic enters the IPSec tunnel through the adaptive services IPSec inside interface on Router 3, passes through the loopback interface on Router 2, and ends at the so-0/0/0 interface on Router 1.

```
user@R4> traceroute 10.1.12.2
traceroute to 10.1.12.2 (10.1.12.2), 30 hops max, 40 byte packets
 1  10.1.15.2 (10.1.15.2)  0.987 ms  0.630 ms  0.563 ms
 2  10.0.0.2 (10.0.0.2)  1.194 ms  1.058 ms  1.033 ms
 3  10.1.12.2 (10.1.12.2)  1.073 ms  0.949 ms  0.932 ms
```

For additional information on using digital certificates, see the *JUNOS Services Interfaces Configuration Guide* and the *JUNOS System Basics and Services Command Reference*.

Example: Dynamic Endpoint Tunneling Configuration

Figure 52: IPSec Dynamic Endpoint Tunneling Topology Diagram

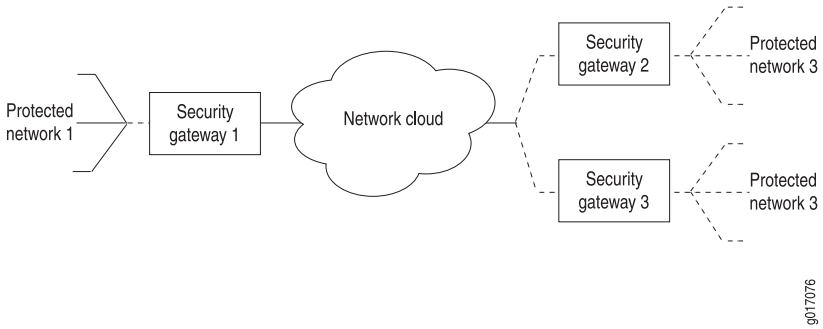


Figure 52 on page 505 shows a local network N-1 located behind security gateway SG-1. SG-1 is a Juniper Networks routing platform terminating dynamic peer endpoints. The tunnel termination address on SG-1 is 10.7.7.2 and the local network address is 172.16.1.0/24.

A remote peer router obtains addresses from an ISP pool and runs RFC-compliant IKE. Remote network N-2 has address 172.16.2.0/24 and is located behind security gateway SG-2 with tunnel termination address 10.7.7.1.

On Router SG-1, configure an IKE access profile to accept proposals from SG-2. Apply the interface identifier from the access profile to the inside services interface and apply the IKE access profile itself to the IPSec next-hop style service set.

Router SG-1

```
[edit]
access {
   profile ike_access {
      client * { # Accepts proposals from specified peers that use the preshared key.
```

```
            ike {
               allowed-proxy-pair local 10.255.14.63/32 remote 10.255.14.64/32;
               pre-shared-key ascii-text "$9$1hoESeLxdgoGvWoGDif5IEc"; # SECRET-DATA
               interface-id test_id; # Apply this ID to the inside services interfaces.
            }
         }
      }
   }
   interfaces {
      fe-0/0/0 {
         description "Connection to the local network";
         unit 0 {
            family inet {
               address 172.16.1.1/24;
            }
         }
      }
      so-1/0/0 {
         description "Connection to SG-2";
         no-keepalives;
         encapsulation cisco-hdlc;
         unit 0 {
            family inet {
               address 10.7.7.2/30;
            }
         }
      }
      sp-3/3/0 {
         unit 0 {
            family inet;
         }
         unit 3 {
            dial-options {
               ipsec-interface-id test_id; # Accepts dynamic endpoint tunnels.
               shared;
            }
            service-domain inside;
         }
         unit 4 {
            family inet;
            service-domain outside;
         }
      }
   }
   services {
      service-set dynamic_nh_ss { # Create a next-hop service set
         next-hop-service { # for the dynamic endpoint tunnels.
            inside-service-interface sp-3/3/0.3;
            outside-service-interface sp-3/3/0.4;
         }
         ipsec-vpn-options {
            local-gateway 10.7.7.2;
            ike-access-profile ike_access; # Apply the IKE access profile here.
         }
      }
   }
```

Verifying Your Work

To verify proper operation of a dynamic endpoint tunnel configured on the AS PIC, use the following command:

```
show services ipsec-vpn ipsec security-associations (detail)
```

The following section shows output from this command used with the configuration example. The dynamically created rule _junos_ appears in the output, as well as the establishment of the inbound and outbound dynamically created tunnels.

```
user@router> show services ipsec-vpn ipsec security-associations detail
Service set: dynamic_nh_ss

  Rule:  _junos_ , Term: tunnel4, Tunnel index: 4
  Local gateway: 10.7.7.2, Remote gateway: 10.7.7.1
  Local identity: ipv4(any:0,[0..3]=10.255.14.63)
  Remote identity: ipv4(any:0,[0..3]=10.255.14.64)

    Direction: inbound , SPI: 428111023, AUX-SPI: 0
     Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 27660 seconds
    Hard lifetime: Expires in 27750 seconds
    Anti-replay service: Enabled, Replay window size: 64

    Direction: outbound , SPI: 4035429231, AUX-SPI: 0
     Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 27660 seconds
    Hard lifetime: Expires in 27750 seconds
    Anti-replay service: Enabled, Replay window size: 64
```

For More Information

For additional information about IPSec, see the following:

- *JUNOS System Basics Configuration Guide*
- *JUNOS Services Interfaces Configuration Guide*
- *JUNOS System Basics and Services Command Reference*
- RFC 1321, *The MD5 Message-Digest Algorithm*
- RFC 2104, *HMAC: Keyed-Hashing for Message Authentication*
- RFC 2401, *Security Architecture for the Internet Protocol*
- RFC 2402, *IP Authentication Header*
- RFC 2405, *The ESP DES-CBC Cipher Algorithm With Explicit IV*
- RFC 2406, *IP Encapsulating Security Payload (ESP)*
- RFC 2408, *Internet Security Association and Key Management Protocol (ISAKMP)*
- RFC 2409, *The Internet Key Exchange (IKE)*

- RFC 2460, *Internet Protocol, Version 6 (IPv6) Specification*
- RFC 3174, *US Secure Hash Algorithm 1 (SHA1)*
- RFC 3602, *The AES-CBC Cipher Algorithm and Its Use with IPsec*
- *Data Encryption Standard*, Federal Information Processing Standards (FIPS) Publication 46-3, http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf (includes information about DES and 3DES)
- *Descriptions of SHA-256, SHA-384, and SHA-512*, http://csrc.nist.gov/cryptval/shs/sha256-384-512.pdf
- Internet draft draft-eastlake-sha2-02.txt, *US Secure Hash Algorithms (SHA and HMAC-SHA)* (expires July 2006)

Revision History

13 April 2009—9.5R1 Release. Roy Spencer.

15 January 2009—9.4R1 Release. Roy Spencer.

10 October 2008—9.3R1 Release. Roy Spencer.

10 July 2008—9.2R1 Release. Roy Spencer.

10 April 2008—9.1R1 Release. Roy Spencer.

1 February 2008—9.0R1 Release. Fawn Damitio.

27 March 2007—Added support for IPSec over OSPFv2 links. 8.2R1 Release. Fawn Damitio.

12 January 2007—Added support for M120 routers. 8.2R1 Release. Fawn Damitio.

15 September 2006—Added support for certificate revocation lists (CRLs), and IPSec IKE in routing instances, 8.1R1 Release. Ines Salazar and Richard Hendricks.

29 June 2006—8.0R1 Release. Richard Hendricks.

27 March 2006—Added support for AES encryption and SHA-256 authentication on J-series Services Routers and AS PICs installed in M-series routers, and IPv6-based IPSec for AS PICs installed in M-series and T-series routing platforms, 7.6R1 Release. Richard Hendricks.

9 January 2006—Added support for digital certificates on J-series Services Routers and AS PICs installed in M-series and T-series routing platforms, and support for the IPSec Monitoring MIB, 7.5R1 Release. Richard Hendricks.

14 September 2005—Added support for dynamic endpoint tunneling and configuring multiple routed tunnels in a single next-hop service set, 7.4R1 Release. Richard Hendricks.

13 June 2005—7.3R1 Release. Richard Hendricks.

5 April 2005—Added support for transport mode IPSec in Routing Engines running OSPFv3 and support for the AS II FIPS PIC, 7.2R1 Release. Richard Hendricks.

2 February 2005—Document converted to Feature Guide format, thoroughly revised, and enhanced with updated examples, JUNOS Release 7.1R1. Richard Hendricks.

07 March 2002—Initial Quick Start Guide written. Tony Sinopoli.

Part 4
VPNs

Chapter 12

Layer 2 Circuits

This feature guide covers these topics:

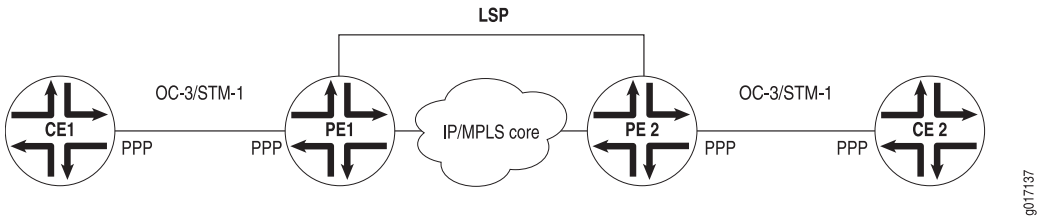
Overview

After the introduction and early adoption of Layer 3 virtual private networks (VPNs) based on RFC 4364 (also known as RFC 2547bis), many customers asked their service providers to offer VPNs that would preserve data at Layer 2. One of the Layer 2 VPN options that has emerged is known as a Layer 2 circuit. It is based on a series of Internet Engineering Task Force (IETF) drafts and RFCs authored by Luca Martini. These so-called "Martini-drafts" include Internet draft draft-martini-l2circuit-encap-mpls-11.txt, *Encapsulation Methods for Transport of Layer 2 Frames Over MPLS Networks* (expires August 2006) and Internet draft draft-martini-l2circuit-trans-mpls-19.txt, *Transport of Layer 2 Frames Over MPLS* (expires September 2006), and establish the basis for Juniper Networks implementation of Layer 2 circuits. This guide shows how to implement Layer 2 circuits in a variety of ways.

Layer 2 circuits allow for the creation of point-to-point Layer 2 connections over an IP and Multiprotocol Label Switching (MPLS)-based network. Physical circuits with the same Layer 2 encapsulations can be connected together across such a network. Layer 2 circuits can allow for the replacement of end-to-end Asynchronous Transfer Mode (ATM) networks, Frame Relay networks, and some portions of Time-Division Multiplexing (TDM) networks, with an IP and MPLS-based network.

In Figure 53 on page 514, an OC3/STM1 interface encapsulated with the Point-to-Point Protocol (PPP) on Router PE1 is connected over a Layer 2 circuit to reach an OC3/STM1 interface encapsulated with PPP on Router PE2. To enable the Layer 2 circuits to operate, the provider edge (PE) routers in Figure 53 on page 514 are part of an MPLS network. Routers PE1 and PE2 must also be Label Distribution Protocol (LDP) peers. Additionally, any interface on the PE routers that connects to a customer edge (CE) router must support circuit cross-connect (CCC) interface encapsulations.

Figure 53: Layer 2 Circuit Connection



Layer 2 circuits are very similar to Layer 2 VPNs. However, there are some significant differences:

- You configure Layer 2 VPNs in a routing instance. As a result, Layer 2 VPNs have unique site and VPN identifiers. However, Layer 2 circuits do not require a routing instance configuration and instead use an alternate method of identifying circuits. Layer 2 circuit peer relationships are established with three components: a logical

interface on the local PE router, the IP address of the remote PE router neighbor, and a virtual circuit identifier.

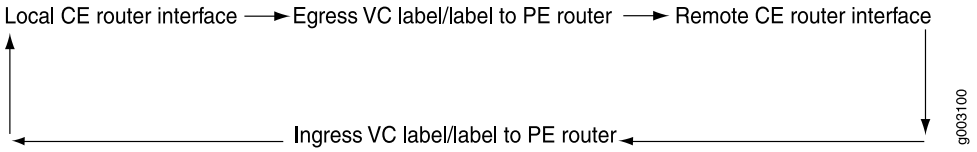
- Layer 2 VPNs, like Layer 3 VPNs, require Border Gateway Protocol (BGP) for transport of traffic between PE routers. In contrast, Layer 2 circuits do not require BGP. Instead, Layer 2 circuits rely on LDP and MPLS for their operation. As a result, Layer 2 circuits require less configuration than Layer 2 VPNs.

Layer 2 circuits are configured between two peers. The peers must use the same interior gateway protocol (IGP), such as Open Shortest Path First (OSPF) or Intermediate System-to-Intermediate System (IS-IS). Also, the peers must have a symmetrical Layer 2 configuration and belong to the same routing domain or autonomous system.

The basic building block for a Layer 2 circuit is a virtual circuit (VC). A VC is a point-to-point Layer 2 connection that is transported over MPLS or any other tunneling technology in a service provider network. A VC is similar to a CCC connection except that multiple VCs are transported over a single MPLS label-switched path (LSP) tunnel between two provider edge (PE) routers. In contrast, CCC only supports a single Layer 2 connection over a single LSP.

In Figure 54 on page 515, the basic inner workings of Layer 2 circuits are explained. Two customer edge (CE) router logical interfaces, one local and one remote, are running the same Layer 2 protocol. Packets are sent from the local CE router to the remote CE router over an egress label advertised by the remote PE router. The label is transported over an LDP LSP (or LDP tunneled through RSVP) to the remote PE router that is connected to the remote CE router. Return traffic from the remote CE router is sent over an ingress label advertised by the local PE router. Once again, the label rides over an LDP LSP (or LDP tunneled through RSVP) to the local PE router from the remote PE router.

Figure 54: Layer 2 Circuit Concept



The Layer 2 circuit framework requires LDP to be used as the signaling protocol for advertising ingress labels. In most cases, it is not necessary to transport the Layer 2 encapsulation across the network; rather, the Layer 2 header can be stripped at one PE router, and reproduced at the egress PE router. Such Layer 2 information is carried in a special Layer 2 circuit header called a *control word*.

In the Layer 2 circuit IETF drafts, the control word is optional for most Layer 2 protocols, except Frame Relay and ATM AAL5 where it is required. However, in JUNOS Release 5.6 and later, a control word for all forms of Layer 2 circuits is sent by default. If you are establishing a Layer 2 circuit between a router running JUNOS Release 5.5 or earlier and a router running JUNOS Release 5.6 or later, use of the control word is negotiated automatically.

The Layer 2 protocols that are supported for Layer 2 circuits are:

- ATM cell-relay mode and ATM Adaptation Layer 5 (AAL5) mode on ATM2 intelligent queuing (IQ) interfaces
- Cisco High-Level Data Link Control (HDLC), Frame Relay, and PPP on SONET/SDH-based interfaces
- Ethernet, VLAN, and Extended VLAN on Ethernet-based interfaces

For an Ethernet 802.1q VLAN or simple Ethernet, the entire Ethernet frame without the preamble or frame check sequence (FCS) is transported. For ATM cell-relay mode, ATM cells are transported without a SAR process. For Cisco HDLC, the frame is transported in its entirety except for HDLC flags and the FCS. For PPP, the frame is transported in its entirety except for any media-specific framing information.

For most protocols, a null control word consisting of all zeroes is sent between Layer 2 circuit neighbors. However, individual bits are available in a control word that can carry Layer 2 protocol control information. The control information is mapped into the control word, which allows the header of a Layer 2 protocol to be stripped from the frame. The remaining data and control word can be sent over the Layer 2 circuit, and the frame can be reassembled with the proper control information at the egress point of the circuit.

The Layer 2 protocols that map Layer 2 control information into special bit fields in the control word are as follows:

- Frame Relay—This control word supports the transport of discard eligible (DE), forward explicit congestion notification (FECN), and backward explicit congestion notification (BECN) information. (For configuration information, see “Option: Mapping Layer 2 Protocol Control Information into a Layer 2 Circuit” on page 524.)
- ATM AAL5 mode—This control word supports the transport of sequence number processing, ATM cell loss priority (CLP), and explicit forward congestion indication (EFCI) information. When you configure an AAL5 mode Layer 2 circuit, the control information is carried by default and no additional configuration is needed.
- ATM cell-relay mode—This control word supports sequence number processing only. When you configure a cell-relay mode Layer 2 circuit, the sequence number information is carried by default and no additional configuration is needed.

The JUNOS software implementation of sequence number processing for ATM cell-relay mode and AAL5 mode is not the same as that described in Sec. 3.1.2 of the IETF draft *Encapsulation Methods for Transport of Layer 2 Frames Over MPLS Networks*. The differences are as follows:

- A packet with a sequence number of 0 is treated as out of sequence.
- Any packet which does not have the next incremental sequence number is considered out of sequence.
- When out-of-sequence packets arrive, the sequence number in the Layer 2 circuit control word increments by one and becomes the expected sequence number for the neighbor.

System Requirements

To implement Layer 2 circuits, your system must meet these minimum requirements:

- JUNOS Release 9.1 or later for nonstop routing (NSR) support.
- JUNOS Release 8.2 or later for M120 and MX-series routing platform support.
- JUNOS Release 7.3 or later for configuring Layer 2 circuits simultaneously over RSVP and LDP LSPs
- JUNOS Release 7.1 or later for local interface switching for Layer 2 circuits
- JUNOS Release 7.0 or later for specifying a unique maximum transmission unit (MTU) for each Layer 2 circuit
- JUNOS Release 6.4 or later for ATM2 IQ interface-based CoS and bandwidth reservation for trunks
- JUNOS Release 6.2 or later for Layer 2 circuit trunk mode on T-series and M320 routing platforms and bandwidth reservation for Layer 2 circuits
- JUNOS Release 6.1 or later for Automatic Protection Switching (APS) for Layer 2 circuits
- JUNOS Release 6.0 or later for Layer 2 circuit traffic engineering, and Frame Relay or ATM control word mapping
- JUNOS Release 5.7 or later for ATM cell-relay mode or AAL5 Layer 2 circuits
- JUNOS Release 5.6 or later for Frame Relay, HDLC, and PPP-based Layer 2 circuits
- JUNOS Release 5.2 or later for Ethernet-based Layer 2 circuits
- Five Juniper Networks M-series, MX-series, or T-series routing platforms

Terms and Acronyms

C

circuit cross-connect (CCC) A Juniper Networks method of exchanging frames between one router interface running a Layer 2 protocol and another router interface using the same Layer 2 protocol. For more information about CCC, see either the *JUNOS Network Interfaces Configuration Guide* or the *JUNOS MPLS Applications Configuration Guide*.

control word A 32-bit field used in Layer 2 circuits to transport sequence information, Layer 2 media control information, and padding.

L

Layer 2 circuit A method of transporting Layer 2 frames between provider edge (PE) routers across an MPLS backbone using LDP signaling.

Configuring Layer 2 Circuits

When you configure Layer 2 circuits, you can use Ethernet, SONET/SDH, and ATM2 IQ interfaces on a PE router. The specific steps you must take to configure these interface types for Layer 2 circuits are described as follows:

Configuring CCC Encapsulation on CE-Facing Ethernet Interfaces

On Ethernet-based CE-facing PE router interfaces, you must configure one of the three Ethernet CCC encapsulation types—Ethernet CCC, VLAN CCC, or Extended VLAN CCC. Use the following guidelines to configure an Ethernet-based interface CCC encapsulation:

- ethernet-ccc—Use Ethernet CCC encapsulation on Ethernet interfaces that must accept packets carrying standard Tag Protocol ID (TPID) values.
- extended-vlan-ccc—Use extended VLAN CCC encapsulation on Ethernet interfaces that have VLAN 802.1Q tagging and must accept packets carrying TPIDs 0x8100, 0x9100, and 0x9901.
- vlan-ccc—Use VLAN CCC encapsulation on Ethernet interfaces with VLAN tagging enabled. VLAN CCC encapsulation supports TPID 0x8100 only. You must configure this encapsulation type on both the physical interface and the logical interface.

- You can configure multiple logical interfaces on Ethernet interfaces configured for VLAN mode. The valid VLAN ID range for the logical interfaces depends upon your PIC type:
  - 1 to 1023—Aggregated Ethernet, 4-port, 8-port, and 12-port Fast Ethernet, and management and internal Ethernet interfaces
  - 1 to 4094—48-port Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet interfaces
  - VLAN ID 0 is reserved for tagging the priority of frames on all Ethernet PICs.
- For encapsulation type vlan-ccc, VLAN IDs from 1 through 511 are reserved for normal VLANs, and VLAN IDs from 512 through 1023 are reserved for CCC VLANs. For encapsulation type extended-vlan-ccc, all valid VLAN IDs per PIC type are valid for CCC VLANs.

For more information about VLAN IDs, see "Binding a VLAN ID to a Logical Interface" in the *JUNOS Network Interfaces Configuration Guide*.

To configure CCC interface encapsulation, include the encapsulation statement at the [edit interfaces *ethernet-interface-fpc/pic/port*] hierarchy level and select ethernet-ccc, vlan-ccc, or extended-vlan-ccc as the encapsulation type. If you select the VLAN CCC encapsulation, also include the vlan-ccc statement at the [edit interfaces *ethernet-interface-fpc/pic/port* unit *unit-number* encapsulation] logical interface hierarchy level. When using either VLAN CCC or extended VLAN CCC encapsulations, include the vlan-tagging statement at the [edit interfaces *ethernet-interface-fpc/pic/port*] hierarchy level.

```
[edit]
interfaces {
    fe-0/1/0 {
        vlan-tagging;
        encapsulation vlan-ccc;
        unit 0 {
            encapsulation vlan-ccc;
            vlan-id 600;
        }
    }
}
```

Configuring CCC Encapsulation on CE-Facing SONET/SDH Interfaces

On SONET/SDH interfaces in a PE router, you can use Frame Relay CCC, Cisco HDLC CCC, or PPP CCC encapsulation for Layer 2 circuits:

- To configure Frame Relay CCC interface encapsulation, include the encapsulation statement at the [edit interfaces so-*fpc/pic/port*] hierarchy level and select frame-relay-ccc as the encapsulation type. To enable Frame Relay CCC at the logical interface level, include the encapsulation frame-relay-ccc statement at the [edit interfaces so-*fpc/pic/port* unit *unit-number*] hierarchy level.
- To configure Cisco HDLC CCC interface encapsulation, include the encapsulation statement at the [edit interfaces so-*fpc/pic/port*] hierarchy level and select cisco-hdlc-ccc as the encapsulation type.

- To configure PPP CCC interface encapsulation, include the encapsulation statement at the [edit interfaces so-*fpc/pic/port*] hierarchy level and select ppp-ccc as the encapsulation type.

```
[edit]
interfaces {
   so-0/0/0 {
      encapsulation (frame-relay-ccc | cisco-hdlc-ccc | ppp-ccc);
      unit 0 {
         encapsulation frame-relay-ccc;
      }
   }
}
```

Configuring a CCC Encapsulation and a Layer 2 Circuit Mode on CE-Facing ATM2 IQ Interfaces

On ATM2 IQ interfaces in a PE router, you need to configure two encapsulations to enable Layer 2 circuits: one at the [edit interfaces at-*fpc/pic/port*] hierarchy level and the other at the [edit chassis fpc *fpc-slot* pic *pic-slot*] hierarchy level. There are two types of ATM2 IQ Layer 2 circuits: cell-relay mode and ATM Adaptation Layer 5 (AAL5) mode. For both modes, you must specify the Physical Interface Card (PIC) type with the pic-type atm2 statement at the [edit interfaces at-*fpc/pic/port* atm-options] hierarchy level. You can configure only one mode per PIC at a time. If you need to enable both ATM2 IQ Layer 2 circuit modes in the same router, you must configure the different modes on different PICs.

To configure a cell-relay mode Layer 2 circuit, include the atm-l2circuit-mode cell statement at the [edit chassis fpc *fpc-slot* pic *pic-slot*] hierarchy level and the encapsulation atm-ccc-cell-relay statement at both the [edit interfaces at-*fpc/pic/port*] physical hierarchy level and the [edit interfaces at-*fpc/pic/port* unit *unit-number*] logical interface hierarchy level.

```
[edit]
chassis {
   fpc 0 {
      pic 1 {
         atm-l2circuit-mode {
            cell;
         }
      }
   }
}
interfaces {
   at-0/1/0 {
      encapsulation atm-ccc-cell-relay;
      atm-options {
         cell-bundle-size 4;
         pic-type atm2;
         vpi 0;
      }
      unit 0 {
         encapsulation atm-ccc-cell-relay;
```

```
            vci 32;
            cell-bundle-size 10;
        }
    }
}
```

For ATM2 IQ Layer 2 circuit cell-relay mode only, you can adjust the cell bundle size at the physical interface level and the logical interface level. To configure, include the cell-bundle-size statement at either the [edit interfaces at-*fpc/pic/port* atm-options] physical interface hierarchy level or the [edit interfaces at-*fpc/pic/port* unit *unit-number*] logical interface hierarchy level. If the statement is included at both levels, the logical interface setting takes precedence. The default value for cell bundle size is 1 and the maximum value is 190. If you configure the cell bundle size statement, you should configure the same value on all ATM2 IQ neighbors.

To configure an AAL5 mode Layer 2 circuit, include the atm-l2circuit-mode aal5 statement at the [edit chassis fpc*fpc-slot* pic *pic-slot*] hierarchy level and the encapsulation atm-ccc-vc-mux statement at the [edit interfaces at-*fpc/pic/port*] hierarchy level:

```
[edit]
chassis {
    fpc 1 {
        pic 2 {
            atm-l2circuit-mode {
                aal5;
            }
        }
    }
}
interfaces {
    at-1/2/0 {
        atm-options {
            pic-type atm2;
            vpi 0;
        }
        unit 0 {
            encapsulation atm-ccc-vc-mux;
            vci 32;
        }
    }
}
```

For more information on how to configure interfaces with CCC encapsulation types, see the *JUNOS MPLS Applications Configuration Guide* or the *JUNOS Network Interfaces Configuration Guide*.

Configuring the MPLS Family on Core Interfaces

Because LDP is used as the signaling protocol to transport MPLS labels across the core of the network, you must include the family mpls statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level. Include the statement on all router interfaces in the path from the local PE router to the remote PE router across the core network that transports the Layer 2 circuit traffic.

```
[edit]
interfaces {
   interface-name  {
      unit logical-unit-number {
         family inet {
            address ip-address/prefix;
         }
         family mpls;
      }
   }
}
```

Configuring the Layer 2 Circuit Neighbor Address and Virtual Circuit Identifier

After you enable the PE router interfaces with the proper encapsulations, you then configure Layer 2 circuits (also referred to as VCs) between two or more PE router neighbors. To configure a Layer 2 circuit, include the l2circuit statement at the [edit protocols] hierarchy level.

Each Layer 2 circuit is represented by a logical interface on the local PE router, the IP address of the remote PE router neighbor, and a virtual circuit identifier. The logical interface connects the local PE router to the local CE router. The loopback address and router ID of the PE neighbor is commonly the neighbor's IP address. This address is also the destination end-point of the LSP tunnel, which transports the Layer 2 circuit to the neighbor. The virtual circuit ID uniquely identifies the VC to a specific neighbor.

This combination of logical interface, neighbor address, and virtual circuit ID is used to map a particular LDP forwarding equivalence class (FEC) received from a specific neighbor to a local VC. The egress label is added to a table and is used for sending traffic on that VC between the CE routers.

Both ends of a Layer 2 circuit must use the same Layer 2 technology because the Layer 2 encapsulation type is carried in the LDP FEC. The encapsulation type from a received FEC is matched against the local encapsulation type of the VC. If there is a mismatch, the VC is not established.

To add the IP address of the remote PE router neighbor into a Layer 2 circuit, include the neighbor *ip-address* statement at the [edit protocols l2circuit] hierarchy level. To map the remote neighbor to the local interface that connects to the CE router, include the interface statement at the [edit protocols l2circuit neighbor *ip-address*] hierarchy level. To select the identifier for the virtual circuit, include the virtual-circuit-identifier statement at the [edit protocols l2circuit neighbor *ip-address* interface *interface-name*] hierarchy level. To disable default control word processing, include the no-control-word statement at the [edit protocols l2circuit neighbor *ip-address* interface *interface-name*] hierarchy level. Finally, to assign the Layer 2 circuit to a community, include the community *community-name*statement at the [edit protocols l2circuit neighbor *ip-address* interface *interface-name*] hierarchy level.

NOTE: On M-series routing platforms only, if you include the control-word statement at the [edit protocols l2circuit neighbor *address* interface *interface-name*] hierarchy level (the default setting for Layer 2 circuits), the software cannot rewrite MPLS EXP bits.

```
[edit]
protocols {
   l2circuit {
      traceoptions {
         file name [replace] [size size] [files files ] [nostamp];
         flag (error | topology | nlri | connections | route) [detail];
      }
      neighbor ip-address {
         interface interface-name {
            virtual-circuit-id identifier;
            no-control-word;
            community community-name;
         }
      }
   }
}
```

You do not need to specify the encapsulation type at the [edit protocols l2circuit] hierarchy level because it is already specified in the interface configuration.

Configuring LDP and an IGP to Transport Layer 2 Circuits

LDP is used as the signaling protocol to advertise the ingress MPLS label to the remote PE router. For this purpose, a remote LDP neighbor is established using the extended discovery mechanism described in RFC 3036, *LDP Specification*, and a session is established.

No new configuration is necessary in LDP because the LDP protocol recognizes the Layer 2 circuit configuration and initiates extended neighbor discovery for all Layer 2 circuit neighbors on the remote PE routers. This is very similar to the behavior of LDP when it is tunneled over RSVP. However, you must configure LDP on the lo0.0 interface for extended neighbor discovery to function correctly.

LDP relies on an underlying IGP, such as OSPF or IS-IS. Therefore, configure LDP and your IGP on all routers in the path from the local PE router to the remote PE router across the service provider backbone.

```
[edit]
protocols {
   ospf {
      traffic-engineering;
      area 0.0.0.0 {
         interface so-0/1/0.0;
         interface lo0.0;
      }
   }
   ldp {
      interface so-0/1/0.0;
```

```
        interface lo0.0;
    }
}
```

Option: Applying Traffic Engineering to a Layer 2 Circuit

To traffic engineer Layer 2 circuits over multiple LSPs, you must create a community, assign a set of Layer 2 circuits to that community, define a policy to send the community traffic over a desired LSP, and apply the policy to the forwarding table.

To create a community, include the community *community-name* statement at the [edit policy-options] hierarchy level. To assign a Layer 2 circuit to a community, include the community *community-name* statement at the [edit protocols l2circuit neighbor *ip-address* interface *interface-name*] hierarchy level. To create a policy that sends community traffic over a specific LSP, include the community *community-name*statement at the [edit policy-options policy-statement *policy-name* term *term-name* from] hierarchy level and the install-next-hop lsp *lsp-name* statement at the [edit policy-options policy-statement *policy-name* term *term-name* then] hierarchy level. To apply the policy to the forwarding table, include the export *policy-name* statement at the [edit routing-options forwarding-table] hierarchy level.

```
[edit]
routing-options {
    forwarding-table {
        export policy-name;
    }
}
protocols {
    l2circuit {
        neighbor ip-address {
            interface interface-name {
                virtual-circuit-id identifier;
                community community-name;
            }
        }
    }
}
policy-options {
    policy-statement policy-name {
        from community community-name;
        then {
            install-nexthop lsp lsp-name;
            accept;
        }
    }
}
community community-name members value;
```

Option: Mapping Layer 2 Protocol Control Information into a Layer 2 Circuit

The control word is defined in Internet draft draft-martini-l2circuit-encap-mpls-07.txt *Encapsulation Methods for Transport of Layer 2 Frames Over IP and MPLS Networks*.

It is a set of fields that carry Layer 2 control information across a Layer 2 circuit. The following control word support is available for PE routers:

- Frame Relay—To carry Frame Relay FECN/BECN information in a Layer 2 circuit control word, include the translate-fecn-and-becn statement at the [edit interfaces *interface-name* encapsulation frame-relay-ccc unit *unit-number* encapsulation frame-relay-ccc family ccc] hierarchy level. To carry Frame Relay DE information in a Layer 2 circuit control word, include the translate-discard-eligible statement at the [edit interfaces *interface-name* encapsulation frame-relay-ccc unit *unit-number* encapsulation frame-relay-ccc family ccc] hierarchy level.

```
[edit]
interfaces {
   interface-name {
      encapsulation frame-relay-ccc;
      unit 0 {
         encapsulation frame-relay-ccc;
         point-to-point;
         dlci 512;
         family ccc {
            translate-fecn-and-becn;
            translate-discard-eligible;
         }
      }
   }
}
```

- ATM AAL5 mode—For ATM2 IQ interfaces, the ATM AAL5 control word contains bit fields to carry sequence numbers, ATM cell loss priority (CLP), and explicit forward congestion indication (EFCI) information. When you configure ATM Layer 2 circuits, the control word carries the sequence number, CLP, and EFCI information by default. No additional configuration is necessary.
- ATM cell-relay mode—For ATM2 IQ interfaces, the ATM cell-relay control word supports sequence number processing only. Once you configure a cell-relay mode Layer 2 circuit, the sequence number information is carried by default. No additional configuration is necessary.

Option: Configuring APS for Layer 2 Circuits

To apply Automatic Protection Switching (APS) to a Layer 2 circuit, you must configure an APS working circuit and a protect circuit on PE router interfaces that support SONET options (such as SONET/SDH, ATM, and ATM2 IQ interfaces) and circuit cross-connect (CCC) encapsulation types. Then, you must configure the working circuit as the primary Layer 2 circuit interface and the protect circuit as the protected Layer 2 circuit interface. Note that APS only protects the PE-CE link and not the entire Layer 2 circuit.

To configure an APS working circuit, include the working-circuit statement at the [edit interfaces *interface-name* sonet-options aps] hierarchy level. To configure an APS protect circuit, include the protect-circuit statement at the [edit interfaces *interface-name* sonet-options aps] hierarchy level. To configure the primary Layer 2 circuit interface, include the interface statement at the [edit protocols l2circuit neighbor *ip-address*] hierarchy level. To configure the protected Layer 2 circuit interface, include the

protect-interface statement at the [edit protocols l2circuit neighbor *ip-address* interface *interface-name*] hierarchy level.

```
[edit]
interfaces {
   at-0/0/1 {
      description " APS protect circuit";
      encapsulation CCC-encapsulation-type;
      sonet-options {
         aps {
            protect-circuit name;
         }
      }
   }
   at-1/3/1 {
      description "APS working circuit";
      encapsulation CCC-encapsulation-type;
      sonet-options {
         aps {
            working-circuit name;
         }
      }
   }
}
protocols {
   l2circuit {
      neighbor ip-address {
         interface at-1/3/1.0 {
            protect-interface at-0/0/1.0;
            virtual-circuit-id number;
         }
      }
   }
}
```

Option: Configuring Layer 2 Circuit Trunk Mode on ATM2 IQ Interfaces

When you configure Layer 2 circuits on CE-facing ATM2 IQ interfaces in a PE router that connects to some vendors' ATM switches, you can create a trunk. The trunk bundles several ATM cell streams into one LSP, preserves the cell loss priority (CLP) and class-of-service (CoS) information of the cells within the experimental (EXP) bits of the MPLS header, and provides network-to-network interface (NNI) or user-to-network interface (UNI) information within a proprietary header. A physical interface supports a total of 32 logical trunks in NNI mode and 8 logical trunks when you use the UNI option. To configure a trunk, include the trunk statement at the [edit chassis fpc *fpc-slot* pic *pic-slot* atm-l2circuit-mode] hierarchy level, select NNI or UNI mode with the nni or uni statement, and specify a number of bits in the ATM header that will carry an identifier with the id-width statement. You can choose a value from 1 through 8 for the identifier width.

```
[edit]
chassis {
   fpc fpc-slot {
      pic pic-clot {
```

```
        atm-l2circuit-mode {
            trunk {
                (nni | uni) {
                    id-width number;
                }
            }
        }
    }
}
```

You can also configure several trunk options at the [edit interfaces at-*fpc/pic/port* unit *unit-number*] hierarchy level:

- To specify an ATM interface as the control channel for a Layer 2 circuit trunk, include the control-channel statement at the [edit interfaces at-*fpc/pic/port* unit *unit-number*] hierarchy level.
- To specify a trunk identifier, include the trunk-id statement at the [edit interfaces at-*fpc/pic/port* unit *unit-number*] hierarchy level. Trunk ID values range from 0 through 31.
- To configure the amount of bandwidth reserved for the trunk, include the trunk-bandwidth statement at the [edit interfaces at-*fpc/pic/port* unit *unit-number*] hierarchy level and specify a value from 1,000,000 bps (1 Mbps) through 542,526,792 bps.
- To apply a CoS scheduler map to the trunk, include the atm-scheduler-map statement at the [edit interfaces at-*fpc/pic/port* unit *unit-number*] hierarchy level. This statement must reference an ATM2 IQ interface-based scheduler map at the [edit interfaces at-*fpc/pic/port* atm-options scheduler-maps *map-name*] hierarchy level.

```
interfaces {
    at-fpc/pic/port {
        atm-options {
            pic-type atm2;
            scheduler-maps {
                map-name {
                    ...
                }
            }
            unit unit-number {
                atm-scheduler-map map-name;
                control-channel;
                trunk-id id-number;
                trunk-bandwidth bandwidth;
            }
        }
    }
}
```

You can configure a variety of CoS-related statements for an ATM2 IQ interface-based scheduler map. To select the CoS mode used for virtual circuits, include the vc-cos-mode statement at the [edit interfaces at-*fpc/pic/port* atm-options scheduler-maps

map-name] hierarchy level. To specify forwarding class settings, include the priority, transmit-weight, and epd-threshold statements at the [edit interfaces at-*fpc/pic/port* atm-options scheduler-maps *map-name* forwarding-class *class-name*] hierarchy level. For more information about CoS, see the *JUNOS Class of Service Configuration Guide*.

```
[edit]
interfaces {
   at-fpc/pic/port {
      atm-options {
         pic-type atm2;
         scheduler-maps {
            map-name {
               vc-cos-mode (alternate | strict);
               forwarding-class  class-name {
                  priority (high | low);
                  transmit-weight (cells number-of-cells | percent percentage);
                  epd-threshold plp0-threshold  plp1 plp1-threshold;
               }
            }
         }
      }
   }
}
```

Option: Reserving LSP Bandwidth for a Layer 2 Circuit

You can specify the amount of bandwidth in bytes per second that must be available on an LSP for a specific Layer 2 circuit. By using a bandwidth constraint for a Layer 2 circuit, the router performs a type of call admission control. If an LSP exists that contains the required bandwidth, the Layer 2 circuit is established. If the bandwidth is not available on an LSP, the Layer 2 circuit is not established.

To configure bandwidth requirements for a Layer 2 circuit, include the bandwidth statement at the [edit protocols mpls label-switched-path *lsp-name*] hierarchy level and the [edit protocols l2circuit neighbor *ip-address* interface *interface-name*] hierarchy level.

```
[edit]
protocols {
   mpls {
      label-switched-path lsp-name {
         bandwidth traffic-class bytes-per-second;
      }
   }
   l2circuit {
      neighbor ip-address {
         interface interface-name {
            bandwidth bytes-per-second;
         }
      }
   }
}
```

You can also assign minimum bandwidth requirements for class-of-service (CoS) queues within a Layer 2 circuit and its corresponding LSP. Class type (CT) queues 0,

1, 2, and 3 in a Layer 2 circuit or LSP correspond to the standard four CoS queues available on M-series, MX-series, and T-series routing platforms. To enable mapping of class type queues to the standard CoS queues, include the bandwidth-model statement at the [edit protocols mpls diffserv-te] hierarchy level. To assign specific bandwidth requirements to each class type queue, include the ct0, ct1, ct2, and ct3 statements at the [edit protocols mpls label-switched-path *lsp-name* bandwidth] hierarchy level and the [edit protocols l2circuit neighbor *ip-address* interface *interface-name* bandwidth] hierarchy level.

```
[edit]
protocols {
   mpls {
      diffserv-te {
         bandwidth-model extended-mam;
      }
      label-switched-path lsp-name {
         bandwidth {
            ct0 100m;
            ct1 100m;
            ct2 50m;
            ct3 5m;
         }
      }
   }
   l2circuit {
      neighbor ip-address {
         interface interface-name {
            bandwidth {
               ct0 100m;
               ct1 100m;
               ct2 50m;
               ct3 5m;
            }
         }
      }
   }
}
```

For more information about class of service, see the *JUNOS Class of Service Configuration Guide*.

Option: Selecting an MTU for a Layer 2 Circuit

To configure the MTU for each individual Layer 2 circuit, include the mtu statement at the [edit protocols l2circuit neighbor *ip-address* interface *interface-name*] hierarchy level. If the MTU setting between Layer 2 circuit neighbors does not match, the Layer 2 circuit is torn down.

NOTE: If you configure an MTU value for an ATM cell relay interface on an ATM2 PIC and simultaneously configure an MTU value for a Layer 2 circuit that uses the same ATM2 PIC, the MTU value for the Layer 2 circuit takes precedence when calculating the cell bundle size and is advertised to Layer 2 circuit neighbors.

Option: Configuring Local Interface Switching for a Layer 2 Circuit

You can terminate a Layer 2 circuit locally on an ingress PE router. To configure a locally terminated circuit, include the local-switching statement at the [edit protocols l2circuit] hierarchy level. Select the Layer 2 circuit interfaces you want to connect locally, specify any APS protect interfaces, and configure an end interface. To select the Layer 2 circuit interfaces that are connected locally, include the interface statement at the [edit protocols l2circuit local-switching] hierarchy level. To configure an end interface, include the end-interface statement at the [edit protocols l2circuit local-switching interface *interface-name*] hierarchy level. To specify APS protect interfaces, include the protect-interface statement at the [edit protocols l2circuit local-switching interface *interface-name*] or [edit protocols l2circuit local-switching interface *interface-name* end-interface *interface-name*] hierarchy levels.

```
[edit]
protocols {
   l2circuit {
      local-switching {
         interface interface1 {
            protect-interface interface2;
            end-interface interface3 {
               protect-interface interface4;
            }
         }
         interface interface5 {
            protect-interface interface6;
            end-interface interface7 {
               protect-interface interface8;
            }
         }
      }
   }
}
```

Option: Configuring Layer 2 Circuits Simultaneously over RSVP and LDP LSPs

You can configure a Layer 2 circuit simultaneously over an RSVP LSP and an LDP LSP between the same two routing platforms. To accomplish this, do the following:

- Configure two loopback addresses—a primary and a secondary loopback.
- Configure the RSVP LSP using the primary loopback address.
- Configure the LDP LSP using the secondary loopback address. You can accomplish this by advertising the secondary loopback address as a forwarding equivalence class (FEC) in LDP.
- Configure the transport tunnel endpoint (also known as a packet-switched network [PSN] tunnel endpoint in the IETF drafts) to be used for transporting the Layer 2 circuit traffic. To configure the tunnel endpoint, include the psn-tunnel-endpoint statement at the [edit protocols l2circuit] hierarchy level. By default, the tunnel endpoint is the same as the address of the Layer 2 circuit neighbor.

To verify that your configuration is operational, issue the show l2circuit connections command. This command has been enhanced to display tunnel endpoints. For more information about configuring Layer 2 circuits simultaneously over RSVP and LDP LSPs, see the *JUNOS VPNs Configuration Guide*.

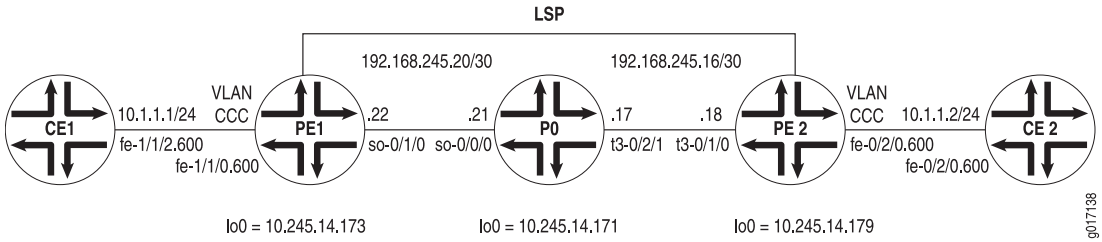
Layer 2 Circuit Configuration Examples

This section contains configuration examples and commands you can issue to verify Layer 2 circuit configurations:

- Example: Ethernet-Based Layer 2 Circuit Configuration on page 531
- Example: SONET/SDH-Based Layer 2 Circuit Configuration on page 538
- Example: ATM2 IQ-Based Layer 2 Circuit Configuration on page 543
- Example: Layer 2 Circuit Traffic Engineering over Multiple LSPs Configuration on page 552
- Example: APS for a Layer 2 Circuit Configuration on page 562

Example: Ethernet-Based Layer 2 Circuit Configuration

Figure 55: Ethernet-Based Layer 2 Circuit Topology Diagram



In Figure 55 on page 531, a Layer 2 circuit is established between routers PE1 and PE2 to deliver Layer 2 traffic between customer routers CE1 and CE2. A Layer 2 circuit VC connection is configured on the PE routers only. No special configuration is required on the CE routers, and the provider core P0 router only requires MPLS and LDP on the appropriate interfaces to enable labels to be shared between the PE routers.

On Router CE1, configure the Fast Ethernet interface to handle VLAN traffic. Be sure to use the same VLAN ID both here and on the Fast Ethernet interface of Router CE2.

Router CE1

```
[edit]
interfaces {
   fe-1/1/2 {
      description "to PE1 fe-1/1/0";
      vlan-tagging;
      unit 600 {
         vlan-id 600; # Be sure this VLAN ID matches the VLAN ID of your CE neighbor.
         family inet {
            address 10.1.1.1/24;
         }
```

```
        }
    }
}
```

On Router PE1, configure the Ethernet-based CE-facing interface with the CCC encapsulation type of your choice. If you use VLAN CCC, include the vlan-tagging statement at the [edit interfaces *ethernet-interface-fpc*/*pic* /*port*] hierarchy level. Also, include the encapsulation vlan-ccc statement at both the [edit interfaces *ethernet-interface-fpc*/*pic*/*port*] and [edit interfaces *ethernet-interface-fpc* /*pic*/*port* unit *unit-number*] hierarchy levels.

Establish your Layer 2 circuit with configuration of the l2circuit statement at the [edit protocols] hierarchy level. Remember to include in your Layer 2 circuit configuration the IP address of your remote PE neighbor (usually the loopback address of the neighbor), the interface connected to the CE router, and a virtual circuit identifier for this VC. Then, configure MPLS, LDP, and an IGP (such as OSPF) to enable signaling for your Layer 2 circuit.

Router PE1

```
[edit]
interfaces {
    so-0/1/0 {
        description "to P0 so-0/0/0";
        unit 0 {
            family inet {
                address 192.168.245.22/30;
            }
            family mpls; # Include the MPLS family on core-facing interfaces.
        }
    }
    fe-1/1/0 {
        description "to CE1 fe-1/1/2";
        vlan-tagging;
        encapsulation vlan-ccc; # Configure CCC encapsulation on CE-facing interfaces.
        unit 600 {
            encapsulation vlan-ccc; # Enable this encapsulation on the logical interface.
            vlan-id 600;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.245.14.173/32;
            }
        }
    }
}
protocols {
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface so-0/1/0.0;
            interface lo0.0;
        }
    }
    ldp { # LDP is required as the signaling protocol for Layer 2 circuits.
```

```
        interface so-0/1/0.0;
        interface lo0.0;# You must include the loopback address in LDP.
    }
    l2circuit {
        neighbor 10.245.14.179 { # This points to the loopback of the PE neighbor.
            interface fe-1/1/0.600 { # Here you include the local CE-facing interface.
                virtual-circuit-id 5; # Be sure this ID matches the ID of your PE neighbor.
            }
        }
    }
}
```

On Router P0, configure LDP, MPLS, and OSPF on the interfaces connected to the PE routers. The core router provides the MPLS backbone needed to tunnel Layer 2 traffic from the ingress PR router to the egress PE router.

Router P0

```
[edit]
interfaces {
    so-0/0/0 {
        description "to PE1 so-0/1/0";
        unit 0 {
            family inet {
                address 192.168.245.21/30;
            }
            family mpls; # Include the MPLS family on core interfaces.
        }
    }
    t3-0/2/1 {
        description "to PE2 t3-0/1/0";
        unit 0 {
            family inet {
                address 192.168.245.17/30;
            }
            family mpls; # Include the MPLS family on core interfaces.
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.245.14.171/32;
            }
        }
    }
}
protocols {
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface t3-0/2/1.0;
        }
    }
    ldp { # LDP is required as the signaling protocol for Layer 2 circuits.
        interface so-0/0/0.0;
        interface t3-0/2/1.0;
    }
```

```
}
```

On Router PE2, complete the Layer 2 circuit by configuring statements to match those previously set on Router PE1. Configure the Ethernet-based CE-facing interface with CCC encapsulation. Again, you must include the vlan-tagging statement at the [edit interfaces *ethernet-interface-fpc/pic/port*] hierarchy level when you use VLAN CCC. Also, include the encapsulation vlan-ccc statement at both the [edit interfaces *ethernet-interface-fpc/pic/port*] and [edit interfaces *ethernet-interface-fpc/pic/port* unit *unit-number*] hierarchy levels.

Establish your Layer 2 circuit with configuration of the l2circuit statement at the [edit protocols] hierarchy level. Remember to include in your Layer 2 circuit configuration the IP address of your remote PE neighbor (Router PE1), the virtual circuit identifier previously configured on Router PE1, and the interface connected to the CE router. Finally, configure MPLS, LDP, and OSPF to enable signaling for your Layer 2 circuit.

Router PE2

```
[edit]
interfaces {
   t3-0/1/0 {
      description "P0 t3-0/2/1";
      unit 0 {
         family inet {
            address 192.168.245.18/30;
         }
         family mpls; # Include the MPLS family on core-facing interfaces.
      }
   }
   fe-0/2/0 {
      description "to CE2 fe-0/2/0";
      vlan-tagging;
      encapsulation vlan-ccc; # Configure CCC encapsulation on CE-facing interfaces.
      unit 600 {
         encapsulation vlan-ccc; # Enable this encapsulation on the logical interface.
         vlan-id 600;
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.245.14.179/32;
         }
      }
   }
}
protocols {
   ospf {
      traffic-engineering;
      area 0.0.0.0 {
         interface t3-0/1/0.0;
         interface lo0.0;
      }
   }
   ldp { # LDP is required as the signaling protocol for Layer 2 circuits.
      interface t3-0/1/0.0;
      interface lo0.0; # You must include the loopback address in LDP.
```

```
    }
    l2circuit {
      neighbor 10.245.14.173 { # This points to the loopback of the PE neighbor.
        interface fe-0/2/0.600 { # Here you include the local CE-facing interface.
          virtual-circuit-id 5; # Be sure this ID matches the ID of your PE neighbor.
        }
      }
    }
}
```

On Router CE2, configure the Fast Ethernet interface to handle VLAN traffic. Be sure to use the same VLAN ID on this interface as the one seen on the Fast Ethernet interface of Router CE1.

Router CE2

```
[edit]
interfaces {
  fe-0/2/0 {
    description "to PE2 fe-0/2/0";
    vlan-tagging;
    unit 600 {
      vlan-id 600; # Be sure this VLAN ID matches the VLAN ID of your CE neighbor.
      family inet {
        address 10.1.1.2/24;
      }
    }
  }
}
```

Verifying Your Work

To verify proper operation of Layer 2 circuits, use the following commands:

- ping mpls l2circuit interface*interface-name*
- ping mpls l2circuit virtual-circuit *virtual-circuit-id* neighbor *ip-address*
- show l2circuit connections
- Options: [brief] | [down] | [extensive] | [history] | [instance] | [local-site] | [remote-site] |[status] | [summary] | [up] | [up-down]
- show ldp database

In addition to displaying bindings for IP prefixes, the show ldp database command also displays the bindings for the Layer 2 FECs.

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

- Router PE1 Status on page 536
- Router P0 Status on page 536
- Router PE2 Status on page 537

Router PE1 Status

```
user@PE1> show l2circuit connections
Layer-2 Circuit Connections:

Legend for connection status (St)   Legend for interface status
EI -- encapsulation invalid         UP -- operational
MM -- mtu mismatch                  Dn -- down
EM -- encapsulation mismatch        NP -- no present
OL -- no outgoing label             DS -- disabled
Dn -- down                          WE -- wrong encapsulation
VC-Dn -- Virtual circuit Down       UN -- uninitialized
UP -- operational
XX -- unknown

Neighbor: 10.245.14.179
    Interface                 Type  St     Time last up          # Up trans
    fe-1/1/0.600 (vc 5)       rmt   Up     Nov 30 00:54:55 2001           1
      Local interface: fe-1/1/0.600, Status: Up,  Encapsulation: VLAN
      Remote PE: 10.245.14.179,  Negotiated control-word: Yes (Null)
      Incoming label: 100007, Outgoing label: 100000

user@PE1>  show ldp database

Input label database, 10.245.14.173:0-10.245.14.171:0
  Label     Prefix
 100019     10.245.14.173/32
 100020     10.245.14.179/32
      3     10.245.14.171/32

Output label database, 10.245.14.173:0-10.245.14.171:0
  Label     Prefix
 100009     10.245.14.179/32
      3     10.245.14.173/32
 100008     10.245.14.171/32

Input label database, 10.245.14.173:0-10.245.14.179:0
  Label     Prefix
 100001     10.245.14.171/32
 100002     10.245.14.173/32
      3     10.245.14.179/32
 100000     L2CKT VLAN VC 5

Output label database, 10.245.14.173:0-10.245.14.179:0
  Label     Prefix
 100009     10.245.14.179/32
      3     10.245.14.173/32
 100008     10.245.14.171/32
 100007     L2CKT VLAN VC 5
```

Router P0 Status

```
user@P0> show ldp database

Input label database, 10.245.14.171:0-10.245.14.173:0
  Label     Prefix
```

```
      3     10.245.14.173/32
 100009     10.245.14.179/32
 100008     10.245.14.171/32

Output label database, 10.245.14.171:0-10.245.14.173:0
  Label     Prefix
 100019     10.245.14.173/32
 100020     10.245.14.179/32
      3     10.245.14.171/32

Input label database, 10.245.14.171:0-10.245.14.179:0
  Label     Prefix
 100001     10.245.14.171/32
      3     10.245.14.179/32
 100002     10.245.14.173/32

Output label database, 10.245.14.171:0-10.245.14.179:0
  Label     Prefix
 100019     10.245.14.173/32
 100020     10.245.14.179/32
      3     10.245.14.171/32
```

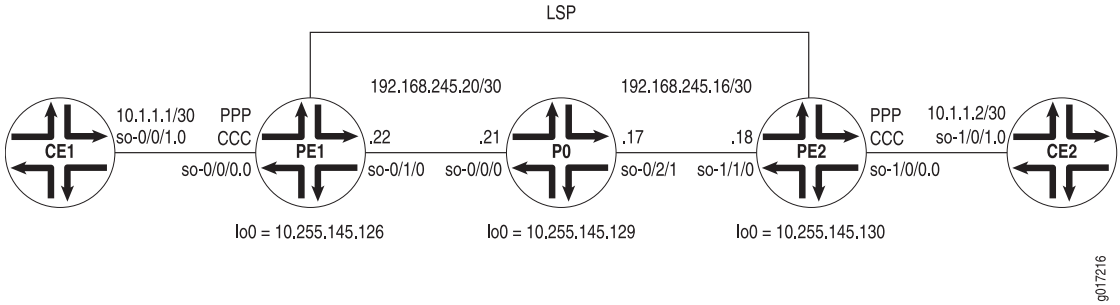
Router PE2 Status

```
user@PE2> show l2circuit connections

Layer-2 Circuit Connections:

Legend for connection status (St)   Legend for interface status
EI -- encapsulation invalid         UP -- operational
MM -- mtu mismatch                  Dn -- down
EM -- encapsulation mismatch        NP -- no present
OL -- no outgoing label             DS -- disabled
Dn -- down                          WE -- wrong encapsulation
VC-Dn -- Virtual circuit Down       UN -- uninitialized
UP -- operational
XX -- unknown

Neighbor: 10.245.14.173
    Interface                 Type  St     Time last up          # Up trans
    fe-0/2/0.600 (vc 5)       rmt   Up     Nov 30 00:54:54 2001           1
      Local interface: fe-0/2/0.600, Status: Up,  Encapsulation: VLAN
      Remote PE: 10.245.14.173,  Negotiated control-word: Yes (Null)
      Incoming label: 100000, Outgoing label: 100007

user@PE2>  show ldp database

Input label database, 10.245.14.179:0-10.245.14.171:0
  Label     Prefix
 100019     10.245.14.173/32
      3     10.245.14.171/32
 100020     10.245.14.179/32

Output label database, 10.245.14.179:0-10.245.14.171:0
  Label     Prefix
 100001     10.245.14.171/32
 100002     10.245.14.173/32
      3     10.245.14.179/32
```

```
Input label database, 10.245.14.179:0-10.245.14.173:0
  Label     Prefix
      3     10.245.14.173/32
 100008     10.245.14.171/32
 100009     10.245.14.179/32
 100007     L2CKT VLAN VC 5

Output label database, 10.245.14.179:0-10.245.14.173:0
  Label     Prefix
 100001     10.245.14.171/32
 100002     10.245.14.173/32
      3     10.245.14.179/32
 100000     L2CKT VLAN VC 5
```

Example: SONET/SDH-Based Layer 2 Circuit Configuration

Figure 56: SONET/SDH-Based Layer 2 Circuit Topology Diagram



In this second Layer 2 circuit example shown in Figure 56 on page 538, you configure a Layer 2 circuit for a SONET/SDH interface encapsulated with PPP.

On Router CE1, configure the SONET/SDH interface to handle PPP traffic. Be sure to use the same IP address prefix both here and on the SONET/SDH interface of Router CE2.

Router CE1

```
[edit]
interfaces {
   so-0/0/1 {
      description "to PE1 so-0/0/0";
      encapsulation ppp;
      unit 0 {
         family inet {
            address 10.1.1.1/30;
         }
      }
   }
}
```

On Router PE1, configure the PPP-based CE-facing interface with PPP CCC encapsulation. Establish your Layer 2 circuit with configuration of the l2circuit statement at the [edit protocols] hierarchy level. Remember to include in your Layer 2 circuit configuration the IP address of your remote PE neighbor (usually the loopback address of the neighbor), the interface connected to the CE router, and a virtual

circuit identifier for this VC. Then, configure MPLS, LDP, and an IGP (such as OSPF) to enable signaling for your Layer 2 circuit.

Router PE1

```
[edit]
interfaces {
   so-0/0/0 {
      description "to CE1 so-0/0/1";
      encapsulation ppp-ccc; # Configure CCC encapsulation on CE-facing interfaces.
      unit 0;
   }
   so-0/1/0 {
      description "to P0 so-0/0/0";
      unit 0 {
         family inet {
            address 192.168.245.22/30;
         }
         family mpls; # Include the MPLS family on core-facing interfaces.
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.255.145.126/32;
         }
      }
   }
}
protocols {
   ospf {
      traffic-engineering;
      area 0.0.0.0 {
         interface so-0/1/0.0;
         interface lo0.0;
      }
   }
   ldp { # LDP is required as the signaling protocol for Layer 2 circuits.
      interface so-0/1/0.0;
      interface lo0.0;
   }
   l2circuit {
      neighbor 10.255.145.130 { # This points to the loopback of the PE neighbor.
         interface so-0/0/0.0 { # Here you include the local CE-facing interface.
            virtual-circuit-id 1; # Be sure this ID matches the ID of your PE neighbor.
         }
      }
   }
}
```

On Router P0, configure LDP, MPLS, and OSPF on the interfaces connected to the PE routers. The core router provides the MPLS backbone needed to tunnel Layer 2 traffic from the ingress PR router to the egress PE router.

Router P0

```
[edit]
interfaces {
   so-0/0/0 {
      description "to PE1 so-0/1/0";
```

```
        unit 0 {
          family inet {
            address 192.168.245.21/30;
          }
          family mpls; # Include the MPLS family on core interfaces.
        }
      }
      so-0/2/1 {
        description "to PE2 so-1/1/0";
        unit 0 {
          family inet {
            address 192.168.245.17/30;
          }
          family mpls; # Include the MPLS family on core interfaces.
        }
      }
      lo0 {
        unit 0 {
          family inet {
            address 10.255.145.129/32;
          }
        }
      }
    }
    protocols {
      ospf {
        traffic-engineering;
        area 0.0.0.0 {
          interface so-0/0/0.0;
          interface so-0/2/1.0;
        }
      }
      ldp { # LDP is required as the signaling protocol for Layer 2 circuits.
        interface so-0/0/0.0;
        interface so-0/2/1.0;
      }
    }
```

On Router PE2, complete the Layer 2 circuit by configuring statements to match those previously set on Router PE1. Configure the PPP-based CE-facing interface with PPP CCC encapsulation. Complete your Layer 2 circuit with configuration of the l2circuit statement at the [edit protocols] hierarchy level. Remember to include in your Layer 2 circuit configuration the IP address of your remote PE neighbor (Router PE1), the interface connected to the CE router, and a virtual circuit identifier for this VC. Then, configure MPLS, LDP, and an IGP (such as OSPF) to enable signaling for your Layer 2 circuit.

Router PE2

```
[edit]
interfaces {
  so-1/0/0 {
    description "to CE1 so-1/0/1";
    encapsulation ppp-ccc; # Configure CCC encapsulation on CE-facing interfaces.
    unit 0;
  }
  so-1/1/0 {
```

```
        description "to P0 so-0/2/1";
        unit 0 {
          family inet {
            address 192.168.245.18/30;
          }
          family mpls; # Include the MPLS family on core-facing interfaces.
        }
      }
      lo0 {
        unit 0 {
          family inet {
            address 10.255.145.130/32;
          }
        }
      }
    }
    protocols {
      ospf {
        traffic-engineering;
        area 0.0.0.0 {
          interface so-1/1/0.0;
          interface lo0.0;
        }
      }
      ldp { # LDP is required as the signaling protocol for Layer 2 circuits.
        interface so-1/1/0.0;
        interface lo0.0;
      }
      l2circuit {
        neighbor 10.255.145.126 { # This points to the loopback of the PE neighbor.
          interface so-1/0/0.0 { # Here you include the local CE-facing interface.
            virtual-circuit-id 1; # Be sure this ID matches the ID of your PE neighbor.
          }
        }
      }
    }
```

On Router CE2, configure the SONET/SDH interface to handle PPP traffic. Be sure to use the same IP address prefix both here and on the SONET/SDH interface of Router CE1.

Router CE2

```
[edit]
interfaces {
  so-1/0/1 {
    description "to PE2 so-1/0/0";
    encapsulation ppp;
    unit 0 {
      family inet {
        address 10.1.1.2/30;
      }
    }
  }
}
```

Verifying Your Work

To verify proper operation of Layer 2 circuits, use the following commands:

- ping mpls l2circuit interface *interface-name*
- ping mpls l2circuit virtual-circuit *virtual-circuit-id* neighbor *ip-address*
- show l2circuit connections
- Options: [brief] | [down] | [extensive] | [history] | [instance] | [local-site] | [remote-site] |[status] | [summary] | [up] | [up-down]
- show ldp database
- show route table

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

```
user@PE1> show l2circuit connections
Layer-2 Circuit Connections:

Legend for connection status (St)
EI -- encapsulation invalid      NP -- interface not present
MM -- mtu mismatch               Dn -- down
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
OL -- no outgoing label          XX -- unknown
NC -- intf encaps not CCC/TCC

Legend for interface status
Up -- operational
Dn -- down

Neighbor: 10.255.145.130
     Interface                 Type  St     Time last up          # Up trans
     so-0/0/0.0 (vc 1)         rmt   Up     Jan 26 14:13:54 2003           1
       Local interface: so-0/0/0.0, Status: Up, Encapsulation: PPP
       Remote PE: 10.255.145.130, Negotiated control-word: Yes (Null)
       Incoming label: 100000, Outgoing label: 100000

user@PE1>  show ldp database l2circuit
Input label database, 10.255.145.126:0--10.255.145.130:0
   Label     Prefix
  100000     L2CKT CtrlWord PPP VC 1

Output label database, 10.255.145.126:0--10.255.145.130:0
   Label     Prefix
  100000     L2CKT CtrlWord PPP VC 1

user@PE1> show ldp database l2circuit detail
Input label database, 10.255.145.126:0--10.255.145.130:0
   Label     Prefix
  100000     L2CKT CtrlWord PPP VC 1
             State: Active
             Age: 5:37
```

```
Output label database, 10.255.145.126:0--10.255.145.130:0
   Label     Prefix
  100000     L2CKT CtrlWord PPP VC 1
             State: Active
             Age: 5:37

user@PE1> show route table mpls.0

mpls.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both
0                  *[MPLS/0] 00:05:04, metric 1
                      Receive
1                  *[MPLS/0] 00:05:04, metric 1
                      Receive
2                  *[MPLS/0] 00:05:04, metric 1
                      Receive
100000             *[L2VPN/7] 00:04:50
                     > via so-0/0/0.0, Pop       Offset: 4
100016             *[LDP/9] 00:04:52, metric 1
                     > via so-0/1/0.0, Pop
100016(S=0)        *[LDP/9] 00:04:52, metric 1
                     > via so-0/1/0.0, Pop
so-0/0/0.0         *[L2VPN/7] 00:04:50
                     > via so-0/1/0.0, Push 100000 Offset: -4
```

Example: ATM2 IQ-Based Layer 2 Circuit Configuration

Figure 57: ATM2 IQ-Based Layer 2 Circuit Topology Diagram

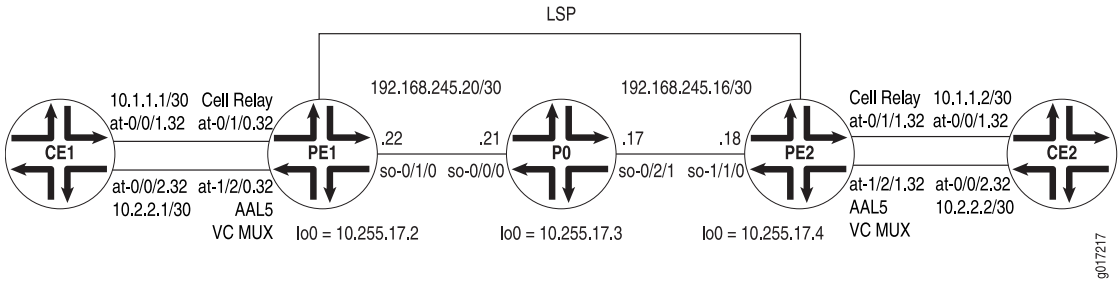


Figure 57 on page 543 shows a similar network topology to our previous two examples. In this example, Routers PE1 and PE2 use ATM cell-relay mode on a CE-facing interface and ATM AAL5 mode on a second CE-facing interface.

On Router CE1, configure the ATM2 IQ interfaces to handle ATM traffic. Interface at-0/0/1 handles standard ATM traffic while interface at-0/0/2 handles AAL5 traffic.

Router CE1

```
[edit]
interfaces {
   at-0/0/1 {
      description "to PE1 at-0/1/0";
      atm-options {
         pic-type atm2; # Layer 2 circuits are compatible with
         vpi 0; # ATM2 IQ interfaces.
      }
      unit 0 {
```

```
            vci 32;
            family inet {
               address 10.1.1.1/30;
            }
         }
      }
      at-0/0/2 {
         description "to PE1 at-1/2/0";
         atm-options {
            pic-type atm2; # Layer 2 circuits are compatible with
            vpi 0; # ATM2 IQ interfaces.
         }
         unit 0 {
            encapsulation atm-vc-mux;
            vci 32;
            family inet {
               address 10.2.2.1/30;
            }
         }
      }
   }
```

On Router PE1, configure the ATM2 IQ-based CE-facing interfaces: one with ATM cell-relay mode CCC encapsulation and the other with ATM VC multiplexing CCC encapsulation. Also enable the corresponding Layer 2 circuit modes at the [edit chassis] hierarchy level. In this case, you must configure cell-relay mode on Physical Interface Card (PIC) 1 in Flexible PIC Concentrator (FPC) 0 and AAL5 mode on PIC 2 in FPC 1.

Establish your Layer 2 circuit with configuration of the l2circuit statement at the [edit protocols] hierarchy level. Remember to include in your Layer 2 circuit configuration the IP address of your remote PE neighbor (usually the loopback address of the neighbor), the interfaces connected to the CE router, and a virtual circuit identifier for each VC. In this case, you will establish one VC for cell-relay mode traffic and a second VC for AAL5 traffic. Then, configure MPLS, LDP, and an IGP (such as OSPF) to enable signaling for your Layer 2 circuit.

Router PE1

```
[edit]
chassis {
   fpc 0 {
      pic 1 {
         atm-l2circuit-mode {
            cell; # This dedicates FPC 0 PIC 1 to cell-relay mode.
         }
      }
   }
   fpc 1 {
      pic 2 {
         atm-l2circuit-mode {
            aal5; # This dedicates FPC 1 PIC 2 to AAL5 mode.
         }
      }
   }
}
interfaces {
```

```
    at-0/1/0 {
        description "to CE1 at-0/0/1";
        encapsulation atm-ccc-cell-relay; # Cell-relay requires cell-relay encapsulation.
        atm-options {
            cell-bundle-size 4; # This sets the cell bundle size for the interface.
            pic-type atm2; # Layer 2 circuits are compatible with
            vpi 0; # ATM2 IQ interfaces.
        }
        unit 0 {
            encapsulation atm-ccc-cell-relay; # Encapsulation for the logical interface.
            vci 32;
            cell-bundle-size 10; # The cell bundle size for the logical interface overrides
        }       # the physical interface setting.
    }
    at-1/2/0 {
        description "to CE1 at-0/0/2";
        atm-options {
            pic-type atm2; # Layer 2 circuits are compatible with
            vpi 0; # ATM2 IQ interfaces.
        }
        unit 0 {
            encapsulation atm-ccc-vc-mux; # AAL5 requires CCC VC MUX encapsulation.
            vci 32;
        }
    }
    so-0/1/0 {
        description "to P0 so-0/0/0";
        unit 0 {
            family inet {
                address 192.168.245.22/30;
            }
            family mpls; # Include the MPLS family on core-facing interfaces.
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.17.2/32;
            }
        }
    }
}
protocols {
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface so-0/1/0.0;
            interface lo0.0;
        }
    }
    ldp { # LDP is required as the signaling protocol for Layer 2 circuits.
        interface so-0/1/0.0;
        interface lo0.0;
    }
    l2circuit {
        neighbor 10.255.17.4 { # This points to the loopback of the PE neighbor.
```

```
            interface at-0/1/0.32 { # Here you include the local CE-facing interface.
               virtual-circuit-id 1; # Be sure this ID matches the ID of your PE neighbor.
            }
         }
         neighbor 10.255.17.4 { # This points to the loopback of the PE neighbor.
            interface at-1/2/0.32 { # Here you include the local CE-facing interface.
               virtual-circuit-id 2; # Be sure this ID matches the ID of your PE neighbor.
            }
         }
      }
   }
```

On Router P0, configure LDP, MPLS, and OSPF on the interfaces connected to the PE routers. The core router provides the MPLS backbone needed to tunnel Layer 2 traffic from the ingress PR router to the egress PE router.

Router P0

```
[edit]
interfaces {
   so-0/0/0 {
      description "to PE1 so-0/1/0";
      unit 0 {
         family inet {
            address 192.168.245.21/30;
         }
         family mpls; # Include the MPLS family on core interfaces.
      }
   }
   so-0/2/1 {
      description "to PE2 so-1/1/0";
      unit 0 {
         family inet {
            address 192.168.245.17/30;
         }
         family mpls; # Include the MPLS family on core interfaces.
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.255.17.3/32;
         }
      }
   }
}
protocols {
   ospf {
      traffic-engineering;
      area 0.0.0.0 {
         interface so-0/0/0.0;
         interface so-0/2/1.0;
      }
   }
   ldp { # LDP is required as the signaling protocol for Layer 2 circuits.
      interface so-0/0/0.0;
      interface so-0/2/1.0;
   }
```

```
}
```

On Router PE2, complete the Layer 2 circuit by configuring statements to match those previously set on Router PE1.

Configure the ATM2 IQ-based CE-facing interfaces: one with ATM cell-relay mode CCC encapsulation and the other with ATM VC multiplexing CCC encapsulation. Also enable the corresponding Layer 2 circuit modes at the [edit chassis] hierarchy level. In this case, you must configure cell-relay mode on PIC 1 in FPC 0 and AAL5 mode on PIC 2 in FPC 1.

Complete your Layer 2 circuit with configuration of the l2circuit statement at the [edit protocols] hierarchy level. Remember to include in your Layer 2 circuit configuration the IP address of your remote PE neighbor (Router PE1), the interfaces connected to the CE router, and a virtual circuit identifier for each VC. In this case, you will establish one VC for cell-relay mode traffic and a second VC for AAL5 traffic. Then, configure MPLS, LDP, and an IGP (such as OSPF) to enable signaling for your Layer 2 circuit.

Router PE2

```
[edit]
chassis {
   fpc 0 {
      pic 1 {
         atm-l2circuit-mode {
            cell; # This dedicates FPC 0 PIC 1 to cell-relay mode.
         }
      }
   }
   fpc 1 {
      pic 2 {
         atm-l2circuit-mode {
            aal5; # This dedicates FPC 1 PIC 2 to AAL5 mode.
         }
      }
   }
}
interfaces {
   at-0/1/1 {
      description "to CE2 at-1/0/1";
      encapsulation atm-ccc-cell-relay; # Cell-relay requires cell-relay encapsulation.
      atm-options {
         cell-bundle-size 4; # This sets the cell bundle size for the physical interface.
         pic-type atm2; # Layer 2 circuits are compatible with
         vpi 0; # ATM2 IQ interfaces.
      }
      unit 0 {
         encapsulation atm-ccc-cell-relay; # Also configure the encapsulation
         vci 32; # on the logical interface.
         cell-bundle-size 10; # The cell bundle size for the logical interface overrides
      }        # the physical interface setting.
   }
   at-1/2/1 {
      description "to CE2 at-1/0/2";
      atm-options {
         pic-type atm2; # Layer 2 circuits are compatible with
```

```
            vpi 0; # ATM2 IQ interfaces.
        }
        unit 0 {
            encapsulation atm-ccc-vc-mux; # AAL5 requires CCC VC MUX encapsulation.
            vci 32;
        }
    }
    so-1/1/0 {
        description "to P0 so-0/2/1";
        unit 0 {
            family inet {
                address 192.168.245.18/30;
            }
            family mpls; # Include the MPLS family on core-facing interfaces.
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.17.4/32;
            }
        }
    }
}
protocols {
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface so-1/1/0.0;
            interface lo0.0;
        }
    }
    ldp {    # LDP is required as the signaling protocol for Layer 2 circuits.
        interface so-1/1/0.0;
        interface lo0.0;
    }
    l2circuit {
        neighbor 10.255.17.2 { # This points to the loopback of the PE neighbor.
            interface at-0/1/1.32 { # Here you include the local CE-facing interface.
                virtual-circuit-id 1; # Be sure this ID matches the ID of your PE neighbor.
            }
        }
        neighbor 10.255.17.2 { # This points to the loopback of the PE neighbor.
            interface at-1/2/1.32 { # Here you include the local CE-facing interface.
                virtual-circuit-id 2; # Be sure this ID matches the ID of your PE neighbor.
            }
        }
    }
}
```

On Router CE2, configure the ATM2 IQ interfaces to handle ATM traffic. Interface at-1/0/1 handles standard ATM traffic while interface at-1/0/2 handles AAL5 traffic.

Router CE2

```
[edit]
interfaces {
    at-1/0/1 {
```

```
        description "to PE2 at-0/1/1";
        atm-options {
            pic-type atm2; # Layer 2 circuits are compatible with
            vpi 0;    # ATM2 IQ interfaces.
        }
        unit 0 {
            vci 32;
            family inet {
                address 10.1.1.2/30;
            }
        }
    }
    at-1/0/2 {
        description "to PE2 at-1/2/1";
        atm-options {
            pic-type atm2; # Layer 2 circuits are compatible with
            vpi 0;   # ATM2 IQ interfaces.
        }
        unit 0 {
            encapsulation atm-vc-mux;
            vci 32;
            family inet {
                address 10.2.2.2/30;
            }
        }
    }
}
```

Verifying Your Work

To verify proper operation of Layer 2 circuits, use the following commands:

- ping mpls l2circuit interface *interface-name*
- ping mpls l2circuit virtual-circuit *virtual-circuit-id* neighbor *ip-address*
- show l2circuit connections
- Options: [brief] | [down] | [extensive] | [history] | [instance] | [local-site] | [remote-site] |[status] | [summary] | [up] | [up-down]
- show interfaces
- show route table l2circuit.0
- show ldp database l2circuit detail

This is what the operational command output looks like for cell-relay mode on Router PE1:

```
user@PE1> show l2circuit connections
Layer-2 Circuit Connections:

Legend for connection status (St)
EI -- encapsulation invalid      NP -- interface not present
MM -- mtu mismatch               Dn -- down
```

```
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
OL -- no outgoing label          XX -- unknown
NC -- intf encaps not CCC/TCC

Legend for interface status
Up -- operational
Dn -- down

Neighbor: 10.255.17.4
    Interface                 Type  St     Time last up          # Up trans
    at-0/1/0.0 (vc 32)        rmt   Up     Jan 22 15:15:52 2003           1
      Local interface: at-0/1/0.0, Status: Up, Encapsulation: ATM CELL (VC Mode)

      Remote PE: 10.255.17.4, Negotiated control-word: Yes (Non-null)
      Incoming label: 100000, Outgoing label: 100000

user@PE1>  show route table l2circuit.0 detail

l2circuit.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
10.255.17.4:CtrlWord:9:32:Local/96 (1 entry, 1 announced)
         *L2CKT   Preference: 7
                Next hop: via so-0/2/0.0 weight 1, selected
                Label-switched-path PE1-PE2
                Protocol next hop: 10.255.17.4 Indirect next hop: 85135e8 367
                State: <Active Int>
                Local AS:    69
                Age: 2:34
                 Task: l2 circuit
                Announcement bits (1): 0-LDP
                AS path: I
                VC Label 100000, MTU 0,  cell-bundle size 80

10.255.17.4:CtrlWord:9:32:Remote/96 (1 entry, 1 announced)
         *LDP     Preference: 9
                Next hop type: Discard
                State: <Active Int>
                Local AS:    69
                Age: 28:11
                 Task: LDP
                Announcement bits (1): 1-l2 circuit
                AS path: I
                VC Label 100000, MTU 0,  cell-bundle size 80

user@PE1>  show interfaces at-0/1/0.0 extensive
  Logical interface at-0/1/0.0 (Index 66) (SNMP ifIndex 40) (Generation 4)
    Flags: Point-To-Point SNMP-Traps  Encapsulation: ATM-CCC-Cell-Relay
     L2 circuit cell bundle size: 10 , bundle timeout: 125 usec, timeout count:
0
    L2 circuit out-of-sequence count: 0
    Traffic statistics:
[...]

user@PE1>  show interfaces media at-0/1/0
Physical interface: at-0/1/0, Enabled, Physical link is Up
 Interface index: 154, SNMP ifIndex: 50
 Link-level type: ATM-CCC-Cell-Relay,  MTU: 4482, Clocking: Internal, SONET mode,

 L2 circuit mode: Cell, Speed: OC12, Loopback: None
[...]
```

```
user@PE1>  show ldp database l2circuit detail
Input label database, 10.255.17.2:0--10.255.17.4:0
    Label     Prefix
   100000     L2CKT CtrlWord ATM CELL (VC Mode) VC 32
              Cell bundle size: 80
              State: Active
              Age: 9:48

Output label database, 10.255.17.2:0--10.255.17.4:0
    Label     Prefix
   100000     L2CKT CtrlWord ATM CELL (VC Mode) VC 32
              Cell bundle size: 80
              State: Active
              Age: 9:48
```

This is what the operational command output looks like on Router PE1 if AAL5 mode is used:

```
user@PE1> show l2circuit connections
Layer-2 Circuit Connections:

Legend for connection status (St)
EI -- encapsulation invalid      NP -- interface not present
MM -- mtu mismatch               Dn -- down
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
OL -- no outgoing label          XX -- unknown
NC -- intf encaps not CCC/TCC

Legend for interface status
Up -- operational
Dn -- down

Neighbor: 10.255.17.4
    Interface                Type  St     Time last up          # Up trans
    at-1/2/0.0 (vc 32)       rmt   Up     Feb 18 18:00:00 2003           1
      Local interface: at-1/2/0.0, Status: Up, Encapsulation: ATM AAL5
      Remote PE: 10.255.17.4, Negotiated control-word: Yes (Non-null)
      Incoming label: 100016, Outgoing label: 100032

user@PE1> show interfaces media at-0/1/0
Physical interface: at-0/1/0, Enabled, Physical link is Up
 Interface index: 154, SNMP ifIndex: 50
 Link-level type: ATM-PVC, MTU: 4482, Clocking: Internal, SONET mode,
 L2 circuit mode: AAL5, Speed: OC12, Loopback: None, Payload scrambler: Enabled
[...]

user@PE1>  show interfaces at-1/2/0.0 extensive
  Logical interface at-1/2/0.0 (Index 68) (SNMP ifIndex 40) (Generation 38)
    Flags: Point-To-Point SNMP-Traps  Encapsulation: ATM-CCC-VCMUX
    L2 circuit out-of-sequence count: 0
    Traffic statistics:[...]
```

Example: Layer 2 Circuit Traffic Engineering over Multiple LSPs Configuration

Figure 58: Layer 2 Circuit Traffic Engineering Topology Diagram

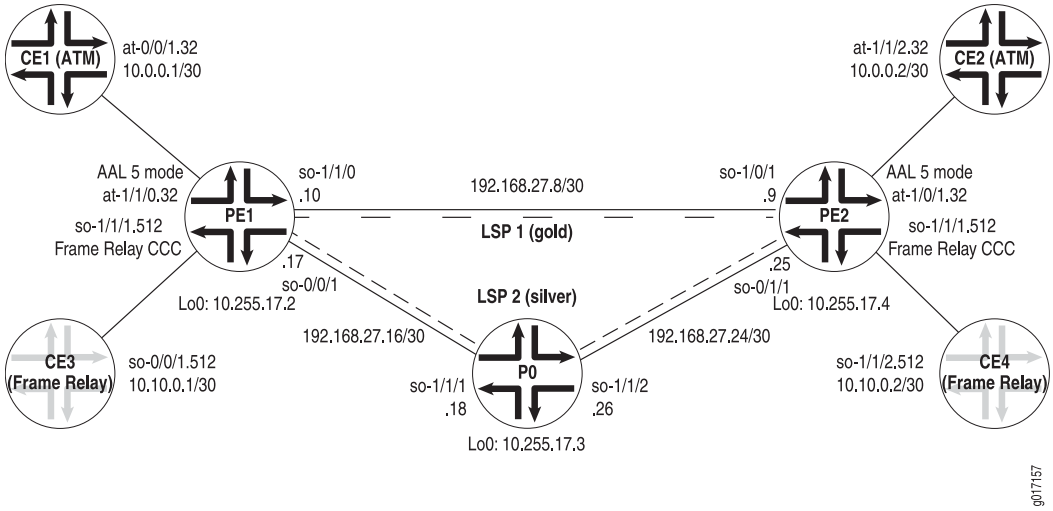


Figure 58 on page 552 shows a network topology designed to traffic engineer different Layer 2 circuits over select LSPs. Across provider edge routers PE1 and PE2, an ATM AAL5 mode Layer 2 circuit connects customer edge routers CE1 and CE2, and a Frame Relay Layer 2 circuit connects routers CE3 and CE4. To maintain traffic separation, the ATM traffic is mapped onto LSP1 with a community named gold, and the Frame Relay traffic is mapped onto LSP2 with a community named silver. LSP1 takes the direct route between routers PE1 and PE2, while LSP2 travels from Router PE1 to PE2 through Router P0.

In addition to traffic engineering, you can send Layer 2 control information in the control word of a Layer 2 circuit. In this case, Frame Relay discard eligible (DE), forward explicit congestion notification (FECN), and backward explicit congestion notification (BECN) information is mapped into the control word. Likewise, ATM cell loss priority (CLP) and explicit forward congestion indicator (EFCI) information is mapped into the control word.

To traffic engineer Layer 2 circuits over multiple LSPs, you assign a set of Layer 2 circuits to a community and then apply a policy to send the community traffic over a desired LSP. To create communities, include the community *community-name* statement at the [edit policy-options] hierarchy level. To assign a Layer 2 circuit to a community, include the community *community-name* statement at the [edit protocols l2circuit neighbor *neighbor-id* interface *interface-name*] hierarchy level. To send community traffic over a specific LSP, include the community *community-name* statement at the [edit policy-options policy-statement *policy-name* term *term-name* from] hierarchy level and the install-next-hop lsp *lsp-name* statement at the [edit policy-options policy-statement *policy-name* term *term-name* then] hierarchy level.

On Router CE1, configure the ATM2 IQ interface at-0/0/1.32 to handle ATM AAL5 traffic:

Router CE1

```
[edit]
interfaces {
   at-0/0/1 {
      description "to PE1 at-1/1/0";
      atm-options {
         pic-type atm2; # Layer 2 circuits are compatible with
         vpi 0; # ATM2 IQ interfaces.
      }
      unit 0 {
         encapsulation atm-vc-mux; # Use ATM VC MUX encapsulation on the CE.
         point-to-point;
         vci 0.32;
         family inet {
            address 10.0.0.1/30;
         }
      }
   }
}
```

On Router CE3, configure the SONET/SDH interface at so-0/0/1 to handle Frame Relay traffic:

Router CE3

```
[edit]
interfaces
so-0/0/1 {
   description “to PE1 so-1/1/1”
   encapsulation frame-relay; # Use Frame Relay encapsulation on the CE router.
   unit 0 {
      encapsulation frame-relay;
      point-to-point;
      dlci 512;
      family inet {
         address 10.10.0.1/30;
      }
   }
}
```

On Router PE1, configure the ATM2 IQ-based CE1-facing interface at-1/1/0 with ATM VC multiplexing CCC encapsulation on the logical interface. Also enable the corresponding Layer 2 circuit modes at the [edit chassis] hierarchy level. In this case, you must configure AAL5 mode on PIC 1 in FPC 1. Once you configure the ATM2 IQ-based Layer 2 circuit, the CLP and EFCI bits are mapped to the control word by default.

Next, configure the Frame Relay interface so-1/1/1 with Frame Relay CCC encapsulation on both the physical and logical interface. Map the DE, FECN, and BECN bits to the control word with the translate-fecn-and-becn and translate-discard-eligible statements at the [edit interfaces so-*fpc/pic/port* unit *unit-number* family ccc] hierarchy level.

Establish your Layer 2 circuits with configuration of the l2circuit statement at the [edit protocols] hierarchy level. Remember to include in your Layer 2 circuit configuration the IP address of your remote PE neighbor (usually the loopback address of the neighbor), the interfaces connected to the CE router, and a virtual circuit identifier for each VC. In this case, you will establish one VC for ATM AAL5 traffic

and a second VC for Frame Relay traffic. Then, configure MPLS, LDP, and an IGP (such as OSPF) to enable signaling for your Layer 2 circuit. Two LSPs are established for the ATM and Frame Relay traffic: LSP1 for ATM traffic going directly to Router PE2 and LSP 2 for Frame Relay traffic going through Router P0 before going on to Router PE2.

Finally, configure a community for traffic separation for the ATM and Frame Relay Layer 2 circuits. Assign community gold to the ATM VC and community silver to the Frame Relay VC. Remember to give the communities numerical values and configure a routing policy to match the communities to specific LSPs. This policy is applied as an export policy for the forwarding table at the [edit routing-options] hierarchy level.

Router PE1

```
[edit]
chassis {
    fpc 1 {
        pic 1 {
            atm-l2circuit-mode {
                aal5; # This dedicates FPC 1 PIC 1 to AAL5 mode.
            }
        }
    }
}
interfaces {
    at-1/1/0 {
        description "to CE1 at-0/0/2";
        atm-options {
            pic-type atm2; # Layer 2 circuits are compatible with
            vpi 0; # ATM2 IQ interfaces.
        }
        unit 0 {
            encapsulation atm-ccc-vc-mux; # CLP/EFCI bits are mapped to control word.
            vci 0.32;
        }
    }
    so-0/1/0 {
        description "to P0 so-0/0/0";
        unit 0 {
            family inet {
                address 192.168.27.17/30;
            }
            family mpls; # Include the MPLS family on core-facing interfaces.
        }
    }
    so-1/1/0 {
        description "to PE2 so-1/0/1";
        unit 0 {
            family inet {
                address 192.168.27.10/30;
            }
            family mpls; # Include the MPLS family on core-facing interfaces.
        }
    }
    so-1/1/1 {
        description "to CE3 so-0/0/1";
        dce;
```

```
        encapsulation frame-relay-ccc;
        unit 0 {
            encapsulation frame-relay-ccc;
            point-to-point;
            dlci 512;
            family ccc {
                translate-fecn-and-becn; # Option to map FECN/BECN bits to control word.
                translate-discard-eligible; # Option to map DE bit to control word.
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.17.2/32;
            }
        }
    }
}
routing-options {
    forwarding-table {
        export layer2communities; # This applies communities to the Layer 2 circuits.
    }
}
protocols {
    mpls {
        label-switched-path lsp1 { # ATM LSP 1 goes directly to PE2.
            to 10.255.17.4;
            primary direct;
        }
        label-switched-path lsp2 { # Frame Relay LSP 2 goes through P0.
            to 10.255.17.4;
            primary thruP0;
        }
        path direct {
            192.168.27.9 strict;
        }
        path thruP0 {
            192.168.27.18 strict;
            192.168.27.25 strict;
        }
        interface so-0/0/1.0;
        interface so-1/1/0.0;
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface so-0/0/1.0;
            interface so-1/1/0.0;
            interface lo0.0;
        }
    }
    ldp { # LDP is required as the signaling protocol for Layer 2 circuits.
        interface so-0/0/1.0;
        interface so-1/1/0.0;
        interface lo0.0;
```

```
    }
    l2circuit {
      neighbor 10.255.17.4 {# This points to the loopback of the PE neighbor.
        interface at-1/1/0.32 { # Here you include the local CE-facing interface.
          virtual-circuit-id 1; # Be sure this ID matches the ID of your PE neighbor.
          community gold;# Assigns the ATM Layer 2 circuit to the gold community.
        }
      }
      neighbor 10.255.17.4 {# This points to the loopback of the PE neighbor.
        interface so-1/1/1.512 { # Here you include the local CE-facing interface.
          virtual-circuit-id 2; # Be sure this ID matches the ID of your PE neighbor.
          community silver; # Assigns the Frame Relay Layer 2 circuit to silver.
        }
      }
    }
  }
  policy-options {
    policy-statement layer2communities { # Here you map the communities to LSPs.
      term 10 {
        from community gold;  # Apply community gold to LSP 1.
        then {
          install-nexthop lsp lsp1;
          accept;
        }
        term 20 {
          from community silver; # Apply community silver to LSP 2.
          then {
            install-nexthop lsp lsp2;
            accept;
          }
          community gold members 103:1; # Assign numerical value to community
            gold.
          community silver members 103:2; # Assign numerical value to community
            silver.
        }
      }
    }
  }
```

On Router P0, configure LDP, MPLS, and OSPF on the interfaces connected to the PE routers. The core router provides the MPLS backbone needed to tunnel Layer 2 traffic from the ingress PR router to the egress PE router. Only LSP 2 for Frame Relay passes through Router P0.

Router P0

```
[edit]
interfaces {
  so-1/1/1 {
    description "to PE1 so-0/0/1";
    unit 0 {
      family inet {
        address 192.168.27.18/30;
      }
      family mpls; # Include the MPLS family on core interfaces.
    }
  }
  so-1/1/2 {
```

```
        description "to PE2 so-0/1/1";
        unit 0 {
          family inet {
            address 192.168.27.26/30;
          }
          family mpls; # Include the MPLS family on core interfaces.
        }
      }
      lo0 {
        unit 0 {
          family inet {
            address 10.255.17.3/32;
          }
        }
      }
    }
    protocols {
      mpls {
        interface so-1/1/1.0;
        interface so-1/1/2.0;
      }
      ospf {
        traffic-engineering;
        area 0.0.0.0 {
          interface so-1/1/1.0;
          interface so-1/1/2.0;
        }
      }
      ldp {# LDP is required as the signaling protocol for Layer 2 circuits.
        interface so-1/1/1.0;
        interface so-1/1/2.0;
      }
    }
```

On Router PE2, complete the Layer 2 circuit by configuring statements to match those previously set on Router PE1.

Establish your Layer 2 circuits with configuration of the l2circuit statement at the [edit protocols] hierarchy level. Remember to include in your Layer 2 circuit configuration the IP address of your remote PE neighbor (usually the loopback address of the neighbor), the interfaces connected to the CE router, and a virtual circuit identifier for each VC. In this case, you will establish one VC for ATM AAL5 traffic and a second VC for Frame Relay traffic. Then, configure MPLS, LDP, and an IGP (such as OSPF) to enable signaling for your Layer 2 circuit. Two LSPs are established for the ATM and Frame Relay traffic: LSP1 for ATM traffic going directly to Router PE2 and LSP 2 for Frame Relay traffic going through Router P0 before going on to Router PE2.

Finally, configure a community for traffic separation for the ATM and Frame Relay Layer 2 circuits. The ATM VC has community gold and the Frame Relay VC has community silver. Remember to give the communities numerical values and configure a routing policy to match the communities to specific LSPs. This policy is applied as an export policy for the forwarding table at the [edit routing-options] hierarchy level.

Router PE2 [edit]

```
chassis {
   fpc 1 {
      pic 0 {
         atm-l2circuit-mode {
            aal5; # This dedicates FPC 1 PIC 0 to AAL5 mode.
         }
      }
   }
}
interfaces {
   at-1/0/1 {
      description "to CE2 at-1/1/2";
      atm-options {
         pic-type atm2; # Layer 2 circuits are compatible with
         vpi 0; # ATM2 IQ interfaces.
      }
      unit 0 {
         encapsulation atm-ccc-vc-mux; # CLP and EFCI appear in the control word.
         vci 0.32;
      }
   }
   so-0/1/1 {
      description "to P0 so-1/1/2";
      unit 0 {
         family inet {
            address 192.168.27.25/30;
         }
         family mpls; # Include the MPLS family on core-facing interfaces.
      }
   }
   so-1/0/1 {
      description "to PE1 so-1/1/0";
      unit 0 {
         family inet {
            address 192.168.27.9/30;
         }
         family mpls; # Include the MPLS family on core-facing interfaces.
      }
   }
   so-1/1/1 {
      description "to CE4 so-1/1/2";
      dce;
      encapsulation frame-relay-ccc;
      unit 0 {
         encapsulation frame-relay-ccc;
         point-to-point;
         dlci 512;
         family ccc {
            translate-fecn-and-becn; # Option to map FECN/BECN bits to control word.
            translate-discard-eligible; # Option to map DE bit to control word.
         }
      }
   }
   lo0 {
      unit 0 {
         family inet {
```

```
            address 10.255.17.4/32;
          }
        }
      }
    }
    routing-options {
      forwarding-table {
        export layer2communities; # This maps communities to the Layer 2 circuits.
      }
    }
    protocols {
      mpls {
        label-switched-path lsp1 { # ATM LSP 1 goes directly to Router PE2.
          to 10.255.17.2;
          primary direct;
        }
        label-switched-path lsp2 { # Frame Relay LSP 2 goes through Router P0.
          to 10.255.17.2;
          primary thruP0;
        }
        path direct {
          192.168.27.10 strict;
        }
        path thruP0 {
          192.168.27.26 strict;
          192.168.27.17 strict;
        }
        interface so-0/1/1.0;
        interface so-1/0/1.0;
      }
      ospf {
        traffic-engineering;
        area 0.0.0.0 {
          interface so-0/1/1.0;
          interface so-1/0/1.0;
          interface lo0.0;
        }
      }
      ldp { # LDP is required as the signaling protocol for Layer 2 circuits.
        interface so-0/1/1.0;
        interface so-1/0/1.0;
        interface lo0.0;
      }
      l2circuit {
        neighbor 10.255.17.2 {# This points to the loopback of the PE neighbor.
          interface at-1/0/1.32 { # Here you include the local CE-facing interface.
            virtual-circuit-id 1; # Be sure this ID matches the ID of your PE neighbor.
            community gold;# Assigns the ATM Layer 2 circuit to the gold community.
          }
        }
        neighbor 10.255.17.2 {# This points to the loopback of the PE neighbor.
          interface so-1/1/1.512 { # Here you include the local CE-facing interface.
            virtual-circuit-id 2; # Be sure this ID matches the ID of your PE neighbor.
            community silver; # Assigns the Frame Relay Layer 2 circuit to silver.
          }
        }
```

```
    }
  }
  policy-options {
    policy-statement layer2communities { # Here you map communities to LSPs.
      term 10 {
        from community gold; # Apply community gold to LSP 1.
        then {
          install-nexthop lsp lsp1;
          accept;
        }
        term 20 {
          from community silver; # Apply community silver to LSP 2.
          then {
            install-nexthop lsp lsp2;
            accept;
          }
          community gold members 103:1; # Assign numerical value to community
            gold.
          community silver members 103:2; # Assign numerical value to community
            silver.
        }
      }
    }
  }
```

On Router CE2, configure the ATM2 IQ interfaces to handle ATM traffic. Interface at-1/0/1 handles AAL5 traffic.

Router CE2

```
[edit]
interfaces {
  at-1/1/2 {
    description "to PE2 at-1/0/1";
    atm-options {
      pic-type atm2; # Layer 2 circuits are compatible with
      vpi 0; # ATM2 IQ interfaces.
    }
    unit 0 {
      encapsulation atm-vc-mux; # Use ATM VC MUX encapsulation on the CE.
      point-to-point;
      vci 0.32;
      family inet {
        address 10.0.0.2/30;
      }
    }
  }
}
```

On Router CE4, configure the SONET/SDH interface at so-1/1/2 to handle Frame Relay traffic:

Router CE3

```
[edit]
interfaces {
  so-1/1/2 {
    description " to PE2 so-1/1/1";
    encapsulation frame-relay-ccc; # Use Frame Relay encapsulation on the CE.
```

```
        unit 0 {
          encapsulation frame-relay-ccc;
          point-to-point;
          dlci 512;
          family inet {
            address 10.10.0.2/30;
          }
        }
      }
    }
```

Verifying Your Work

To verify proper operation of traffic engineered Layer 2 circuits, use the following command:

```
show route table mpls.0 detail
```

On Router PE1, you can see that ATM traffic is part of the gold community that has a value of 103:1 and is associated with LSP 1. Likewise, Frame Relay traffic is part of the silver community that has a value of 103:2 and is associated with LSP 2:

```
user@PE1> show route table mpls.0 detail
mpls.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
at-1/1/0.32  (1 entry, 1 announced)
          *L2CKT  Preference: 7
                  Next hop: 192.168.27.9 via so-1/1/0.0 weight 1, selected
                 Label-switched-path lsp1
                 Label operation: Push 100032 Offset: -4
                 Next hop: via so-0/0/1.0 weight 1
                 Label-switched-path lsp2
                 Label operation: Push 100032 Offset: -4
                 Protocol next hop: 10.255.17.4
                 Push 100032 Offset: -4
                  Indirect next hop: 8576bd0 300
                 State: <Active Int>
                 Age: 7:18
                 Task: Common L2 VC
                 Announcement bits (2): 0-KRT 1-Common L2 VC
                 AS path: I
                 Communities: 103:1 # This is the  gold  community.

so-1/1/1.512 (1 entry, 1 announced)
          *L2CKT  Preference: 7
                  Next hop: 192.168.27.9 via so-1/1/0.0 weight 1
                  Label-switched-path lsp1
                  Label operation: Push 100048 Offset: -4
                  Next hop: via so-0/0/1.0 weight 1, selected
                  Label-switched-path lsp2
                  Label operation: Push 100048 Offset: -4
                  Protocol next hop: 10.255.17.4
                  Push 100048 Offset: -4
                   Indirect next hop: 860f1f8 293
                  State: <Active Int>
                  Age: 5:15
                  Task: Common L2 VC
                  Announcement bits (2): 0-KRT 1-Common L2 VC
```

```
AS path: I
Communities: 103:2 # This is the  silver  community.
```

Example: APS for a Layer 2 Circuit Configuration

Figure 59: APS for a Layer 2 Circuit Topology Diagram

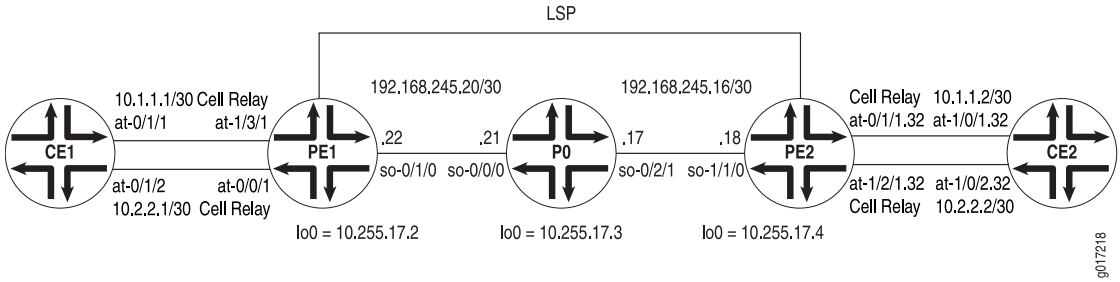


Figure 59 on page 562 shows that APS is configured on a PE router to protect a PE-CE link in a Layer 2 circuit. This example shows only the PE router configuration and assumes that you have preconfigured a full Layer 2 circuit topology. For more information about configuring Layer 2 circuits, see "Configuring Layer 2 Circuits" on page 518.

On Router PE1, configure ATM2 IQ interface at-0/0/1 as an APS protect circuit and ATM2 IQ interface at-1/3/1 as a working circuit. Also, configure the working circuit interface as the primary interface for your Layer 2 circuit and configure the protect circuit interface as the protected interface for your Layer 2 circuit.

Router PE1

```
[edit]
chassis {
    fpc 0 {
        pic 0 {
            atm-l2circuit-mode {
                cell; # This dedicates FPC 0 PIC 0 to cell-relay mode.
            }
        }
    }
    fpc 1 {
        pic 3 {
            atm-l2circuit-mode {
                cell; # This dedicates FPC 1 PIC 3 to cell-relay mode.
            }
        }
    }
}
interfaces {
    at-0/0/1 {
        description "To CE1 at-0/1/2";
        encapsulation atm-ccc-cell-relay;
        sonet-options {
            aps {
                protect-circuit TEST; # This interface is the APS protect circuit.
            }
```

```
        }
        atm-options {
            pic-type atm2;
            promiscuous-mode;
        }
        unit 0 {
            allow-any-vci;
        }
    }
    at-1/3/1 {
        description "To CE1 at-0/1/1";
        encapsulation atm-ccc-cell-relay;
        sonet-options {
            aps {
                working-circuit TEST; # This interface is the APS working circuit.
            }
        }
        atm-options {
            pic-type atm2;
            promiscuous-mode;
        }
        unit 0 {
            allow-any-vci;
        }
    }
}
protocols {
    l2circuit {
        neighbor 10.255.17.4 {
            interface at-1/3/1.0 { # The Layer 2 circuit interface is the working circuit.
                protect-interface at-0/0/1.0; # The protect-interface is the protect circuit.
                virtual-circuit-id 100;
            }
        }
    }
}
```

Verifying Your Work

To verify proper operation of APS for Layer 2 circuits, use the following command:

```
show l2circuit connections
```

After you configure the Layer 2 circuit and the APS working and protect circuits, you can see which APS circuit is active for the Layer 2 circuit with the show l2circuit connections command. The first local interface that is displayed is always the active circuit. If the second local interface field indicates Protect-Inactive, the working circuit is active, as shown in this output sample.

```
user@PE0> show l2circuit connections
Layer-2 Circuit Connections:

Legend for connection status (St)
EI -- encapsulation invalid      NP -- interface h/w not present
```

```
MM -- mtu mismatch               Dn -- down
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
OL -- no outgoing label          XX -- unknown
NC -- intf encaps not CCC/TCC
CB -- rcvd cell-bundle size bad

Legend for interface status
Up -- operational
Dn -- down

Neighbor: 10.255.17.4
Interface                 Type  St     Time last up          # Up trans
 at-1/3/1.0(vc 100)        rmt   Up     Sep  3 17:48:25 2003           1
   Local interface: at-1/3/1.0, Status: Up, Encapsulation: ATM CELL (PORT Mode)
   Remote PE: 10.255.17.4, Negotiated control-word: Yes (Null)
   Incoming label: 100368, Outgoing label: 100112
   Local interface: at-0/0/1.0, Status: Dn, Encapsulation: ATM CELL (PORT Mode),

   Protect-Inactive
```

Conversely, if the first local interface is marked with the Protect-Active indicator, and the second local interface indicates Primary-Inactive, the protect circuit is active, as shown here:

```
user@PE0> show l2circuit connections
Layer-2 Circuit Connections:

Legend for connection status (St)
EI -- encapsulation invalid      NP -- interface h/w not present
MM -- mtu mismatch               Dn -- down
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
OL -- no outgoing label          XX -- unknown
NC -- intf encaps not CCC/TCC
CB -- rcvd cell-bundle size bad

Legend for interface status
Up -- operational
Dn -- down

Neighbor: 10.255.17.4
 Interface                 Type  St     Time last up          # Up trans
 at-1/3/1.0(vc 100)        rmt   Up     Sep  3 17:51:06 2003           2
   Local interface: at-0/0/1.0, Status: Up,  Encapsulation: ATM CELL (PORT Mode),

    Protect-Active
   Remote PE: 10.255.17.4, Negotiated control-word: No
   Incoming label: 100368, Outgoing label: 100112
   Local interface: at-1/3/1.0, Status: Dn,  Encapsulation: ATM CELL (PORT Mode),

    Primary-Inactive
```

For More Information

For additional information about Layer 2 circuits, see the following:

- *JUNOS Network Interfaces Configuration Guide*
- *JUNOS Class of Service Configuration Guide*
- *JUNOS VPNs Configuration Guide*
- RFC 3036, *LDP Specification*
- RFC 4447, *Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)* (except section 5.3)
- RFC 4448, *Encapsulation Methods for Transport of Ethernet over MPLS Networks*
- Internet draft draft-martini-atm-encap-mpls-01.txt, *Encapsulation Methods for Transport of ATM Cells/Frame Over IP and MPLS Networks* (expires December 2002)
- Internet draft draft-martini-l2circuit-encap-mpls-11.txt, *Encapsulation Methods for Transport of Layer 2 Frames over MPLS Networks* (expires August 2006)
- Internet draft draft-martini-l2circuit-trans-mpls-19.txt, *Transport of Layer 2 Frames Over MPLS* (expires September 2006)
- Internet draft draft-kompella-ppvpn-l2vpn-03.txt, *Layer 2 VPNs Over Tunnels* (expires October 2003)

Revision History

13 April 2009—9.5R1 Release. Roy Spencer.

15 January 2009—9.4R1 Release. Roy Spencer.

10 October 2008—9.3R1 Release. Roy Spencer.

10 July 2008—9.2R1 Release. Roy Spencer.

10 April 2008—9.1R1 Release. Roy Spencer.

1 February 2008—9.0R1 Release. Fawn Damitio.

29 June 2007—8.4R1 Release. Fawn Damitio.

27 March 2007—8.3R1 Release. Fawn Damitio.

12 January 2007—Added support for M120 router and MX960 routers. 8.2R1 Release. Fawn Damitio.

15 September 2006—Updated the references to the Layer 2 Circuits IETF drafts and RFCs, 8.1R1 Release. Richard Hendricks.

29 June 2006—8.0R1 Release. Richard Hendricks.

27 March 2006—7.6R1 Release. Richard Hendricks.

9 January 2006—7.5R1 Release. Richard Hendricks.

14 September 2005—7.4R1 Release. Richard Hendricks.

13 June 2005—Added support for configuring Layer 2 circuits simultaneously over RSVP and LDP LSPs, 7.3R1 Release. Richard Hendricks.

5 April 2005—7.2R1 Release. Richard Hendricks.

2 February 2005—Added local interface switching for Layer 2 circuits, 7.1R1 Release. Richard Hendricks.

6 October 2004—Added support for specifying a unique MTU for each Layer 2 circuit, 7.0R1 Release. Richard Hendricks.

6 July 2004—Added ATM2 IQ interface-based CoS and additional trunk-related statements, 6.4R1 Release. Richard Hendricks.

5 April 2004—Added ping mpls l2circuit commands, 6.3R1 Release. Richard Hendricks.

22 December 2003—Added Layer 2 circuit trunk mode and bandwidth reservation for Layer 2 circuits, 6.2R1 Release. Richard Hendricks.

22 September 2003—Added APS for Layer 2 circuits information and updated the traffic engineering example, 6.1R1 Release. Richard Hendricks.

30 June 2003—Added traffic engineering and control word mapping for Frame Relay and ATM2, 6.0R1 Release. Walter Goralski.

2 April 2003—Added PPP, HDLC, ATM Cell Mode, and AAL5 information, 5.7R1 Release. Richard Hendricks.

10 July 2002—Reformatted document. Richard Hendricks.

21 November 2001—Initial document written. Bill Nowak.

Chapter 13

Multicast over Layer 3 VPNs

This feature guide covers these topics:

Multicast over Layer 3 VPNs Overview

The JUNOS software provides three ways to configure IP version 4 (IPv4) multicast over Layer 3 virtual private networks (VPNs):

- Multiprotocol BGP-based multicast VPNs: next-generation, defined by a set of sender sites and a set of receiver sites and use BGP as the signaling protocol. We recommend using this method to configure multicast on Layer 3 VPNs because it is a simpler implementation than draft-rosen multicast VPNs.
- Draft-rosen multicast VPNs with service provider tunnels operating in any-source multicast (ASM) mode—Described in RFC 4364, *BGP/MPLS IP Virtual Private Networks (VPNs)* and based on Section Two of the IETF Internet draft draft-rosen-vpn-mcast-06.txt, *Multicast in MPLS/BGP VPNs*. This information is provided to you in case you already have dual PIM multicast VPNs configured on your network.
- Draft-rosen multicast VPNs with service provider tunnels operating in source-specific multicast (SSM) mode—Described in RFC 4364, *BGP/MPLS IP Virtual Private Networks (VPNs)* and based on the IETF Internet draft draft-rosen-vpn-mcast-07.txt, *Multicast in MPLS/BGP IP VPNs*. For more information about these types of draft-rosen multicast VPNs, see the *JUNOS Multicast Protocols Configuration Guide*.

This document assumes the reader is familiar with Layer 3 VPN operation on Juniper Networks routers, as well as standard PIM configurations. For more information on Protocol Independent Multicast (PIM) and Multicast Source Discovery Protocol (MSDP) and their usage in a Layer 3 VPN, see the *JUNOS Multicast Protocols Configuration Guide*. For more information on Layer 3 VPN configuration, see the *JUNOS VPNs Configuration Guide*. Both manuals are located at http://www.juniper.net/techpubs/software/index.html.

Multiprotocol BGP-Based Multicast VPNs: Next-Generation

Multiprotocol BGP-based multicast VPNs (also referred to as next-generation Layer 3 VPN multicast) constitute the next evolution after dual multicast VPNs (Draft-Rosen) and provide a simpler solution for administrators who want to configure multicast over Layer 3 VPNs.

The main characteristics of multiprotocol BGP-based multicast VPNs are:

- They extend Layer 3 VPN service (RFC 2547) to support IP multicast for Layer 3 VPN service providers
- They follow the same architecture as specified by RFC 2547 for unicast VPNs. Specifically, BGP is used as the control plane.
- They eliminate the requirement for the virtual router (VR) model, which is specified in Internet draft draft-rosen-vpn-mcast, *Multicast in MPLS/BGP VPNs*, for multicast VPNs.
- They rely on RFC-based unicast with extensions for intra-AS and inter-AS communication.

Multiprotocol BGP-based VPNs are defined by two sets of sites: a sender set and a receiver set. Hosts within a receiver site set can receive multicast traffic and hosts within a sender site set can send multicast traffic. A site set can be both receiver and sender, which means that hosts within such a site can both send and receive multicast traffic. Multiprotocol BGP-based VPNS can span organizations (so the sites can be intranets or extranets), can span service providers, and can overlap.

Site administrators configure multiprotocol BGP-based VPNs based on customer requirements and the existing BGP and MPLS VPN infrastructure. For more detailed information about multiprotocol BGP-based VPN configuration statements, see the *JUNOS VPNs Configuration Guide*.

Dual PIM Multicast VPNs: Draft Rosen

JUNOS software supports Layer 3 VPNs based on the Internet draft draft-rosen-rfc2547bis, *BGP/MPLS VPNs*. This Internet draft defines a mechanism by which service providers can use their IP backbones to provide Layer 3 VPN services to their customers. The sites that make up a Layer 3 VPN are connected over a provider's existing public Internet backbone.

VPNs based on draft-rosen-rfc2547bis are also known as BGP/MPLS VPNs because BGP is used to distribute VPN routing information across the provider's backbone, and MPLS is used to forward VPN traffic across the backbone to remote VPN sites.

Customer networks, because they are private, can use either public addresses or private addresses, as defined in RFC 1918, *Address Allocation for Private Internets*. When customer networks that use private addresses connect to the public Internet infrastructure, the private addresses might overlap with the private addresses used by other network users. BGP/MPLS VPNs solve this problem by prefixing a VPN identifier to each address from a particular VPN site, thereby creating an address that is unique both within the VPN and within the public Internet. In addition, each VPN has its own VPN-specific routing table that contains the routing information for that VPN only.

In a unicast environment for Layer 3 VPNs, all VPN states are contained within the provider edge (PE) routers. With multicast over Layer 3 VPNs, two PIM adjacencies are established: one between the customer edge (CE) and PE routers through a VPN routing and forwarding (VRF) routing instance, the second between the main PE routers and their service provider core neighbors.

The set of master PIM adjacencies throughout the service provider's network makes up the forwarding path, and eventually forms a rendezvous point (RP) multicast distribution tree. The tree is rooted at the RP contained within the service provider's network. Because of this, core provider transit routers within the service provider's network must maintain multicast state information for the VPNs.

For multicast in Layer 3 VPNs to work correctly, there must be two types of rendezvous points. The VPN customer rendezvous point (VPN C-RP) is an RP that resides within a VPN that connects the segments of a customer network. The service provider rendezvous point (SP-RP) resides within the service provider network itself. Because a PE router connects to both the customer network and the service provider network, a PE router can act as an SP-RP, a VPN C-RP, or both.

NOTE: If you configure auto-RP or bootstrap router (BSR) on a PE router, the PE router cannot act as a VPN C-RP in a routing instance, but can learn about another router acting as the VPN C-RP.

System Requirements for Multiprotocol BGP-Based Multicast VPNs: Next-Generation

To implement multiprotocol BGP-based multicast VPNs, your system must meet these minimum requirements:

- JUNOS Release 9.2 or later for PIM dense mode, bootstrap router (BSR), auto-RP, and configuration of a PE router as the VPN C-RP.
- JUNOS Release 8.5 or later for point-to-multipoint TE provider tunnels (next-generation multicast VPN only).
- JUNOS Release 8.4 or later.
- Any hardware needed in your network to enable your Juniper Networks routers to act as PE routers.
- On M-series and T-series routers, a Tunnel Services PIC or for any provider core router acting as an SP-RP.
- On M-series and T-series routers, a Tunnel Services PIC for any PE router where GRE tunneling is needed.
- On M-series and T-series routers, a Tunnel Services PIC for any CE or PE router acting as a DR or VPN C-RP.
- On M-series and T-series routers, a Tunnel Services PIC is required for GRE tunneling, as specified in Section Two of the IETF Internet draft *Multicast in MPLS/BGP VPNs*.

System Requirements for Dual PIM Multicast VPNs: Draft Rosen

- JUNOS Release 8.2 or later for support on MX-series routing platforms.
- JUNOS Release 7.2 or later for MSDP in a Layer 3 VPN.
- JUNOS Release 7.1 or later for multicast distribution trees for data.

- JUNOS Release 6.4 or later for PIM sparse mode graceful restart and configuring a PE router as the VPN C-RP.
- JUNOS Release 5.5 or later for PIM dense mode and logical loopback interfaces.
- JUNOS Release 5.3 or later for PIM sparse mode.
- Any hardware needed in your network to enable your Juniper Networks routers to act as PE routers.
- On M-series and T-series routers, a Tunnel Services PIC or for any provider core router acting as an SP-RP.
- On M-series and T-series routers, a Tunnel Services PIC for any PE router where GRE tunneling is needed.
- On M-series and T-series routers, a Tunnel Services PIC for any CE or PE router acting as a DR or VPN C-RP.
- On M-series and T-series routers, a Tunnel Services PIC is required for GRE tunneling, as specified in Section Two of the IETF Internet draft *Multicast in MPLS/BGP VPNs*.
- For point-to-multipoint TE, a Tunnel Services PIC or vrf-tabel-label configuration.

Terms and Acronyms

I

inclusive tree In multiprotocol BGP multicast VPNs, a single multicast distribution tree in the backbone that carries all the multicast traffic from a specified set of one or more multicast VPNs. An inclusive tree that carries the traffic of more than one multicast VPNs is an *aggregate inclusive tree*. An inclusive tree contains, as its members, all the PEs that attach to receiver sites of any of the multicast VPNs using the tree.

M

master PIM instance The global instance of PIM that is configured at the [edit protocols pim] hierarchy level.

multicast domain The set of VPN routing and forwarding (VRF) instances associated with interfaces that can send multicast traffic to one another.

S

selective tree In multiprotocol BGP multicast VPNs, a single multicast distribution tree in the backbone that carries traffic that belongs to a specified set of one or more multicast groups from one or more multicast VPNs. Such a tree is referred to as an *aggregate selective tree* when the multicast groups belong to different multicast VPNs.

SP-RP The rendezvous point (RP) for the service provider (this RP is not contained within the VPN).

V

VPN C-RP The customer RP for the VPN (this RP is contained within the VPN).

Configuring Multiprotocol BGP-Based Multicast VPNs: Next-Generation

To implement multiprotocol BGP-based multicast VPNs, you must perform one or more of the following procedures:

- Creating a Unique Logical Loopback Interface for the Routing Instance on page 572
- Configuring Interfaces for Layer 3 VPNs on page 572
- Configuring BGP, MPLS, RSVP, and an IGP on the PE and Core Routers on page 573
- Creating a Routing Instance for Multiprotocol BGP-Based Multicast VPN on page 573
- Option: Configuring Sender and Receiver Sites on page 574
- Option: Specifying Route Targets on page 574
- Configuring Provider Tunnels on page 576
- Enabling Multicast VPN in BGP on page 577
- Configuring Intra-AS Inclusive Point-to-Multipoint TE LSPs on page 577
- Configuring Intra-AS Selective Provider Tunnels on page 579
- Configuring the Master PIM Instance on the PE Router for BGP-based Multicast VPNs on page 581
- Configuring the Router's IPv4 Bootstrap Router Priority on page 582

Creating a Unique Logical Loopback Interface for the Routing Instance

To facilitate the PIM protocol within a Layer 3 VPN, configure a unique loopback interface for the routing instance at the [edit interfaces lo0 unit] hierarchy level:

```
[edit interfaces]
lo0 {
    unit 1 {
        family inet {
            address  ip-address;
        }
    }
}
```

Configuring Interfaces for Layer 3 VPNs

Configure the Layer 3 VPN logical interfaces and specify the family as inet:

```
[edit interfaces]
```

```
interface-name {
   unit logical-unit-number {
      family inet;
   }
}
```

Configuring BGP, MPLS, RSVP, and an IGP on the PE and Core Routers

To send multicast traffic across a Layer 3 VPN, you must configure network protocols to handle *intradomain routing* (an interior gateway protocol [IGP], such as Open Shortest Path First [OSPF] or Intermediate System-to-Intermediate System [IS-IS]), *interdomain routing* (Border Gateway Protocol [BGP]), *label switching* (Multiprotocol Label Switching [MPLS]), and *path signaling* (Resource Reservation Protocol [RSVP]). For more information about these protocols and examples of how to configure these protocols to support a Layer 3 VPN, see the *JUNOS VPNs Configuration Guide*.

NOTE: In multicast Layer 3 VPNs, the multicast PE routers must use the primary loopback address (or router ID) for sessions with their internal BGP peers. If the PE routers use a route reflector with next-hop self-configured, Layer 3 multicast over VPN does not work because PIM cannot transmit upstream interface information for multicast sources behind remote PE routers into the network core. Multicast Layer 3 VPNs require the BGP next-hop address of the VPN route to match the BGP next-hop address of the loopback VRF instance address.

Creating a Routing Instance for Multiprotocol BGP-Based Multicast VPN

By default a multiprotocol BGP-based multicast VPN routing instance attaches to both sender and receiver sites. You can also manually configure the instance to attach to only sender or only receiver sites.

To create a multicast VPN routing instance, include the mvpn statement at the [edit routing-instances *routing-instance name* protocols] hierarchy level:

```
[edit]
routing-instances {
   vpn-a {
      instance-type vrf;
      protocols {
         mvpn { # Enables BGP/MPLS multicast VPN configuration.
         }
      }
   }
}
```

NOTE: You cannot configure PIM within a nonforwarding instance. If you try to do so, the router displays a commit check error and does not complete the configuration commit process.

Option: Configuring Sender and Receiver Sites

By default, multiprotocol BGP-based VPNs are attached to both sender and receiver sites. To specify that a VPN be attached only to a sender site or only to a receiver site, include the receiver-site or sender-site statement at the [edit routing instances *routing-instance-name* protocols mvpn] hierarchy level:

```
[edit]
routing-instances {
    vpn-a {
        instance-type vrf;
        protocols {
            mvpn {
                receiver-site;
                sender-site;
            }
        }
    }
}
```

Option: Specifying Route Targets

Specifying route targets for sender and receiver sites is useful when there is a mix of sender only, receiver only, and sender and receiver sites. This is because a sender site's routing table is used for exporting routes from a sender site and importing routes from a receiver site. A receiver site's routing table is used for exporting routes from a receiver site and importing routes from a sender site. A sender and receiver site does both. The route targets configured under multicast VPNs apply only to multicast VPN AD routes of type 1, 2, 3, and 5.

A PE router with sites in a specific multicast VPN must determine whether a received automatic discovery route is from a sender site or receiver site based on the following:

- If the PE router is configured to be only in a sender site, route targets are imported only from receiver sites. Imported automatic discovery routes must be from a receiver site.
- If the PE router is configured to be only in a receiver site, route targets are imported only from sender sites. Imported automatic discovery routes must be from a sender site.
- If the PE router is configured to be both in sender and receiver sites, the following guidelines apply:
  - Along with an import route target, you can optionally configure whether the route target is from a receiver or a sender site.
  - If a configuration is not provided, an imported automatic discovery route is treated as belonging to both the sender site and the receiver site.

For more information on route targets, see the *JUNOS VPNs Configuration Guide*.

To specify route targets, include the route-target statement at the [edit routing-instances *routing-instance-name* protocols mvpn] hierarchy level:

```
[edit]
routing-instances {
   vpn-a {
      protocols {
         mvpn {
            route-target {
               export-target { # Export target for multicast VPN AD routes. Overrides default
                  # VRF export target if "export-target unicast" is not configured.
                  target target-community;
                  unicast; # Apply the VRF export target and multicast VPN export route
                     # target to multicast VPN AD routes.
                  apply-groups group-name; # Groups from which to inherit
                     # configuration data.
                  apply-groups-except group-name; # Do not inherit configuration data from
                     # these groups.
               }
               import-target { # Import target for multicast VPN AD routes. Overrides
                  # default VRF import target if "import-target unicast" is not configured.
                  target target-value { # Target community.
                     receiver target-value; # Target community used when importing receiver
                        # site routes.
                     sender target-value; # Target community used when importing sender
                        # site routers.
                  }
                  unicast { #Apply the default VRF import target or multicast VPN
                     # route-target to multicast VPN AD routes.
                     receiver;
                     sender;
                  }
                  apply-groups group-name;
                  apply-groups-except group-name;
               }
            }
         }
      }
   }
}
```

Existing vrf-import or vrf-export policies for importing and exporting vpn routes might prevent import or export of multicast VPN routes if the policies reflect routes based on the policy qualifier protocol. The workaround is to relax the policy to not reflect routes based on the protocol type or to use additional multicast-VPN-specific configuration.

If the vrf-import policy does not import bgp routes, multicast VPN routes of type 1, 2, 3, or 5 imported by BGP will be rejected. There are two workarounds:

- You can add a term to allow routes from protocol bgp with family inet-mvpn:

```
term 2 {
   from {
      protocol bgp;
      family inet-mvpn;
```

```
        community vpn_blue;
      }
      then {
        accept;
      }
    }
```

The customer can configure an import target under **mvpn**. The multicast VPN route of type 1, 2, 3, or 5 matching the configured target will be imported.

```
protocol mvpn {
  import-target {
    target target:2:2;
  }
}
```

- If the vrf-export policy uses a policy qualifier of type protocol to reject routes, the multicast VPN routes of type 1, 2, 3, or 5 will not be exported. This is because JUNOS does not support a policy qualifier for protocol multicast VPN. The workaround is to configure an export target under multicast VPN without configuring unicast:

```
protocol mvpn {
  export-target {
    target target:2:2;
    }
  }
}
```

Configuring Provider Tunnels

The source address for a PIM-SM provider tunnel is the loopback address of the loopback interface in **inet.0**.

To configure a provider tunnel, include the **provider-tunnel** statement at the [**edit routing-instances** *routing-instance-name*] hierarchy level:

```
[edit]
routing-instances {
  vpn-a {
    provider-tunnel {
      pim-asm {
        apply-groups group-name;
        apply-groups-except group-name;
        group-address address;
      }
    }
  }
}
```

Enabling Multicast VPN in BGP

You also must enable multicast VPN by including the inet-mvpn or inet6-mvpn statements at the [edit protocols bgp family] hierarchy level:

```
[edit]
protocols {
   bgp {
      family {
         inet-mvpn; # Enables IPv4 multicast VPN.
         inet6-mvpn; # Enables IPv6 multicast VPN.
      }
   }
}
```

Configuring Intra-AS Inclusive Point-to-Multipoint TE LSPs

Point-to-multipoint TE LSPs are supported as the data plane for intra-AS inclusive provider tunnels. A multicast VPN can be configured to use inclusive trees or selective trees or a combination of both. Aggregation is not supported for point-to-multipoint TE LSPs.

NOTE: Configure either LDP or regular MPLS LSPs between PE routers to ensure VPN unicast connectivity. Point-to-multipoint LSPs are used for multicast data forwarding only.

You must configure the following when configuring point-to-multipoint LSPs in provider tunnels:

- The BGP multicast VPN control plane, as described in "Creating a Routing Instance for Multiprotocol BGP-Based Multicast VPN" on page 573.
- Point-to-multipoint TE as the provider tunnel technology on each PE configured for multicast VPN that belongs to the sender site.
- Either a VT interface or a vrf-table-label on the multicast VPN instance. For more information about configuring VT interfaces, see the *JUNOS VPNs Configuration Guide*.
- Point-to-multipoint TE support on each P router.

On each PE router, a point-to-multipoint TE LSP must be configured for every multicast VPN instance that belongs to a sender site set. This means that if there are four multicast VPN instances configured on a PE router and three of these multicast VPN instances belong to sender site sets, three point-to-multipoint TE LSPs must be configured on this PE router. The PE would be the root of the three point-to-multipoint TE LSPs, and the leaves of the LSPs would be determined either dynamically or through a static configuration.

If the multicast VPN instance is configured for dynamic leaf discovery, the leaves are automatically discovered through intra-AS autodiscovery routes. The

point-to-multipoint LSPs can be signaled using default attributes or configured attributes. If you configure the multicast VPN instance to use default attributes, the LSPs cannot be signaled with bandwidth reservation and do not support CAC. Point-to-multipoint LSPs with configured attributes support both bandwidth reservation and CAC. In addition, they can be configured to support traffic engineering attributes such as fast-reroute.

If the multicast VPN instance is configured for static leaf discovery, you configure the leafs statically. Point-to-multipoint LSPs that are configured statically support all traffic engineering attributes.

To configure dynamic leaf discovery, include the label-switched-path-template statement at the [edit routing-instance *routing-instance-name* provider-tunnel rsvp-te] hierarchy level. Dynamic discovery can be configured by using default attributes with the default-template statement at the [edit routing-instance *routing-instance-name* provider-tunnel rsvp-te label-switched-path-template] hierarchy level.

If you want to signal with bandwidth reservation, use CAC, or use other traffic engineering options such as link protection, configure a template for dynamic leaf discovery by including the label-switched-path-template *template-name* statement at the [edit protocols mpls] hierarchy level:

```
[edit]
protocols {
    mpls {
        label-switched-path mvpn-example-p2mp-temlate {
            template;
            p2mp;
            link-protection;
            optimize-timer 50;
            traceoptions {
                file mvpn-a-p2mp-lsp.log;
                flag all;
            }
        }
    }
}
```

You can apply the configured or default template by including the template name at the [edit routing-instance *routing-instance-name* provider-tunnel rsvp-te label-switched-path-template] hierarchy level. Be sure to either configure a vt interface or include the vrf-table-label statement in the routing instance.

```
[edit]
routing-instance {
    routing-instance configured-dynamic-example {
    instance-type vrf;
    interface ge-1/0/0.0;
    route-distinguisher 10.255.71.1:100;
    vrf-table-label;
        provider-tunnel {
            rsvp-te label switched-path-template mvpn-example-p2mp-temlate;
        }
    }
}
```

To configure static LSPs, include the label-switched-path *label-switched-path* statement at the [edit protocols mpls] hierarchy level:

```
[edit]
protocols {
   mpls {
      label-switched-path vpls-example-p2mp-s21_lsp_a {
         to 192.168.1.1
         p2mp example-static-lsp;
      }
      label-switched-path vpls-example-p2mp-s21_lsp_b {
         to 192.168.1.2;
         p2mp example-static-lsp;
      }
   }
}
```

To apply statically configured LSPs, include the static statement at the [edit routing-instance *routing-instance-name* provider-tunnel rsvp-te static-lsp *static-lsp-name*] hierarchy level:

```
[edit]
routing-instance example-static {
   provider-tunnel {
      rsvp-te {
         static-lsp example-static-lsp;
      }
   }
}
```

Configuring Intra-AS Selective Provider Tunnels

Point-to-multipoint TE LSPs are supported as the data plane for selective provider tunnels. A multicast VPN can be configured to use inclusive trees or selective trees or a combination of both. Aggregation is not supported for point-to-multipoint TE LSPs.

NOTE: Configure either LDP or regular MPLS LSPs between PE routers to ensure VPN unicast connectivity. Point-to-multipoint LSPs are used for multicast data forwarding only.

You must configure the following when configuring point-to-multipoint LSPs in provider tunnels:

- The BGP multicast VPN control plane, as described in "Creating a Routing Instance for Multiprotocol BGP-Based Multicast VPN" on page 573.
- Point-to-multipoint TE as the provider tunnel technology on each PE configured for multicast VPN that belongs to the sender site.

- Either a VT interface or a vrf-table-label on the multicast VPN instance. For more information about configuring VT interfaces, see the *JUNOS VPNs Configuration Guide*.
- Point-to-multipoint TE support on each P router.

When selective trees are used, there must be a separate point-to-multipoint TE LSP for each multicast distribution tree in the backbone that carries traffic belonging to a specified set of one or more multicast groups, from one or more multicast VPNs. Multiple groups can be bound to the same selective point-to-multipoint LSP if the selective point-to-multipoint LSP leaves are statically configured. If the leaves are dynamically discovered, only one source or group can be bound to it.

Selective point-to-multipoint LSPs can be statically configured or triggered by a bandwidth threshold. If the threshold rate is configured, a S-PMSI auto-discovery route is generated for a particular (C-S, C-G) if it falls in the range specified by (C-S prefix, C-G prefix) and its data rate exceeds the configured threshold rate.

Below is an example configuration for point-to-multipoint LSPs on a selective tunnel with statically configured leafs:

```
[edit]
routing-instances {
   selective-tunnel-example {
      instance-type vrf;
      route-distinguisher 10.255.71.2:100;
      protocols {
         vpls {
            tunnel-services { # This enables vt interfaces for this routing instance.
         }
      }
      provider-tunnel {
         rsvp-te {
            label-switched-path-template {
               mvpn_template;
            }
         }
         }
         selective {
            group 225.10.10.1/32 {
               source 192.2.1.2/32 {
                  rsvp-te {
                     static-lsp lsp1;
                  }
               }
            }
            group 226.10.10.1/32 {
               source 192.2.1.2/32 {
                  rsvp-te {
                     static-lsp lsp1;
                  }
               }
            }
         }
      }
```

```
    }
}
```

The following example shows an example with dynamic selective trees and the default template:

```
[edit]
routing-instances {
    dynamic-selective-tunnel-example {
        instance-type vrf;
        route-distinguisher 10.255.71.2:100;
        protocols {
            vpls {
                tunnel-services {
                }
            }
        provider-tunnel {
            rsvp-te {
                label-switched-path-template {
                    default-template;
                }
            }
            selective {
                group 225.10.10.1/32 {
                    source 192.2.1.2/32 {
                        rsvp-te {
                            label-switched-path-template default-template;
                        }
                    }
                }
                group 226.10.10.1/32 {
                    source 192.2.1.2/32 {
                        rsvp-te {
                            label-switched-path-template default-template;
                        }
                    }
                }
            }
        }
    }
}
```

Configuring the Master PIM Instance on the PE Router for BGP-based Multicast VPNs

To configure the master PIM instance that communicates with other PIM neighbors, include the pim statement at the [edit protocols] hierarchy level. BGP-based multicast VPNs support sparse mode, dense mode, or sparse-dense mode. The first example shown enables PIM sparse mode.

```
[edit protocols]
pim {
    interface all {
        mode sparse;
```

```
        version 2;
    }
}
```

The next example shown enables PIM dense mode.

```
[edit protocols]
pim {
    interface all {
        mode dense;
    }
}
```

Configuring the Router's IPv4 Bootstrap Router Priority

By default, the router has a bootstrap priority of 0, which means the router can never be the bootstrap router. To modify this priority, include the bootstrap-priority statement. The router with the highest priority value is elected to be the bootstrap router.

```
[edit protocols]
pim {
    bootstrap-priority number;
}
```

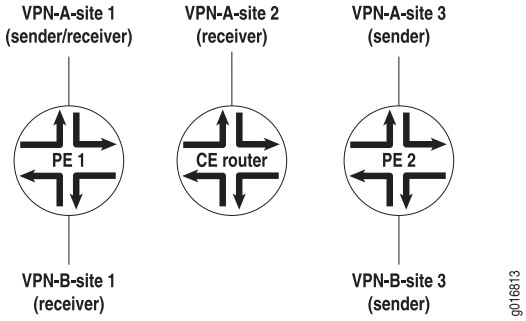
Multiprotocol BGP Multicast VPNs Example

This section contains a configuration example and commands you can issue to verify multiprotocol BGP multicast VPN router configuration.

In the example shown in Figure 60 on page 582, there are three routers: PE1, the CE router, and PE2. Each router is configured to support a specific role. This example is not a complete network.

Router PE1 is configured with multicast VPN A as a sender and a receiver site, and multicast VPN B, which is a receiver site. Router CE is configured with multicast VPN A as a receiver site. Router PE2 is configured with multicast VPN A as a sender site.

Figure 60: Multiprotocol BGP Multicast VPN Example



The relevant configuration for router PE1 follows.

Router PE1

```
[edit]
routing-instances {
    vpn-a {
        instance-type vrf;
        interface so-6/0/0.0;
        interface so-6/0/1.0;
        provider tunnel {
            rsvp-te {
                label-switched-path-template default-template;
            }
        }
        protocols {
            mvpn {
                route-target {
                    export-target unicast target:1:4;
                    import-target unicast sender target target:1:4 receiver;
                }
            }
        }
        route-distinguisher 65535:0;
        vrf-target target:1:1;
        routing-options {
            auto-export;
            static {
                route 172.16.0.0/16 next-hop so-0/0/0.0;
                route 172.17.0.0/16 next-hop so-6/0/1.0;
            }
        }
    }
}

[edit]
routing-instance {
    vpn-b {
        instance-type vrf;
        interface ge-0/3/0.0;
        provider-tunnel {
            pim-sm {
                group-address 224.1.1.2;
            }
        }
        protocols {
            mvpn {
                receiver-site;
                router-target {
                    export target:1:5;
                    import unicast;
                }
            }
        }
        route-distinguisher 65535:1;
        vrf-target target:1:2;
        routing-options {
            auto-export;
        }
    }
```

```
}
```

The relevant configuration for router PE2 follows.

Router PE2

```
[edit]
routing-instances {
   vpn-a {
      instance-type vrf;
      interface so-1/0/0.0;
      provider-tunnel {
         rsvp-te {
            label-switched-path-template default-template;
         }
      }
      protocols {
         mvpn {
            sender-site;
            route-target {
               export target:1:4;
               import unicast;
            }
         }
         route-distinguisher 65535:2;
         vrf-target target:1:1;
         routing-options {
            auto-export;
            static {
               route 172.16.0.0/16 next-hop so-0/0/0.1;
               route 172.17.0.0/16 next-hope so-6/0/1.0;
            }
         }
      }
   }
}

[edit]
routing-instance {
   vpn-b {
      instance-type vrf;
      interface ge-0/3/0.0;
      provider-tunnel {
         pim-sm {
            group-address 224.1.1.2;
         }
      }
      protocols {
         mvpn {
            sender-site;
            router-target {
               export target:1:5;
               import unicast;
            }
         }
      }
      route-distinguisher 65535:1;
      vrf-target target:1:2;
```

```
        routing-options {
            auto-export;
        }
    }
}
```

Verifying Your Work

To verify correct operation of multiprotocol BGP multicast VPNs, use the following commands:

- show mvpn c-multicast
- show mvpn instance
- show mvpn neighbor

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

- show mvpn c-multicast on page 585
- show mvpn instance on page 586
- show mvpn neighbor on page 588

show mvpn c-multicast

```
Router> show mvpn c-multicast
Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  C-mcast IPv4 (S:G)            Ptnl                              St
  192.168.195.78/32:225.5.5.5/32 PIM-SM:10.255.14.144, 239.1.1.1         RM
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-B
  C-mcast IPv4 (S:G)            Ptnl                              St
  192.168.195.94/32:226.6.6.6/32 PIM-SM:10.255.14.144, 239.2.0.0         RM

Router> show mvpn c-multicast extensive
Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  C-mcast IPv4 (S:G)            Ptnl                              St
  192.168.195.78/32:225.5.5.5/32 PIM-SM:10.255.14.144, 239.1.1.1         RM
```

```
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-B
  C-mcast IPv4 (S:G)            Ptnl                              St
  192.168.195.94/32:226.6.6.6/32 PIM-SM:10.255.14.144, 239.2.0.0         RM

Router> show mvpn c-multicast summary
Instance: VPN-A
  C-multicast IPv4 route count: 1
Instance: VPN-B
  C-multicast IPv4 route count: 2
```

show mvpn instance

```
Router> show mvpn instance
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  Provider tunnel: I-P-tnl:PIM-SM:10.255.14.144, 239.1.1.1
  Neighbor                              I-P-tnl
  10.255.14.160                         PIM-SM:10.255.14.160, 239.1.1.1
  10.255.70.17                          PIM-SM:10.255.70.17, 239.1.1.1
  C-mcast IPv4 (S:G)            Ptnl                              St
  192.168.195.78/32:225.5.5.5/32 PIM-SM:10.255.14.144, 239.1.1.1         RM
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-B
  Provider tunnel: I-P-tnl:PIM-SM:10.255.14.144, 239.2.0.0
  Neighbor                              I-P-tnl
  10.255.14.160                         PIM-SM:10.255.14.160, 239.2.0.0
  10.255.70.17                          PIM-SM:10.255.70.17, 239.2.0.0
  C-mcast IPv4 (S:G)            Ptnl                              St
  192.168.195.94/32:226.6.6.6/32 PIM-SM:10.255.14.144, 239.2.0.0         RM

Router> show mvpn instance extensive
Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  Provider tunnel: I-P-tnl:PIM-SM:10.255.14.144, 239.1.1.1
  Neighbor                              I-P-tnl
```

```
  10.255.14.160                          PIM-SM:10.255.14.160, 239.1.1.1
  10.255.70.17                           PIM-SM:10.255.70.17, 239.1.1.1
  C-mcast IPv4 (S:G)            Ptnl                              St
  192.168.195.78/32:225.5.5.5/32 PIM-SM:10.255.14.144, 239.1.1.1         RM
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-B
  Provider tunnel: I-P-tnl:PIM-SM:10.255.14.144, 239.2.0.0
  Neighbor                               I-P-tnl
  10.255.14.160                          PIM-SM:10.255.14.160, 239.2.0.0
  10.255.70.17                           PIM-SM:10.255.70.17, 239.2.0.0
  C-mcast IPv4 (S:G)            Ptnl                              St
  192.168.195.94/32:226.6.6.6/32 PIM-SM:10.255.14.144, 239.2.0.0         RM

Router> show mvpn instance
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  Provider tunnel: I-P-tnl:RSVP-TE P2MP:10.255.71.190, 27859,10.255.71.190
  Neighbor                               I-P-tnl
  10.255.71.2                           RSVP-TE P2MP:10.255.71.2, 39143,10.255.71.2

  C-mcast IPv4 (S:G)            Ptnl                              St
  192.1.1.2/32:225.10.10.1/32   RSVP-TE P2MP:10.255.71.2, 39143,10.255.71.2
    DS
  0.0.0.0/0:225.10.10.1/32
Router> show mvpn instance summary
Instance: VPN-A
  Neighbor count: 2
  C-multicast IPv4 route count: 1
Instance: VPN-B
  Neighbor count: 4
  C-multicast IPv4 route count: 2

Router> show mvpn instance VPN-A
Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  Provider tunnel: I-P-tnl:PIM-SM:10.255.14.144, 239.1.1.1
  Neighbor                               I-P-tnl
  10.255.14.160                          PIM-SM:10.255.14.160, 239.1.1.1
  10.255.70.17                           PIM-SM:10.255.70.17, 239.1.1.1
  C-mcast IPv4 (S:G)            Ptnl                              St
  192.168.195.78/32:225.5.5.5/32 PIM-SM:10.255.14.144, 239.1.1.1         RM

Router> show mvpn instance VPN-A extensive
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel
```

```
Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  Provider tunnel: I-P-tnl:PIM-SM:10.255.14.144, 239.1.1.1
  Neighbor                              I-P-tnl
  10.255.14.160                         PIM-SM:10.255.14.160, 239.1.1.1
  10.255.70.17                          PIM-SM:10.255.70.17, 239.1.1.1
  C-mcast IPv4 (S:G)            Ptnl                          St
  192.168.195.78/32:225.5.5.5/32 PIM-SM:10.255.14.144, 239.1.1.1        RM

Router> show mvpn instance VPN-A summary
Instance: VPN-A
  Neighbor count: 2
  C-multicast IPv4 route count: 1
```

show mvpn neighbor

```
Router> show mvpn neighbor
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  Neighbor                              I-P-tnl
  10.255.14.160                         PIM-SM:10.255.14.160, 239.1.1.1
  10.255.70.17                          PIM-SM:10.255.70.17, 239.1.1.1
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-B
  Neighbor                              I-P-tnl
  10.255.14.160                         PIM-SM:10.255.14.160, 239.2.0.0
  10.255.70.17                          PIM-SM:10.255.70.17, 239.2.0.0

Router> show mvpn neighbor extensive
Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  C-mcast IPv4 (S:G)            Ptnl                          St
  192.168.195.78/32:225.5.5.5/32 PIM-SM:10.255.14.144, 239.1.1.1        RM
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-B
```

```
  C-mcast IPv4 (S:G)            Ptnl                              St
  192.168.195.94/32:226.6.6.6/32 PIM-SM:10.255.14.144, 239.2.0.0         RM
```

The following is output for a p2mp configuration.

```
Router> show mvpn neighbor
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  Neighbor                              I-P-tnl
  10.255.71.2                         RSVP-TE P2MP:10.255.71.2, 39143,10.255.71.2
```

Example: Configuring MBGP Multicast VPNs

This example provides a step-by-step procedure to configure multicast services across a multiprotocol BGP (MBGP) Layer 3 virtual private network.

- Before You Begin on page 589
- Overview and Topology on page 589
- Configuration on page 590

Before You Begin

Depending on the devices you are using, you might be required to configure static routes to the:

- multicast sender.
- Fast Ethernet interface the sender is connected to on the multicast receiver.
- multicast receiver.
- Fast Ethernet interface the receiver is connected to on the multicast sender.

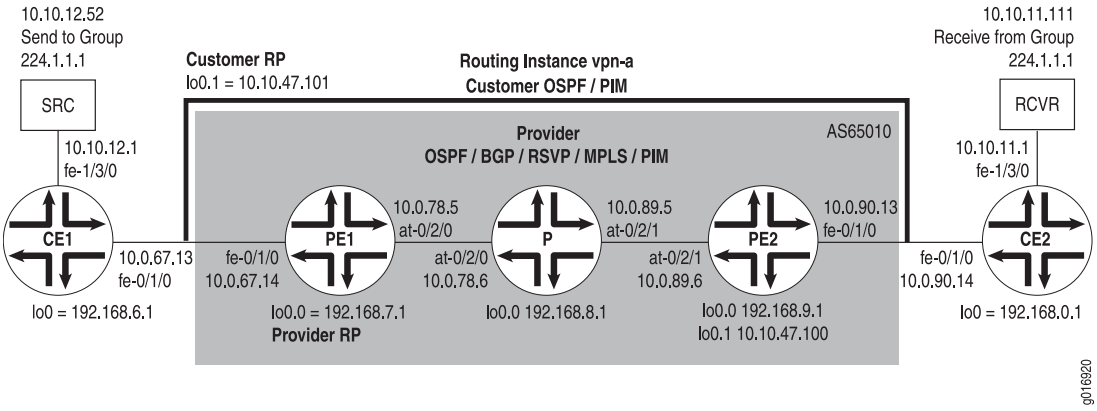
Overview and Topology

This example configures the following technologies:

- IPv4
- BGP
- OSPF
- RSVP
- MPLS
- PIM sparse mode
- Static RP

The topology of the network is shown in Figure 61 on page 590.

Figure 61: Multicast Over Layer 3 VPN Example Topology



Configuration

NOTE: In any configuration session it is a good practice to periodically verify that the configuration can be committed using the commit check command.

In this example, the router being configured is identified using the following command prompts:

- ce1 identifies the customer edge 1 (CE1) router
- pe1 identifies the provider edge 1 (PE1) router
- p identifies the provider core (P) router
- ce2 identifies the customer edge 2 (CE2) router
- pe2 identifies the provider edge 2 (PE2) router

To configure MBGP multicast VPNs for the network shown in Figure 61 on page 590, perform the following steps:

Configuring Interfaces

Step-by-Step Procedure

1. On each router, configure an IP address on the loopback logical interface 0 (lo0.0).

```
user@ce1# set interfaces lo0 unit 0 family inet address 192.168.6.1/32 primary

user@pe1# set interfaces lo0 unit 0 family inet address 192.168.7.1/32
    primary

user@p# set interfaces lo0 unit 0 family inet address 192.168.8.1/32 primary

user@pe2# set interfaces lo0 unit 0 family inet address 192.168.9.1/32
    primary

user@ce2# set interfaces lo0 unit 0 family inet address 192.168.0.1/32 primary
```

   You can verify this step using the show interfaces terse command.

2. On the PE and CE routers, configure the IP address and protocol family on the Fast Ethernet interfaces. Specify the inet family type.

```
user@ce1# set interfaces fe-1/3/0 unit 0 family inet address 10.10.12.1/24
user@ce1# set interfaces fe-0/1/0 unit 0 family inet address 10.0.67.13/30

user@pe1# set interfaces fe-0/1/0 unit 0 family inet address 10.0.67.14/30

user@pe2# set interfaces fe-0/1/0 unit 0 family inet address 10.0.90.13/30

user@ce2# set interfaces fe-0/1/0 unit 0 family inet address 10.0.90.14/30
user@ce2# set interfaces fe-1/3/0 unit 0 family inet address 10.10.11.1/24
```

   You can verify this step using the show interfaces terse command.

3. On the PE and P routers, configure the ATM interfaces' VPI and maximum VCs. If the default PIC type is different on directly connected ATM interfaces, configure the PIC type to be the same. Configure the logical interface VCI, protocol family, local IP address, and destination IP address.

```
user@pe1# set interfaces at-0/2/0 atm-options pic-type atm1
user@pe1# set interfaces at-0/2/0 atm-options vpi 0 maximum-vcs 256
user@pe1# set interfaces at-0/2/0 unit 0 vci 0.128
user@pe1# set interfaces at-0/2/0 unit 0 family inet address 10.0.78.5/32
    destination 10.0.78.6

user@p# set interfaces at-0/2/0 atm-options pic-type atm1
user@p# set interfaces at-0/2/0 atm-options vpi 0 maximum-vcs 256
user@p# set interfaces at-0/2/0 unit 0 vci 0.128
user@p# set interfaces at-0/2/0 unit 0 family inet address 10.0.78.6/32
    destination 10.0.78.5
user@p# set interfaces at-0/2/1 atm-options pic-type atm1
```

```
user@p# set interfaces at-0/2/1 atm-options vpi 0 maximum-vcs 256
user@p# set interfaces at-0/2/1 unit 0 vci 0.128
user@p# set interfaces at-0/2/1 unit 0 family inet address 10.0.89.5/32
  destination 10.0.89.6

user@pe2# set interfaces at-0/2/1 atm-options pic-type atm1
user@pe2# set interfaces at-0/2/1 atm-options vpi 0 maximum-vcs 256
user@pe2# set interfaces at-0/2/1 unit 0 vci 0.128
user@pe2# set interfaces at-0/2/1 unit 0 family inet address 10.0.89.6/32
  destination 10.0.89.5
```

You can verify this step using the show configuration interfaces command.

Configuring OSPF

Step-by-Step Procedure

1. On the P and PE routers, configure the provider instance of Open Shortest Path First (OSPF). Specify the lo0.0 and ATM core-facing logical interfaces. The provider instance of OSPF on the PE router forms adjacencies with the OSPF neighbors on the other PE router and the P router.

```
user@pe1# set protocols ospf area 0.0.0.0 interface at-0/2/0.0
user@pe1# set protocols ospf area 0.0.0.0 interface lo0.0

user@p# set protocols ospf area 0.0.0.0 interface lo0.0
user@p# set protocols ospf area 0.0.0.0 interface all
user@p# set protocols ospf area 0.0.0.0 interface fxp0 disable

user@pe2# set protocols ospf area 0.0.0.0 interface lo0.0
user@pe2# set protocols ospf area 0.0.0.0 interface at-0/2/1.0
```

   You can verify this step using the show ospf interfaces command.

2. On the CE routers, configure the customer instance of Open Shortest Path First (OSPF). Specify the loopback and Fast Ethernet logical interfaces . The customer instance of OSPF on the CE routers form adjacencies with the neighbors within the VPN routing instance of OSPF on the PE routers.

```
user@ce1# set protocols ospf area 0.0.0.0 interface fe-0/1/0.0
user@ce1# set protocols ospf area 0.0.0.0 interface fe-1/3/0.0
user@ce1# set protocols ospf area 0.0.0.0 interface lo0.0

user@ce2# set protocols ospf area 0.0.0.0 interface fe-0/1/0.0
user@ce2# set protocols ospf area 0.0.0.0 interface fe-1/3/0.0
user@ce2# set protocols ospf area 0.0.0.0 interface lo0.0
```

   You can verify this step using the show ospf interfaces command.

3. On the P and PE routers, configure OSPF traffic engineering support for the provider instance of OSPF.

The shortcuts statement enables the master instance of OSPF to use a label-switched path as the next hop.

```
user@pe1# set protocols ospf traffic-engineering shortcuts

user@p# set protocols ospf traffic-engineering shortcuts

user@pe2# set protocols ospf traffic-engineering shortcuts
```

You can verify this step using the show ospf overview and show configuration protocols ospf commands.

Configuring BGP

Step-by-Step Procedure

1. On the P router, configure BGP for the VPN. The local address is the local lo0.0 address. The neighbor addresses are the PE routers' lo0.0 addresses.

   The unicast statement enables the router to use BGP to advertise network layer reachability information (NLRI). The signaling statement enables the router to use BGP as the signaling protocol for the VPN.

   ```
   user@p# set protocols bgp group group-mvpn type internal
   user@p# set protocols bgp group group-mvpn local-address 192.168.8.1
   user@p# set protocols bgp group group-mvpn family inet unicast
   user@p# set protocols bgp group group-mvpn family inet-mvpn signaling
   user@p# set protocols bgp group group-mvpn neighbor 192.168.9.1
   user@p# set protocols bgp group group-mvpn neighbor 192.168.7.1
   ```

   You can verify this step using the show configuration protocols bgp command.

2. On the PE and P routers, configure the BGP local autonomous system number.

   ```
   user@pe1# set routing-options autonomous-system 0.65010

   user@p# set routing-options autonomous-system 0.65010

   user@pe2# set routing-options autonomous-system 0.65010
   ```

   You can verify this step using the show configuration routing-options command.

3. On the PE routers, configure BGP for the VPN. Configure the local address as the local lo0.0 address. The neighbor addresses are the lo0.0 addresses of the P router and the other PE router.

   ```
   user@pe1# set protocols bgp group group-mvpn type internal
   user@pe1# set protocols bgp group group-mvpn local-address 192.168.7.1
   user@pe1# set protocols bgp group group-mvpn family inet-vpn unicast
   user@pe1# set protocols bgp group group-mvpn family inet-mvpn signaling
   user@pe1# set protocols bgp group group-mvpn neighbor 192.168.9.1
   user@pe1# set protocols bgp group group-mvpn neighbor 192.168.8.1
   ```

```
user@pe2# set protocols bgp group group-mvpn type internal
user@pe2# set protocols bgp group group-mvpn local-address 192.168.9.1
user@pe2# set protocols bgp group group-mvpn family inet-vpn unicast
user@pe2# set protocols bgp group group-mvpn family inet-mvpn signaling
user@pe2# set protocols bgp group group-mvpn neighbor 192.168.7.1
user@pe2# set protocols bgp group group-mvpn neighbor 192.168.8.1
```

You can verify this step using the show bgp group command.

4. On the PE routers, configure a policy to export the BGP routes into OSPF.

```
user@pe1# set policy-options policy-statement bgp-to-ospf from protocol bgp
user@pe1# set policy-options policy-statement bgp-to-ospf then accept
```

```
user@pe2# set policy-options policy-statement bgp-to-ospf from protocol bgp
user@pe2# set policy-options policy-statement bgp-to-ospf then accept
```

You can verify this step using the show policy bgp-to-ospf command.

Configuring RSVP

Step-by-Step Procedure

1. On the PE routers, enable RSVP on the interfaces that participate in the LSP. Configure the Fast Ethernet and ATM logical interfaces.

```
user@pe1# set protocols rsvp interface fe-0/1/0.0
user@pe1# set protocols rsvp interface at-0/2/0.0
```

```
user@pe2# set protocols rsvp interface fe-0/1/0.0
user@pe2# set protocols rsvp interface at-0/2/1.0
```

2. On the P router, enable RSVP on the interfaces that participate in the LSP. Configure the ATM logical interfaces.

```
user@p# set protocols rsvp interface at-0/2/0.0
user@p# set protocols rsvp interface at-0/2/1.0
```

You can verify these steps using the show configuration protocols rsvp command.

Configuring MPLS

Step-by-Step Procedure

1. On the PE routers, configure an MPLS LSP to the PE router that is the LSP egress point. Specify the IP address of the lo0.0 interface on the router at the other end of the LSP. Configure MPLS on the ATM, Fast Ethernet, and lo0.0 interfaces.

   To help identify each LSP when troubleshooting, configure a different LSP name on each PE router. In this example we use the name to-pe2 as the name for the LSP configured on PE1 and to-pe1 as the name for the LSP configured on PE2.

```
user@pe1# set protocols mpls label-switched-path to-pe2 to 192.168.9.1
```

```
user@pe1# set protocols mpls interface fe-0/1/0.0
user@pe1# set protocols mpls interface at-0/2/0.0
user@pe1# set protocols mpls interface lo0.0

user@pe2# set protocols mpls label-switched-path to-pe1 to 192.168.7.1
user@pe2# set protocols mpls interface fe-0/1/0.0
user@pe2# set protocols mpls interface at-0/2/1.0
user@pe2# set protocols mpls interface lo0.0
```

You can verify this step using the show configuration protocols mpls and show route label-switched-path to-pe1 commands.

After the configuration is committed, you can verify that the LSP is operational using the show mpls lsp name to-pe1 and show mpls lsp name to-pe2 commands.

2. On the P router, enable MPLS. Specify the ATM interfaces connected to the PE routers.

```
user@p# set protocols mpls interface at-0/2/0.0
user@p# set protocols mpls interface at-0/2/1.0
```

You can verify this step using the show mpls interface command.

3. On the PE and P routers, configure the protocol family on the ATM interfaces associated with the LSP. Specify the mpls family type.

```
user@pe1# set interfaces at-0/2/0 unit 0 family mpls

user@p# set interfaces at-0/2/0 unit 0 family mpls
user@p# set interfaces at-0/2/1 unit 0 family mpls

user@pe2# set interfaces at-0/2/1 unit 0 family mpls
```

You can verify this step using the show mpls interface command.

Configuring the VRF Routing Instance

Step-by-Step Procedure

1. On the PE routers, configure a routing instance for the VPN and specify the vrf instance type. Add the Fast Ethernet and lo0.1 customer-facing interfaces. Configure the VPN instance of OSPF and include the BGP-to-OSPF export policy.

```
user@pe1# set routing-instances vpn-a instance-type vrf
user@pe1# set routing-instances vpn-a interface lo0.1
user@pe1# set routing-instances vpn-a interface fe-0/1/0.0
user@pe1# set routing-instances vpn-a protocols ospf export bgp-to-ospf
user@pe1# set routing-instances vpn-a protocols ospf area 0.0.0.0 interface
  all

user@pe2# set routing-instances vpn-a instance-type vrf
user@pe2# set routing-instances vpn-a interface lo0.1
user@pe2# set routing-instances vpn-a interface fe-0/1/0.0
```

```
user@pe2# set routing-instances vpn-a protocols ospf export bgp-to-ospf
user@pe2# set routing-instances vpn-a protocols ospf area 0.0.0.0 interface
  all
```

You can verify this step using the show configuration routing-instances vpn-a command.

2. On the PE routers, configure a route distinguisher for the routing instance. A route distinguisher allows the router to distinguish between two identical IP prefixes used as VPN routes. Configure a different route distinguisher on each PE router. This example uses 65010:1 on PE1 and 65010:2 on PE2.

```
user@pe1# set routing-instances vpn-a route-distinguisher 65010:1
```

```
user@pe2# set routing-instances vpn-a route-distinguisher 65010:2
```

You can verify this step using the show configuration routing-instances vpn-a command.

3. On the PE routers, configure default VRF import and export policies. Based on this configuration, BGP automatically generates local routes corresponding to the route target referenced in the VRF import policies. This example uses 2:1 as the route target.

NOTE: You must configure the same route target on each PE router for a given VPN routing instance.

```
user@pe1# set routing-instances vpn-a vrf-target target:2:1
```

```
user@pe2# set routing-instances vpn-a vrf-target target:2:1
```

You can verify this step using the show configuration routing-instances vpn-a command.

4. On the PE routers, configure the VPN routing instance for multicast support.

```
user@pe1# set routing-instances vpn-a protocols mvpn
```

```
user@pe2# set routing-instances vpn-a protocols mvpn
```

You can verify this step using the show configuration routing-instance vpn-a command.

5. On the PE routers, configure an IP address on the loopback logical interface 1 (lo0.1) used in the customer routing instance VPN.

```
user@pe1# set interfaces lo0 unit 1 family inet address 10.10.47.101/32
```

```
user@pe2# set interfaces lo0 unit 1 family inet address 10.10.47.100/32
```

You can verify this step using the show interfaces terse command.

Configuring PIM

Step-by-Step Procedure

1. On the PE and P routers, enable the provider instance of PIM. Add the core-facing ATM interfaces. On the PE routers, also configure the lo0.0 interface. Specify the mode as sparse and the version as 2.

```
user@pe1# set protocols pim interface at-0/2/0.0 mode sparse
user@pe1# set protocols pim interface at-0/2/0.0 version 2
user@pe1# set protocols pim interface lo0.0 mode sparse
user@pe1# set protocols pim interface lo0.0 version 2

user@p# set protocols pim interface at-0/2/0.0 mode sparse
user@p# set protocols pim interface at-0/2/0.0 version 2
user@p# set protocols pim interface at-0/2/1.0 mode sparse
user@p# set protocols pim interface at-0/2/1.0 version 2

user@pe2# set protocols pim interface at-0/2/1.0 mode sparse
user@pe2# set protocols pim interface at-0/2/1.0 version 2
user@pe2# set protocols pim interface lo0.0 mode sparse
user@pe2# set protocols pim interface lo0.0 version 2
```

   You can verify this step using the show pim interfaces command.

2. On the PE routers, enable the VPN customer instance of PIM. Configure the lo0.1 and the customer-facing Fast Ethernet interface. Specify the mode as sparse and the version as 2.

```
user@pe1# set routing-instances vpn-a protocols pim interface lo0.1 mode
  sparse
user@pe1# set routing-instances vpn-a protocols pim interface lo0.1 version
  2
user@pe1# set routing-instances vpn-a protocols pim interface fe-0/1/0.0
  mode sparse
user@pe1# set routing-instances vpn-a protocols pim interface fe-0/1/0.0
  version 2

user@pe2# set routing-instances vpn-a protocols pim interface lo0.1 mode
  sparse
user@pe2# set routing-instances vpn-a protocols pim interface lo0.1 version
  2
user@pe2# set routing-instances vpn-a protocols pim interface fe-0/1/0.0
  mode sparse
user@pe2# set routing-instances vpn-a protocols pim interface fe-0/1/0.0
  version 2
```

   You can verify this step using the show pim interfaces instance vpn-a command.

3. On the CE routers, enable the customer instance of PIM. In this example we configure all interfaces. Specify the mode as sparse and the version as 2.

```
user@ce1# set protocols pim interface all
```

```
user@ce2# set protocols pim interface all mode sparse
user@ce2# set protocols pim interface all version 2
```

You can verify this step using the show pim interfaces command.

Configuring the Provider Tunnel

Step-by-Step Procedure

1. On the PE1 router, configure the provider tunnel. Specify the multicast address to be used.

   The provider-tunnel statement instructs the router to send multicast traffic across a tunnel. The pim-asm statement instructs the router to accept the multicast stream from any source.

   ```
   user@pe1# set routing-instances vpn-a provider-tunnel pim-asm group-address
     224.1.1.1
   ```

   You can verify this step using the show configuration routing-instance vpn-a command.

2. On the PE2 router, configure the provider tunnel. Specify the multicast address to be used.

   ```
   user@pe2# set routing-instances vpn-a provider-tunnel pim-asm group-address
     224.1.1.1
   ```

   You can verify this step using the show configuration routing-instance vpn-a command.

Configuring the Rendezvous Point

Step-by-Step Procedure

1. Configure the PE1 router to be the rendezvous point for the provider instance of PIM. Specify the lo0.0 address of the PE1 router.

   ```
   user@pe1# set protocols pim rp local address 192.168.7.1
   ```

   You can verify this step using the show pim rps command.

2. Configure the static rendezvous point on the P router and PE2 router for the provider instance of PIM. Specify the lo0.0 address of the PE1 router. Specify the version as 2.

   ```
   user@p# set protocols pim rp static address 192.168.7.1 version 2
   ```

   ```
   user@pe2# set protocols pim rp static address 192.168.7.1 version 2
   ```

   You can verify this step using the show pim rps command.

3. Configure the PE1 router to be the rendezvous point for the customer instance of PIM. Specify the lo0.1 address of the PE1 router. Specify the multicast address to be used.

```
user@pe1# set routing-instances vpn-a protocols pim rp local address
  10.10.47.101
user@pe1# set routing-instances vpn-a protocols pim rp local group-ranges
  224.1.1.1/32
```

   You can verify this step using the show pim rps instance vpn-a command.

4. On the PE2 router, configure the static rendezvous point for the customer instance of PIM. Specify the lo0.1 address of the PE1 router.

```
user@pe2# set routing-instances vpn-a protocols pim rp static address
  10.10.47.101
```

   You can verify this step using the show pim rps instance vpn-a command.

5. On the CE routers, configure the static rendezvous point for the customer instance of PIM. Specify the lo0.1 address of the PE1 router.

```
user@ce1# set protocols pim rp static address 10.10.47.101 version 2

user@ce2# set protocols pim rp static address 10.10.47.101 version 2
```

   You can verify this step using the show pim rps command.

6. Use the commit check command to verify that the configuration can be successfully committed. If the configuration passes the check, commit the configuration.
7. Start the multicast sender device connected to CE1.
8. Start the multicast receiver device connected to CE2.
9. Verify that the receiver is receiving the multicast stream.
10. Use show commands to verify the routing, VPN, and multicast operation.

Results The relevant sample configuration for the CE1 router follows.

Router CE1

```
interfaces {
    lo0 {
        unit 0 {
            family inet {
                address 192.168.6.1/32 {
                    primary;
                }
            }
        }
    }
    fe-0/1/0 {
        unit 0 {
            family inet {
                address 10.0.67.13/30;
            }
```

```
        }
      }
      fe-1/3/0 {
        unit 0 {
          family inet {
            address 10.10.12.1/24;
          }
        }
      }
    }
    protocols {
      ospf {
        area 0.0.0.0 {
          interface fe-0/1/0.0;
          interface lo0.0;
          interface fe-1/3/0.0;
        }
      }
      pim {
        rp {
          static {
            address 10.10.47.101 {
              version 2;
            }
          }
        }
        interface all;
      }
    }
```

The relevant sample configuration for the PE1 router follows.

Router PE1

```
interfaces {
  lo0 {
    unit 0 {
      family inet {
        address 192.168.7.1/32 {
          primary;
        }
      }
    }
  }
  fe-0/1/0 {
    unit 0 {
      family inet {
        address 10.0.67.14/30;
      }
    }
  }
  at-0/2/0 {
    atm-options {
      pic-type atm1;
      vpi 0 {
        maximum-vcs 256;
      }
```

```
        }
        unit 0 {
            vci 0.128;
            family inet {
                address 10.0.78.5/32 {
                    destination 10.0.78.6;
                }
            }
            family mpls;
        }
    }
    lo0 {
        unit 1 {
            family inet {
                address 10.10.47.101/32;
            }
        }
    }
}
routing-options {
    autonomous-system 0.65010;
}
protocols {
    rsvp {
        interface fe-0/1/0.0;
        interface at-0/2/0.0;
    }
    mpls {
        label-switched-path to-pe2 {
            to 192.168.9.1;
        }
        interface fe-0/1/0.0;
        interface at-0/2/0.0;
        interface lo0.0;
    }
    bgp {
        group group-mvpn {
            type internal;
            local-address 192.168.7.1;
            family inet-vpn {
                unicast;
            }
            family inet-mvpn {
                signaling;
            }
            neighbor 192.168.9.1;
            neighbor 192.168.8.1;
        }
    }
    ospf {
        traffic-engineering {
            shortcuts;
        }
        area 0.0.0.0 {
            interface at-0/2/0.0;
            interface lo0.0;
```

```
        }
    }
    pim {
        rp {
            local {
                address 192.168.7.1;
            }
        }
        interface at-0/2/0.0 {
            mode sparse;
            version 2;
        }
        interface lo0.0 {
            mode sparse;
            version 2;
        }
    }
}
policy-options {
    policy-statement bgp-to-ospf {
        from protocol bgp;
        then accept;
    }
}
routing-instances {
    vpn-a {
        instance-type vrf;
        interface lo0.1;
        interface fe-0/1/0.0;
        route-distinguisher 65010:1;
        provider-tunnel {
            pim-asm {
                group-address 224.1.1.1;
            }
        }
        vrf-target target:2:1;
        protocols {
            ospf {
                export bgp-to-ospf;
                area 0.0.0.0 {
                    interface all;
                }
            }
            pim {
                rp {
                    local {
                        address 10.10.47.101;
                        group-ranges {
                            224.1.1.1/32;
                        }
                    }
                }
                interface lo0.1 {
                    mode sparse;
                    version 2;
                }
```

```
            interface fe-0/1/0.0 {
               mode sparse;
               version 2;
            }
         }
         mvpn;
      }
   }
}
```

The relevant sample configuration for the P router follows.

Router P

```
interfaces {
   lo0 {
      unit 0 {
         family inet {
            address 192.168.8.1/32 {
               primary;
            }
         }
      }
   }
   at-0/2/0 {
      atm-options {
         pic-type atm1;
         vpi 0 {
            maximum-vcs 256;
         }
      }
      unit 0 {
         vci 0.128;
         family inet {
            address 10.0.78.6/32 {
               destination 10.0.78.5;
            }
         }
         family mpls;
      }
   }
   at-0/2/1 {
      atm-options {
         pic-type atm1;
         vpi 0 {
            maximum-vcs 256;
         }
      }
      unit 0 {
         vci 0.128;
         family inet {
            address 10.0.89.5/32 {
               destination 10.0.89.6;
            }
         }
         family mpls;
      }
```

```
    }
}
routing-options {
    autonomous-system 0.65010;
}
protocols {
    rsvp {
        interface at-0/2/0.0;
        interface at-0/2/1.0;
    }
    mpls {
        interface at-0/2/0.0;
        interface at-0/2/1.0;
    }
    bgp {
        group group-mvpn {
            type internal;
            local-address 192.168.8.1;
            family inet {
                unicast;
            }
            family inet-mvpn {
                signaling;
            }
            neighbor 192.168.9.1;
            neighbor 192.168.7.1;
        }
    }
    ospf {
        traffic-engineering {
            shortcuts;
        }
        area 0.0.0.0 {
            interface lo0.0;
            interface all;
            interface fxp0.0 {
                disable;
            }
        }
    }
    pim {
        rp {
            static {
                address 192.168.7.1 {
                    version 2;
                }
            }
        }
        interface at-0/2/0.0 {
            mode sparse;
            version 2;
        }
        interface at-0/2/1.0 {
            mode sparse;
            version 2;
        }
```

```
    }
}
```

The relevant sample configuration for the PE2 router follows.

Router PE2

```
interfaces {
    lo0 {
        unit 0 {
            family inet {
                address 192.168.9.1/32 {
                    primary;
                }
            }
        }
    }
    fe-0/1/0 {
        unit 0 {
            family inet {
                address 10.0.90.13/30;
            }
        }
    }
    at-0/2/1 {
        atm-options {
            pic-type atm1;
            vpi 0 {
                maximum-vcs 256;
            }
        }
        unit 0 {
            vci 0.128;
            family inet {
                address 10.0.89.6/32 {
                    destination 10.0.89.5;
                }
            }
            family mpls;
        }
    }
    lo0 {
        unit 1 {
            family inet {
                address 10.10.47.100/32;
            }
        }
    }
}
routing-options {
    autonomous-system 0.65010;
}
protocols {
    rsvp {
        interface fe-0/1/0.0;
        interface at-0/2/1.0;
    }
```

```
    mpls {
        label-switched-path to-pe1 {
            to 192.168.7.1;
        }
        interface lo0.0;
        interface fe-0/1/0.0;
        interface at-0/2/1.0;
    }
    bgp {
        group group-mvpn {
            type internal;
            local-address 192.168.9.1;
            family inet-vpn {
                unicast;
            }
            family inet-mvpn {
                signaling;
            }
            neighbor 192.168.7.1;
            neighbor 192.168.8.1;
        }
    }
    ospf {
        traffic-engineering {
            shortcuts;
        }
        area 0.0.0.0 {
            interface lo0.0;
            interface at-0/2/1.0;
        }
    }
    pim {
        rp {
            static {
                address 192.168.7.1 {
                    version 2;
                }
            }
        }
        interface lo0.0 {
            mode sparse;
            version 2;
        }
        interface at-0/2/1.0 {
            mode sparse;
            version 2;
        }
    }
}
policy-options {
    policy-statement bgp-to-ospf {
        from protocol bgp;
        then accept;
    }
}
routing-instances {
```

```
    vpn-a {
        instance-type vrf;
        interface fe-0/1/0.0;
        interface lo0.1;
        route-distinguisher 65010:2;
        provider-tunnel {
            pim-asm {
                group-address 224.1.1.1;
            }
        }
        vrf-target target:2:1;
        protocols {
            ospf {
                export bgp-to-ospf;
                area 0.0.0.0 {
                    interface all;
                }
            }
            pim {
                rp {
                    static {
                        address 10.10.47.101;
                    }
                }
                interface fe-0/1/0.0 {
                    mode sparse;
                    version 2;
                }
                interface lo0.1 {
                    mode sparse;
                    version 2;
                }
            }
            mvpn;
        }
    }
}
```

The relevant sample configuration for the CE2 router follows.

Router CE2

```
interfaces {
    lo0 {
        unit 0 {
            family inet {
                address 192.168.0.1/32 {
                    primary;
                }
            }
        }
    }
    fe-0/1/0 {
        unit 0 {
            family inet {
                address 10.0.90.14/30;
            }
```

```
        }
      }
      fe-1/3/0 {
        unit 0 {
          family inet {
            address 10.10.11.1/24;
          }
          family inet6 {
            address fe80::205:85ff:fe88:ccdb/64;
          }
        }
      }
    }
    protocols {
      ospf {
        area 0.0.0.0 {
          interface fe-0/1/0.0;
          interface lo0.0;
          interface fe-1/3/0.0;
        }
      }
      pim {
        rp {
          static {
            address 10.10.47.101 {
              version 2;
            }
          }
        }
        interface all {
          mode sparse;
          version 2;
        }
      }
    }
```

Dual PIM Draft-Rosen Multicast VPN Operation

The operation of Draft-rosen multicast within a Layer 3 VPN domain with provider tunnels operating in any-source (ASM) multicast mode occurs in multiple stages, which are shown in Figure 62 on page 609 and described on the following pages.

Figure 62: Multicast Over Layer 3 VPN Operation

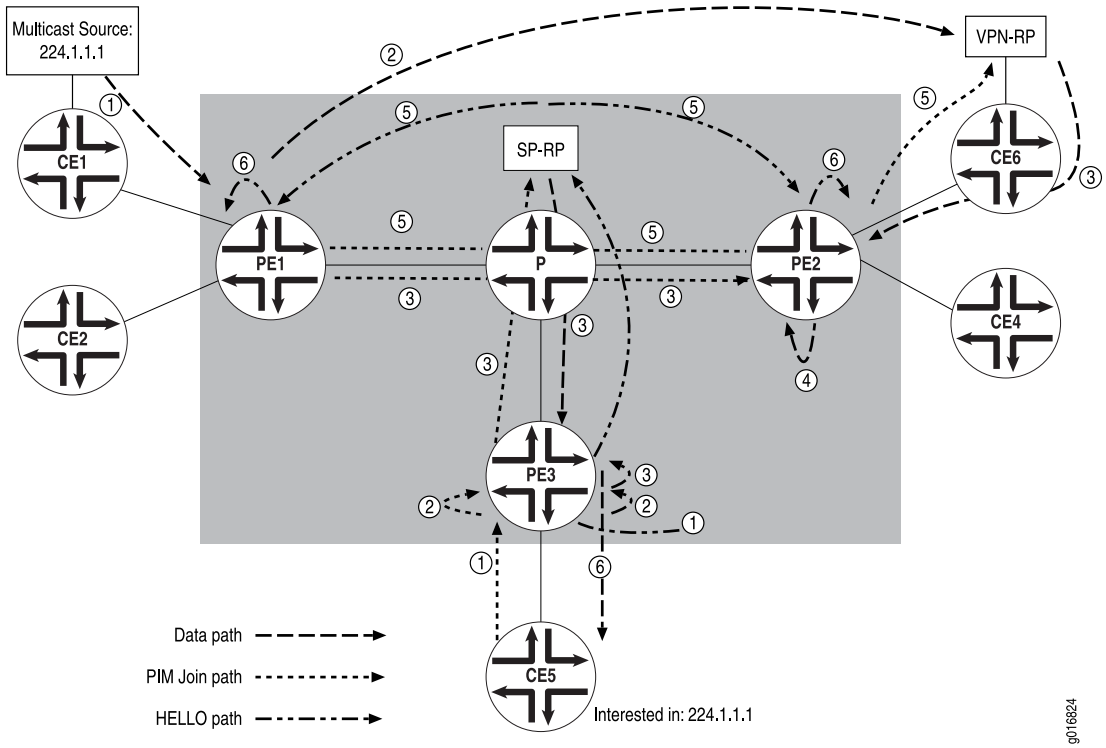


Figure 62 on page 609 shows the various stages that multicast packets pass through in a Layer 3 VPN environment.

- Stage 1: PIM HELLO

1. PIM is configured as part of a VPN routing instance and the configuration is committed. For M-series and T-series routing platforms, a virtual multicast tunnel interface (mt-*fpc/pic/port.abcde*) is created if a Tunnel Services Physical Interface Card (PIC) is installed on the router. On MX-series routers, you can create a virtual multicast tunnel interface by including the tunnel-services statement at the [edit chassis fpc *slot-number* pic *number*] hierarchy level. For more information about configuring tunnel interfaces on MX–series routers, see the *JUNOS System Basics Configuration Guide*. The virtual multicast tunnel interface is used to communicate between the PIM instance within the VRF and the master PIM instance.
2. A PIM HELLO is sent from the VRF across the mt interface. When this happens, a GRE header is prepended to the PIM HELLO with fields containing the VPN group address and the loopback address of the PE router.
3. A PIM register header is prepended to the HELLO as the packet is looped through the pe (PIM encapsulation) interface. This header contains the destination address of the SP-RP and the loopback address of the PE router.
4. The packet is sent to the SP-RP.

5. The SP-RP de-encapsulates the top header off the packet as it travels through the pd (PIM de-encapsulation) interface and sends the remaining GRE encapsulated HELLO to all of the PE routers.
6. The master PIM instance on the PE router handles the GRE encapsulated packet. Because the VPN group address is contained in the packet, the master PIM instance de-encapsulates the packet and sends the HELLO over the mt interface to reach the desired VPN group address within the VRF.

- Stage 2: PIM Join message

1. Router CE5 is interested in receiving from multicast source 224.1.1.1, so a PIM Join message is sent from Router CE5 to Router PE3.
2. The PIM Join message is sent through the mt interface and a GRE header is prepended to it. The GRE header contains the VPN group ID and the loopback address of Router PE3.
3. The GRE encapsulated Join message is sent to other PE routers.
4. Router PE2 receives the packet. Because the VPN C-RP is behind Router PE2, Router PE2 sends the packet through the mt interface, which strips off the GRE header.
5. The PIM Join message is now sent to the VPN C-RP.

- Stage 3: Multicast forwarding

1. The source behind Router CE1 is sending to group 224.1.1.1. The designated router (DR) behind the CE router encapsulates this packet into a PIM register.
2. Because the packet already has the PIM register header, it is forwarded to the VPN C-RP by unicast routing over the Layer 3 VPN.
3. The VPN C-RP de-encapsulates the data packet and sends it out the downstream interfaces (which include the return path interface leading to Router PE3). Router P also forwards the packet to Router PE3.
4. The data packet is sent through the mt interface on Router PE2. In the process, the GRE header is prepended to the packet.
5. The packet is sent to the PE routers with GRE header intact.
6. The "interested" PE routers strip the GRE header off the packet and forward it to the CE routers that requested the PIM join. If there are no PIM-join messages for this group at this site, the PE router drops the packet.

When PIM is configured within a routing instance, two mt interfaces are created:

- mt-*xxxxx* range is 32768 through 49151 for mt-encap
- mt-*yyyyy* range is 49152 through 65535 for mt-decap

PIM is run only on the mt-encap interface. The mt-decap interface is used to populate downstream interface information.

Configuring Draft-Rosen Multicast VPNs

Draft-rosen multicast VPN with provider tunnels operating in any-source multicast (ASM) mode is a legacy feature. We recommend that you implement multiprotocol BGP-based multicast VPNs for multicast VPNs. Use this section to maintain your already existing draft-rosen multicast VPNs with provider tunnels operating in ASM mode. To implement this type of draft-rosen IPv4 multicast for a Layer 3 VPN, configure the following:

- Configuring BGP, MPLS, RSVP, and an IGP on the PE and Core Routers on page 611
- Creating a Unique Logical Loopback Interface for the Routing Instance on page 611
- Configuring the Master PIM Instance on the PE Router in the Service Provider Network on page 612
- Configuring PIM and the VPN Group Address in a Routing Instance on page 612
- Option: Configuring PIM Sparse Mode Graceful Restart for a Layer 3 VPN on page 613
- Option: Configuring Multicast Distribution Trees for Data on page 614
- Option: Configuring MSDP Within a Layer 3 VPN on page 615

Configuring BGP, MPLS, RSVP, and an IGP on the PE and Core Routers

To send multicast traffic across a Layer 3 VPN, you must configure network protocols to handle *intradomain routing* (an interior gateway protocol [IGP], such as Open Shortest Path First [OSPF] or Intermediate System-to-Intermediate System [IS-IS]), *interdomain routing* (Border Gateway Protocol [BGP]), *label switching* (Multiprotocol Label Switching [MPLS]), and *path signaling* (Resource Reservation Protocol [RSVP]). For more information about these protocols and examples of how to configure these protocols to support a Layer 3 VPN, see the *JUNOS VPNs Configuration Guide*.

NOTE: In multicast Layer 3 VPNs, the multicast PE routers must use the primary loopback address (or router ID) for sessions with their internal BGP peers. If the PE routers use a route reflector with next-hop self-configured, Layer 3 multicast over VPN does not work because PIM cannot transmit upstream interface information for multicast sources behind remote PE routers into the network core. Multicast Layer 3 VPNs require the BGP next-hop address of the VPN route to match the BGP next-hop address of the loopback VRF instance address.

Creating a Unique Logical Loopback Interface for the Routing Instance

To facilitate the PIM protocol within a Layer 3 VPN, configure a unique loopback interface for the routing instance at the [edit interfaces lo0 unit] hierarchy level:

[edit interfaces]

```
lo0 {
   unit 1 {
      family inet {
         address  ip-address;
      }
   }
}
```

Configuring the Master PIM Instance on the PE Router in the Service Provider Network

To configure the master PIM instance that communicates with other PIM neighbors and the SP-RP within the service provider network, include the pim statement at the [edit protocols] hierarchy level. The example shown enables PIM sparse mode.

```
[edit protocols]
pim {
   rp {
      static {
         address ip-address;
      }
   }
   interface all {
      mode sparse;
      version 2;
   }
}
```

Configuring PIM and the VPN Group Address in a Routing Instance

The configuration syntax for PIM in a Layer 3 instance is available at the [edit routing-instances protocols pim] hierarchy level. It is similar to the global PIM configuration syntax found at the [edit protocols pim] hierarchy level.

In JUNOS Release 5.3 and later, you can include the vpn-group-address statement at the [edit routing-instances *instance-name* protocols pim] hierarchy level. You include this statement within the routing instance and specify the multicast group address for a particular VPN. Only one vpn-group-address statement can be configured per VPN, and this address should be unique on a per-VPN basis. To review how the VPN group address is used within GRE packet headers, see Stage 2 in "Dual PIM Multicast VPNs: Draft Rosen" on page 569.

Keep in mind that each PE router contains two entries of PIM: one for the master instance of PIM that connects through the service provider network and a second for the routing instance that connects to the CE router. The RP listed within the routing instance is the VPN C-RP, whereas the RP in the master PIM instance is an SP-RP. The following sample configuration shows a PE router with PIM enabled for sparse-dense mode in the VPN instance.

```
[edit]
routing-instances {
   instance-name {
      .....
```

```
        protocols {
            .....
            pim {
                vpn-group-address group-address;
                rp {
                    static {
                        address ip-address;
                    }
                }
                interface interface-name {
                    mode sparse-dense;
                    version 2;
                }
                interface lo0.1 {
                    mode sparse-dense;
                    version 2;
                }
            }
        }
    }
}
```

NOTE: You cannot configure PIM within a nonforwarding instance. If you try to do so, the router displays a commit check error and does not complete the configuration commit process.

NOTE: In JUNOS Release 5.5 and later, you can configure PIM dense mode with the dense statement at the [edit routing-instances pim mode] hierarchy level. Sparse mode is available at this same hierarchy level in JUNOS Release 5.3 and later.

Option: Configuring PIM Sparse Mode Graceful Restart for a Layer 3 VPN

Graceful restart permits a routing platform to continue forwarding multicast traffic to neighbors while the routing protocol process restarts. To enable graceful restart for PIM sparse mode in a Layer 3 VPN, include the graceful-restart statement at both the [edit routing-options] and [edit routing-instances *instance-name* routing-options] hierarchy levels. To disable graceful restart in a Layer 3 VPN, include the disable statement at the [edit routing-instances *instance-name* protocols pim graceful-restart] hierarchy level.

```
[edit]
routing-options {
    graceful-restart {
        disable;
        restart-duration seconds;
    }
}
routing-instances {
    instance-name {
        .....
```

```
        protocols {
            pim {
                graceful-restart {
                    disable;
                    restart-duration seconds;
                }
            }
        }
        routing-options {
            graceful-restart {
                disable;
                restart-duration seconds;
            }
        }
    }
}
```

For more information about PIM sparse mode graceful restart in a Layer 3 VPN, see the *JUNOS High Availability Configuration Guide* or the *JUNOS Multicast Protocols Configuration Guide*.

Option: Configuring Multicast Distribution Trees for Data

By using date multicast distribution trees (MDTs) in a Layer 3 VPN, you can prevent multicast packets from being flooded unnecessarily to specified provider edge (PE) routers within a VPN group. This option is primarily useful for PE routers in your Layer 3 VPN multicast network that have no receivers for the multicast traffic from a particular source.

When a PE router directly connected to the multicast source receives Layer 3 VPN multicast traffic exceeding a configured threshold, a new data MDT tunnel is established between the PE router connected to the site where the multicast source is and its remote PE router neighbors. Neighbors that do not have receivers for the multicast traffic ignore the new tunnel. Conversely, neighbors that do have receivers for the multicast traffic link to the data MDT tunnel, which is created when the site exceeds a traffic rate threshold. If the multicast traffic level drops back below the threshold, the data MDT is torn down automatically and traffic flows back across the default Layer 3 VPN PIM tunnel.

To specify when the PE router directly connected to the multicast source should create a new data MDT, you must configure the maximum threshold value by including the rate statement at the [edit routing-instances *instance-name* protocols pim mdt threshold group *group-address* source *source-address*] hierarchy level. The data rate is specified in kilobits per second (Kbps). To specify the maximum number of data MDTs that can be created for a single routing instance, include the tunnel-limit statement at the [edit routing-instances *instance-name* protocols pim mdt] hierarchy level. To specify the multicast group IP address range used when a new data MDT needs to be initiated on the PE router, include the group-range statement at the [edit routing-instances *instance-name* protocols pim mdt] hierarchy level.

```
[edit routing-instances instance-name  protocols pim]
mdt {
    group-range multicast-prefix;
```

```
    threshold {
      group group-address {
        source source-address {
          rate threshold-rate;
        }
      }
    }
    tunnel-limit limit;
  }
```

NOTE: Because MDTs applies to VPNs and VRF routing instances, you cannot configure MDT statements in the master routing instance. If you configure MDTs in the master routing instance, the configuration commit operation will fail.

For more information about MDT, see the *JUNOS Multicast Protocols Configuration Guide*.

Option: Configuring MSDP Within a Layer 3 VPN

MSDP, defined in RFC 3618, allows a PIM-enabled network to connect multicast routing domains. It typically runs on the same router as a PIM sparse-mode rendezvous point (RP). Each MSDP router establishes adjacencies with internal and external MSDP peers similar to adjacency establishment for BGP peers. MSDP peer routers inform each other about active sources within the domain. When the peers detect active sources, they send explicit Join messages to the active source.

You can configure MSDP in the master instance of a routing platform, or in the following types of routing instances:

- Forwarding
- No forwarding
- Virtual router
- VPLS
- VRF

To configure MSDP in a Layer 3 VPN, include the msdp statement at the [edit routing-instances *instance-name* protocols] hierarchy level and specify local and peer addresses. You must also configure PIM sparse mode in the routing instance and specify a rendezvous point.

```
[edit routing-instances instance-name protocols]
pim {
  rp {
    local {
      address ip-address;
    }
  }
  interface interface-name;
}
```

```
msdp {
   local-address local-ip-address;
   peer peer-ip-address;
}
```

To view information about the operation of MSDP within a Layer 3 VPN instance, issue the show msdp instance, show msdp statistics instance, show msdp source instance, and show msdp source-active instance commands. For more information about MSDP, see the *JUNOS Multicast Protocols Configuration Guide*.

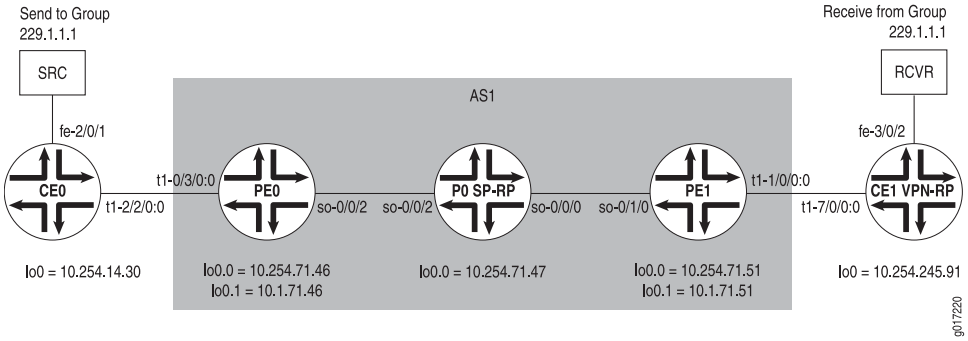
Draft-Rosen Multicast VPNs Examples

This section contains the following configuration examples and commands you can issue to verify your draft-rosen IPv4 multicast configuration:

- Example: Basic IPv4 Multicast over a Layer 3 VPN Configuration on page 616
- Example: IPv4 Multicast with Interprovider VPNs Configuration on page 629

Example: Basic IPv4 Multicast over a Layer 3 VPN Configuration

Figure 63: Basic IPv4 Multicast over a Layer 3 VPN Topology Diagram



In Figure 63 on page 616, the multicast source sends to group 229.1.1.1, and the receiver listens to the same group address. The VPN C-RP is located at Router CE1, whereas the SP-RP is located at Router P0. The routing instances are named VPN-A on both routers PE0 and PE1.

Router CE0

```
[edit]
protocols {
   pim {
      rp {
         dense-groups {
            229.0.0.0/8;
         }
         static {
            address 10.254.245.91;
         }
      }
      interface all {
         mode sparse-dense;
```

```
        version 2;
      }
      interface fxp0.0 {
        disable;
      }
    }
  }
```

In this example, the **interface all** statement is configured. If the topology requires only a few interfaces to be configured for PIM, then loopback interface **lo0** must also be one of the configured interfaces.

Router PE0

```
[edit]
protocols {
  pim {
    rp {
      static {
        address 10.254.71.47;
      }
    }
    interface all {
      mode sparse;
      version 2;
    }
    interface fxp0.0 {
      disable;
    }
  }
}
```

Router PE0 also requires a standard VPN configuration, along with the PIM instance configuration. The **vpn-group-address** command is the only new PIM statement with PIM used exclusively with a routing instance multicast configuration.

```
[edit]
routing-instances {
  VPN-A {
    instance-type vrf;
    interface t1-0/3/0:0.0;
    interface lo0.1
    route-distinguisher 10.254.71.46:100;
    vrf-import VPNA-import;
    vrf-export VPNA-export;
    protocols {
      ospf {
        export bgp-to-ospf;
        area 0.0.0.0 {
          interface t1-0/3/0:0.0;
          interface lo0.1;
        }
      }
      pim {
        dense-groups {
          229.0.0.0/8;
        }
        vpn-group-address 239.1.1.1;
```

```
            rp {
               static {
                  address 10.254.245.91;
               }
            }
            interface t1-0/3/0:0.0 {
               mode sparse-dense;
               version 2;
            }
            interface lo0.1 {
               mode sparse-dense;
               version 2;
            }
         }
      }
   }
}
```

Router P0

```
[edit]
protocols {
   pim {
      rp {
         local {
            address 10.254.71.47;
         }
      }
      interface all {
         mode sparse;
         version 2;
      }
      interface fxp0.0 {
         disable;
      }
   }
}
```

Again, if the configuration calls for specific interfaces to be configured for PIM, loopback interface lo0 must be included as one of the interfaces running PIM.

Router PE1

```
[edit]
protocols {
   pim {
      rp {
         static {
            address 10.254.71.47;
         }
      }
      interface all {
         mode sparse;
         version 2;
      }
      interface fxp0.0 {
         disable;
      }
   }
```

```
}
routing-instances {
   VPN-A {
      instance-type vrf;
      interface t1-1/0/0:0.0;
      interface lo0.1;
      route-distinguisher 10.254.71.51:100;
      vrf-import VPNA-import;
      vrf-export VPNA-export;
      protocols {
         ospf {
            export bgp-to-ospf;
            area 0.0.0.0 {
               interface t1-1/0/0:0.0;
               interface lo0.1;
            }
         }
         pim {
            dense-groups {
               229.0.0.0/8;
            }
            vpn-group-address 239.1.1.1;
            rp {
               static {
                  address 10.254.245.91;
               }
            }
            interface t1-1/0/0:0.0 {
               mode sparse-dense;
               version 2;
            }
            interface lo0.1 {
               mode sparse-dense;
               version 2;
            }
         }
      }
   }
}
```

Router CE1

```
[edit]
protocols {
   pim {
      dense-groups {
         229.0.0.0/8;
      }
      rp {
         local {
            address 10.254.245.91;
         }
      }
      interface all {
         mode sparse-dense;
         version 2;
      }
```

```
        interface fxp0.0 {
            disable;
        }
    }
}
```

Verifying Your Work

To verify correct operation of basic IPv4 multicast over a Layer 3 VPN, use the following commands:

- show pim
- show pim rps
- show pim rps instance *instance-name*
- show pim join
- show pim join extensive
- show pim join extensive instance *instance-name*
- show multicast route extensive
- show multicast next-hops
- show interfaces mt-*fpc/pic/port* extensive

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

- RP Information on page 620
- PIM Information Prior to Multicast Transmission on page 621
- Successful PIM Join Verification on page 622

RP Information

You can view PIM information for the master instance with the show pim command. You can see information on the PIM routing instance with the show pim (rps | join extensive) instance *instance-name* command. Output verifying the SP-RP (10.254.71.47) as well as the VPN C-RP (10.254.245.91) follows.

```
user@PE0> show pim rps
Instance: PIM.master
Family: INET
RP address      Type      Holdtime Timeout Active groups Group prefixes
10.254.71.47    static           0    None             1 224.0.0.0/4
Family: INET6
RP address      Type      Holdtime Timeout Active groups Group prefixes
user@PE0>  show pim rps instance VPN-A
Instance: PIM.VPN-A
Family: INET
RP address      Type      Holdtime Timeout Active groups Group prefixes
10.254.245.91   static           0    None             0 224.0.0.0/4
```

```
Family: INET6
RP address      Type      Holdtime Timeout Active groups Group prefixes
```

PIM Information Prior to Multicast Transmission

With the configuration properly set, the backbone PIM sessions should be established before any traffic is forwarded. In the output below, the routers were configured, but the traffic source was not transmitting and the receiver was not requesting to be part of a group. Notice that there is no PIM join information for the routing instances yet.

Router PE0

```
user@PE0> show pim join extensive
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.47
    Flags: sparse,rptree,wildcard
    Upstream interface: so-0/0/2.0
    Upstream State: Join to RP
    Downstream Neighbors:
        Interface: mt-1/1/0.32769
            0.0.0.0 State: Join   Flags: SRW  Timeout: Infinity
Group: 239.1.1.1
    Source: 10.254.71.46
    Flags: sparse
    Upstream interface: local
    Upstream State: Local Source, Prune to RP
    Keepalive timeout: 166
    Downstream Neighbors:
        Interface: so-0/0/2.0
            192.168.296.70 State: Join   Flags: S    Timeout: 204
Group: 239.1.1.1
    Source: 10.254.71.51
    Flags: sparse,spt-pending
    Upstream interface: so-0/0/2.0
    Upstream State: Join to Source
    Keepalive timeout: 166
    Downstream Neighbors:
        Interface: mt-1/1/0.32769
            0.0.0.0 State: Join   Flags: S    Timeout: Infinity
user@PE0>  show pim join extensive instance VPN-A
Instance: PIM.VPN-A Family: INET
```

Router P0

```
user@P0> show pim join extensive
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.47
    Flags: sparse,rptree,wildcard
    Upstream interface: local
    Upstream State: Local RP
    Downstream Neighbors:
        Interface: so-0/0/0.0
            192.168.296.34 State: Join   Flags: SRW  Timeout: 186
        Interface: so-0/0/2.0
            192.168.296.69 State: Join   Flags: SRW  Timeout: 198
Group: 239.1.1.1
    Source: 10.254.71.46
```

```
    Flags: sparse,spt
    Upstream interface: so-0/0/2.0
    Upstream State: Local RP, Join to Source
    Keepalive timeout: 170
    Downstream Neighbors:
        Interface: so-0/0/0.0
            192.168.296.34 State: Join   Flags: S    Timeout: 186
        Interface: so-0/0/2.0             (pruned)
            192.168.296.69 State: Prune  Flags: SR   Timeout: 198
Group: 239.1.1.1
    Source: 10.254.71.51
    Flags: sparse,spt
    Upstream interface: so-0/0/0.0
    Upstream State: Local RP, Join to Source
    Keepalive timeout: 170
    Downstream Neighbors:
        Interface: so-0/0/0.0             (pruned)
            192.168.296.34 State: Prune  Flags: SR   Timeout: 186
        Interface: so-0/0/2.0
            192.168.296.69 State: Join   Flags: S    Timeout: 198
```

Router PE1

```
user@PE1> show pim join extensive
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.47
    Flags: sparse,rptree,wildcard
    Upstream interface: so-0/1/0.0
    Upstream State: Join to RP
    Downstream Neighbors:
        Interface: mt-1/1/0.32769
            0.0.0.0 State: Join   Flags: SRW  Timeout: Infinity
Group: 239.1.1.1
    Source: 10.254.71.46
    Flags: sparse,spt-pending
    Upstream interface: so-0/1/0.0
    Upstream State: Join to Source
    Keepalive timeout: 180
    Downstream Neighbors:
        Interface: mt-1/1/0.32769
            0.0.0.0 State: Join   Flags: S    Timeout: Infinity
Group: 239.1.1.1
    Source: 10.254.71.51
    Flags: sparse
    Upstream interface: local
    Upstream State: Local Source, Prune to RP
    Keepalive timeout: 180
    Downstream Neighbors:
        Interface: so-0/1/0.0
            192.168.296.33 State: Join   Flags: S    Timeout: 168
```

Successful PIM Join Verification

In the remaining output for this example, the show pim join output shows group participation. Also displayed is the output from the show multicast route extensive and show multicast next-hop commands. The join output for PIM within a VPN will reference the group 229.1.1.1, while the service provider side of the network will reference the join information for group 239.1.1.1 (which is the VPN group ID). In

the show multicast route extensive output, you can view the group, sender, and upstream interface toward the sender.

Router CE0

```
user@CE0> show pim join
Instance: PIM.master Family: INET
Group: 229.1.1.1
    Source: 192.168.295.34
    Flags: dense
    Upstream interface: fe-2/0/1.0
Instance: PIM.master Family: INET6

user@CE0> show multicast route extensive
Family: INET
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout
229.1.1.1       192.168.295.34 /32 A   F  120   8010       0          360
    Upstream interface: fe-2/0/1.0
    Session name: Unknown
    Forwarding rate: 1 kbps (10 pps)
Family: INET6
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout

user@CE0> show multicast next-hops
Family: INET
ID      Refcount  KRefcount Downstream interface
120            2          1 t1-2/2/0:0.0
```

Router PE0

```
user@PE0> show pim join extensive
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.47
    Flags: sparse,rptree,wildcard
    Upstream interface: so-0/0/2.0
    Upstream State: Join to RP
    Downstream Neighbors:
        Interface: mt-1/1/0.32769
            10.1.71.46 State: Join   Flags: SRW  Timeout: Infinity
Group: 239.1.1.1
    Source: 10.254.71.46
    Flags: sparse
    Upstream interface: local
    Upstream State: Local Source, Prune to RP
    Keepalive timeout: 188
    Downstream Neighbors:
        Interface: so-0/0/2.0
            192.168.296.70 State: Join   Flags: S    Timeout: 180
Instance: PIM.master Family: INET6

user@PE0> show interfaces mt-1/1/0 extensive
Physical interface: mt-1/1/0, Enabled, Physical link is Up
  Interface index: 37, SNMP ifIndex: 45, Generation: 36
  Type: Multicast-GRE, Link-level type: GRE, MTU: Unlimited, Speed: 800mbps
  Hold-times     : Up 0 ms, Down 0 ms
  Device flags   : Present Running
  Interface flags: SNMP-Traps
  Statistics last cleared: Never
  Traffic statistics:
   Input  bytes  :              2887970                    0 bps
   Output bytes  :                    0                    0 bps
```

```
  Input  packets:                31896                    0 pps
  Output packets:                    0                    0 pps
 Logical interface mt-1/1/0.32769 (Index 43) (SNMP ifIndex 0) (Generation 46)
   Flags: Point-To-Point SNMP-Traps
   IP-Header 239.1.1.1:10.254.71.46:47:df:64:0000000800000000
   Encapsulation: GRE-NULL
   Traffic statistics:
    Input  bytes  :                    0
    Output bytes  :                 2396
    Input  packets:                    0
    Output packets:                   34
   Local statistics:
    Input  bytes  :                    0
    Output bytes  :                 2396
    Input  packets:                    0
    Output packets:                   34
   Transit statistics:
    Input  bytes  :                    0                    0 bps
    Output bytes  :                    0                    0 bps
    Input  packets:                    0                    0 pps
    Output packets:                    0                    0 pps
   Protocol inet, MTU: 4446, Generation: 79, Route table: 3
     Flags: None
 Logical interface mt-1/1/0.49154 (Index 44) (SNMP ifIndex 0) (Generation 47)
   Flags: Point-To-Point SNMP-Traps Encapsulation: GRE-NULL
   Traffic statistics:
    Input  bytes  :                 1550
    Output bytes  :                    0
    Input  packets:                   33
    Output packets:                    0
   Local statistics:
    Input  bytes  :                 1550
    Output bytes  :                    0
    Input  packets:                   33
    Output packets:                    0
   Transit statistics:
    Input  bytes  :                    0                    0 bps
    Output bytes  :                    0                    0 bps
    Input  packets:                    0                    0 pps
    Output packets:                    0                    0 pps
   Protocol inet, MTU: Unlimited, Generation: 80, Route table: 3
     Flags: None

user@PE0> show pim join extensive instance VPN-A
Instance: PIM.VPN-A Family: INET
Group: 229.1.1.1
    Source: 192.168.295.34
    Flags: dense
    Upstream interface: t1-0/3/0:0.0
    Downstream interfaces:
        mt-1/1/0.32769
Instance: PIM.VPN-A Family: INET6

user@PE0> show pim join
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.47
    Flags: sparse,rptree,wildcard
    Upstream interface: so-0/0/2.0
Group: 239.1.1.1
```

```
    Source: 10.254.71.46
    Flags: sparse
    Upstream interface: local
Instance: PIM.master Family: INET6
user@PE0>  show pim join instance VPN-A
Instance: PIM.VPN-A Family: INET
Group: 229.1.1.1
    Source: 192.168.295.34
    Flags: dense
    Upstream interface: t1-0/3/0:0.0
Instance: PIM.VPN-A Family: INET6

user@PE0> show multicast route extensive
Family: INET
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout
239.1.1.1       10.254.71.46   /32 A   F  86    9174       0          360
    Upstream interface: local
    Session name: Administratively Scoped
    Forwarding rate: 1 kBps (10 pps)
239.1.1.1       10.254.71.51   /32 A   F  96    36         0          360
    Upstream interface: so-0/0/2.0
    Session name: Administratively Scoped
    Forwarding rate: 0 kBps (0 pps)
Family: INET6
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout

user@PE0> show multicast route extensive instance VPN-A
Family: INET
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout
229.1.1.1       192.168.295.34 /32 A   F  85    9408       0          360
    Upstream interface: t1-0/3/0:0.0
    Session name: Unknown
    Forwarding rate: 1 kBps (10 pps)
Family: INET6
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout

user@PE0>  show multicast next-hops
Family: INET
ID      Refcount  KRefcount Downstream interface
86             2          1 so-0/0/2.0
85             2          1 mt-1/1/0.32769
96             2          1 mt-1/1/0.49154
Family: INET6
```

Router P0

```
user@P0> show pim join
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.47
    Flags: sparse,rptree,wildcard
    Upstream interface: local
Group: 239.1.1.1
    Source: 10.254.71.46
    Flags: sparse,spt
    Upstream interface: so-0/0/2.0
Group: 239.1.1.1
    Source: 10.254.71.51
    Flags: sparse,spt
    Upstream interface: so-0/0/0.0
```

```
Instance: PIM.master Family: INET6

user@PO> show multicast route extensive
Family: INET
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout
239.1.1.1       10.254.71.46   /32 A   F  127   9906       195        360
    Upstream interface: so-0/0/2.0
    Session name: Administratively Scoped
    Forwarding rate: 1 kBps (10 pps)
239.1.1.1       10.254.71.51   /32 A   F  126   135        23         359
    Upstream interface: so-0/0/0.0
    Session name: Administratively Scoped
    Forwarding rate: 0 kBps (0 pps)
Family: INET6
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout

user@PO> show multicast next-hops
Family: INET
ID      Refcount  KRefcount Downstream interface
127            2          1 so-0/0/0.0
126            2          1 so-0/0/2.0
Family: INET6
```

Router PE1

```
user@PE1> show pim join extensive
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.47
    Flags: sparse,rptree,wildcard
    Upstream interface: so-0/1/0.0
    Upstream State: Join to RP
    Downstream Neighbors:
        Interface: mt-1/1/0.32769
            10.1.71.51 State: Join   Flags: SRW  Timeout: Infinity
Group: 239.1.1.1
    Source: 10.254.71.46
    Flags: sparse,spt-pending
    Upstream interface: so-0/1/0.0
    Upstream State: Join to Source
    Keepalive timeout: 199
    Downstream Neighbors:
        Interface: mt-1/1/0.32769
            10.1.71.51 State: Join   Flags: S    Timeout: Infinity
Group: 239.1.1.1
    Source: 10.254.71.51
    Flags: sparse
    Upstream interface: local
    Upstream State: Local Source, Prune to RP
    Keepalive timeout: 79
    Downstream Neighbors:
        Interface: so-0/1/0.0
            192.168.296.33 State: Join   Flags: S    Timeout: 174
        Interface: register to RP 10.254.71.47 on pe-1/1/0.32769
Instance: PIM.master Family: INET6

user@PE1>  show pim join extensive instance VPN-A
Instance: PIM.VPN-A Family: INET
Group: 229.1.1.1
    Source: 192.168.295.34
    Flags: dense
```

```
    Upstream interface: mt-1/1/0.32769
    Downstream interfaces:
        t1-1/0/0:0.0
Instance: PIM.VPN-A Family: INET6

user@PE1> show interfaces mt-1/1/0 extensive
Physical interface: mt-1/1/0, Enabled, Physical link is Up
  Interface index: 38, SNMP ifIndex: 45, Generation: 37
  Type: Multicast-GRE, Link-level type: GRE, MTU: Unlimited, Speed: 800mbps
  Hold-times     : Up 0 ms, Down 0 ms
  Device flags   : Present Running
  Interface flags: SNMP-Traps
  Statistics last cleared: Never
  Traffic statistics:
   Input  bytes  :              2265256                 7568 bps
   Output bytes  :                    0                    0 bps
   Input  packets:                24981                   10 pps
   Output packets:                    0                    0 pps
  Logical interface mt-1/1/0.32769 (Index 45) (SNMP ifIndex 0) (Generation 46)
    Flags: Point-To-Point SNMP-Traps
    IP-Header 239.1.1.1:10.254.71.51:47:df:64:0000000800000000
    Encapsulation: GRE-NULL
    Traffic statistics:
     Input  bytes  :                    0
     Output bytes  :                10934
     Input  packets:                    0
     Output packets:                  153
    Local statistics:
     Input  bytes  :                    0
     Output bytes  :                10934
     Input  packets:                    0
     Output packets:                  153
    Transit statistics:
     Input  bytes  :                    0                    0 bps
     Output bytes  :                    0                    0 bps
     Input  packets:                    0                    0 pps
     Output packets:                    0                    0 pps
    Protocol inet, MTU: 4418, Generation: 77, Route table: 1
      Flags: None
  Logical interface mt-1/1/0.49154 (Index 46) (SNMP ifIndex 0) (Generation 47)
    Flags: Point-To-Point SNMP-Traps Encapsulation: GRE-NULL
    Traffic statistics:
     Input  bytes  :              1820512
     Output bytes  :                    0
     Input  packets:                19848
     Output packets:                    0
    Local statistics:
     Input  bytes  :                 5536
     Output bytes  :                    0
     Input  packets:                  120
     Output packets:                    0
    Transit statistics:
     Input  bytes  :              1814976                 7568 bps
     Output bytes  :                    0                    0 bps
     Input  packets:                19728                   10 pps
     Output packets:                    0                    0 pps
    Protocol inet, MTU: Unlimited, Generation: 78, Route table: 1
      Flags: None

user@PE1> show multicast route extensive
```

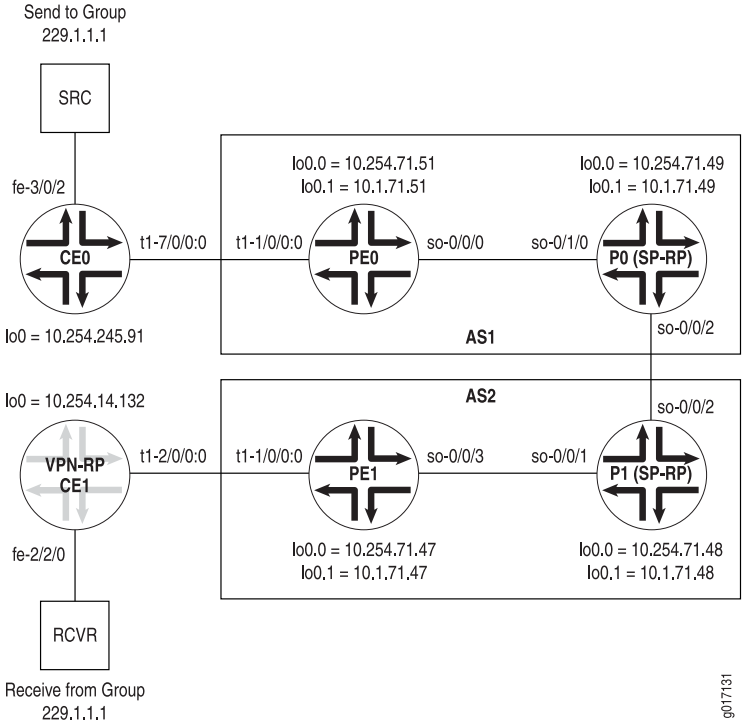
```
Family: INET
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout
239.1.1.1       10.254.71.46   /32 A   F  76    11014      0          360
    Upstream interface: so-0/1/0.0
    Session name: Administratively Scoped
    Forwarding rate: 1 kBps (10 pps)
239.1.1.1       10.254.71.51   /32 A   F  103   1          0          360
    Upstream interface: local
    Session name: Administratively Scoped
    Forwarding rate: 0 kBps (0 pps)
Family: INET6
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout

user@PE1>  show multicast route extensive instance VPN-A
Family: INET
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout
229.1.1.1       192.168.295.34 /32 A   F  99    10976      4          360
    Upstream interface: mt-1/1/0.49154
    Session name: Unknown
    Forwarding rate: 1 kBps (10 pps)
Family: INET6
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout

user@PE1>  show multicast next-hops
Family: INET
ID      Refcount  KRefcount Downstream interface
75             2          1 so-0/1/0.0
99             2          1 t1-1/0/0:0.0
76             2          1 mt-1/1/0.49154
Family: INET6
```

Router CE1

```
user@CE1> show pim join
Instance: PIM.master Family: INET
Group: 229.1.1.1
    Source: 192.168.295.34
    Flags: dense
    Upstream interface: t1-7/0/0:0.0
Instance: PIM.master Family: INET6

user@CE1> show multicast route extensive
          2          1 fe-3/0/2.0

Family: INET
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout
229.1.1.1       192.168.295.34 /32 A   F  120   8010       0          360
    Upstream interface: t1-7/0/0:0.0
    Session name: Unknown
    Forwarding rate: 1 kBps (10 pps)
Family: INET6
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout

user@CE1> show multicast next-hops
Family: INET
ID      Refcount  KRefcount Downstream interface
120
```

Example: IPv4 Multicast with Interprovider VPNs Configuration

Figure 64: IPv4 Multicast with Interprovider VPNs Topology Diagram



Interprovider VPNs are also mentioned in RFC 4364. An example is shown in Figure 64 on page 629. The topology is slightly different; the main difference is the addition of MSDP between the two provider core transit (P) routers. In this limited topology, each P router is an SP-RP for the local autonomous system (AS), and Router CE1 is the VPN C-RP. VPN-A is the name of the routing instance on routers PE0 and PE1.

Router CE0

```
[edit]
protocols {
   pim {
      dense-groups {
         229.0.0.0/8;
      }
      rp {
         static {
            address 10.254.14.132;
         }
      }
      interface all {
         mode sparse-dense;
         version 2;
      }
      interface fxp0.0 {
         disable;
      }
```

```
    }
}
```

Router PE0

```
[edit]
protocols {
    pim {
        rp {
            static {
                address 10.254.71.49;
            }
        }
        interface all {
            mode sparse;
            version 2;
        }
        interface fxp0.0 {
            disable;
        }
    }
}
routing-instances {
    VPN-A {
        protocols {
            pim {
                dense-groups {
                    229.0.0.0/8;
                }
                vpn-group-address 239.1.1.1;
                rp {
                    static {
                        address 10.254.14.132;
                    }
                }
                interface t1-1/0/0:0.0 {
                    mode sparse-dense;
                    version 2;
                }
                interface lo0.1 {
                    mode sparse-dense;
                    version 2;
                }
            }
        }
    }
}
```

Router P0

```
[edit]
protocols {
    ...
    msdp {
        peer 10.254.71.48 {
            local-address 10.254.71.49;
        }
    }
    ...
```

```
    pim {
        rp {
            local {
                address 10.254.71.49;
            }
        }
        interface all {
            mode sparse;
            version 2;
        }
        interface fxp0.0 {
            disable;
        }
    }
}
```

Router P1

```
[edit]
protocols {
    ...
    msdp {
        peer 10.254.71.49 {
            local-address 10.254.71.48;
        }
    }
    ...
    pim {
        rp {
            local {
                address 10.254.71.48;
            }
        }
        interface all {
            mode sparse;
            version 2;
        }
        interface fxp0.0 {
            disable;
        }
    }
}
```

Router PE1

```
[edit]
protocols {
    pim {
        rp {
            static {
                address 10.254.71.48;
            }
        }
        interface all {
            mode sparse;
            version 2;
        }
        interface fxp0.0 {
            disable;
```

```
        }
      }
    }
    routing-instances {
      VPN-A {
        protocols {
          pim {
            dense-groups {
              229.0.0.0/8;
            }
            vpn-group-address 239.1.1.1;
            rp {
              static {
                address 10.254.14.132;
              }
            }
            interface t1-1/0/0:0.0 {
              mode sparse-dense;
              version 2;
            }
            interface lo0.1 {
              mode sparse-dense;
              version 2;
            }
          }
        }
      }
    }
```

Router CE1

```
[edit]
protocols {
  pim {
    dense-groups {
      229.0.0.0/8;
    }
    rp {
      local {
        address 10.254.14.132;
      }
    }
    interface all {
      mode sparse-dense;
      version 2;
    }
    interface fxp0.0 {
      disable;
    }
  }
}
```

Verifying Your Work

The show commands used to verify proper functionality of multicast in an interprovider environment are the same ones used with the first Layer 3 VPN multicast example (see "Verifying Your Work" on page 620).

The following output provides details for RP and the PIM join information:

Router CE0 Status

```
user@CE0> show pim rps extensive
Instance: PIM.master
Family: INET
RP: 10.254.14.132
Learned via: static configuration
Time Active: 00:21:35
Holdtime: 0
Device Index: 119
Subunit: 32769
Interface: pe-6/0/0.32769
Group Ranges:
        224.0.0.0/4
Active groups using RP:
Register State for RP:
Group           Source          FirstHop        RP Address      State    Timeout
Family: INET6

user@CE0> show pim join extensive
Instance: PIM.master Family: INET
Group: 229.1.1.1
    Source: 192.168.295.38
    Flags: dense
    Upstream interface: fe-3/0/2.0
    Downstream interfaces:
        t1-7/0/0:0.0
Instance: PIM.master Family: INET6
```

Router PE0 Status

```
user@PE0> show pim rps extensive
Instance: PIM.master
Family: INET
RP: 10.254.71.49
Learned via: static configuration
```

```
Time Active: 00:22:07
Holdtime: 0
Device Index: 34
Subunit: 32769
Interface: pe-1/1/0.32769
Group Ranges:
        224.0.0.0/4
Active groups using RP:
        239.1.1.1
        total 1 groups active
Register State for RP:
Group           Source          FirstHop        RP Address      State    Timeout
239.1.1.1       10.254.71.51    10.254.71.51    10.254.71.49    Suppress      20
Family: INET6

user@PE0> show pim rps extensive instance VPN-A
Instance: PIM.VPN-A
Family: INET
RP: 10.254.14.132
Learned via: static configuration
Time Active: 00:22:22
Holdtime: 0
Device Index: 34
Subunit: 32771
Interface: pe-1/1/0.32771
Group Ranges:
        224.0.0.0/4
Active groups using RP:
Register State for RP:
Group           Source          FirstHop        RP Address      State    Timeout
Family: INET6

user@PE0>  show pim join extensive
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.49
    Flags: sparse,rptree,wildcard
    Upstream interface: so-0/0/0.0
    Upstream State: Join to RP
    Downstream Neighbors:
        Interface: mt-1/1/0.32769
            0.0.0.0 State: Join   Flags: SRW  Timeout: Infinity
Group: 239.1.1.1
    Source: 10.254.71.47
    Flags: sparse,spt-pending
    Upstream interface: so-0/0/0.0
    Upstream State: Join to Source
    Keepalive timeout: 198
    Downstream Neighbors:
        Interface: mt-1/1/0.32769
            0.0.0.0 State: Join   Flags: S    Timeout: Infinity
Group: 239.1.1.1
    Source: 10.254.71.51
    Flags: sparse
    Upstream interface: local
    Upstream State: Local Source, Prune to RP
    Keepalive timeout: 198
    Downstream Neighbors:
        Interface: so-0/0/0.0
            192.168.296.42 State: Join   Flags: S    Timeout: 176
```

```
Instance: PIM.master Family: INET6

user@PE0> show pim join extensive instance VPN-A
Instance: PIM.VPN-A Family: INET
Group: 229.1.1.1
    Source: 192.168.295.38
    Flags: dense
    Upstream interface: t1-1/0/0:0.0
    Downstream interfaces:
        mt-1/1/0.32769
Instance: PIM.VPN-A Family: INET6
```

Router P0 Status

```
user@P0> show pim rps extensive
Instance: PIM.master
Family: INET
RP: 10.254.71.49
Learned via: static configuration
Time Active: 00:30:43
Holdtime: 0
Device Index: 33
Subunit: 32768
Interface: pd-1/1/0.32768
Group Ranges:
        224.0.0.0/4
Active groups using RP:
        239.1.1.1
        total 1 groups active
Register State for RP:
Group           Source          FirstHop        RP Address      State    Timeout
239.1.1.1       10.254.71.51    10.254.71.51    10.254.71.49    Receive
Family: INET6

user@P0> show pim join extensive
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.49
    Flags: sparse,rptree,wildcard
    Upstream interface: local
    Upstream State: Local RP
    Downstream Neighbors:
        Interface: so-0/1/0.0
            192.168.296.41 State: Join   Flags: SRW  Timeout: 184
Group: 239.1.1.1
    Source: 10.254.71.47
    Flags: sparse,spt-pending
    Upstream interface: so-0/0/2.0
    Upstream State: Local RP, Join to Source
    Keepalive timeout: 207
    Downstream Neighbors:
        Interface: so-0/1/0.0
            192.168.296.41 State: Join   Flags: S    Timeout: 184
Group: 239.1.1.1
    Source: 10.254.71.51
    Flags: sparse,spt
    Upstream interface: so-0/1/0.0
```

```
    Upstream State: Local RP, Join to Source
    Keepalive timeout: 207
    Downstream Neighbors:
        Interface: so-0/0/2.0
            192.168.296.73 State: Join   Flags: S    Timeout: 186
        Interface: so-0/1/0.0             (pruned)
            192.168.296.41 State: Prune  Flags: SR   Timeout: 184
Instance: PIM.master Family: INET6
```

Router P1 Status

```
user@P1> show pim rps extensive
Instance: PIM.master
Family: INET
RP: 10.254.71.48
Learned via: static configuration
Time Active: 06:26:56
Holdtime: 0
Device Index: 32
Subunit: 32768
Interface: pd-1/1/0.32768
Group Ranges:
        224.0.0.0/4
Active groups using RP:
        239.1.1.1
        total 1 groups active
Register State for RP:
Group           Source          FirstHop        RP Address      State    Timeout
239.1.1.1       10.254.71.47    10.254.71.47    10.254.71.48    Receive        0
Family: INET6

user@P1> show pim join extensive
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.48
    Flags: sparse,rptree,wildcard
    Upstream interface: local
    Upstream State: Local RP
    Downstream Neighbors:
        Interface: so-0/0/1.0
            192.168.296.50 State: Join   Flags: SRW  Timeout: 174
Group: 239.1.1.1
    Source: 10.254.71.47
    Flags: sparse,spt
    Upstream interface: so-0/0/1.0
    Upstream State: Local RP, Join to Source
    Keepalive timeout: 196
    Downstream Neighbors:
        Interface: so-0/0/1.0             (pruned)
            192.168.296.50 State: Prune  Flags: SR   Timeout: 174
        Interface: so-0/0/2.0
            192.168.296.74 State: Join   Flags: S    Timeout: 178
Group: 239.1.1.1
    Source: 10.254.71.51
    Flags: sparse,spt-pending
    Upstream interface: so-0/0/2.0
    Upstream State: Local RP, Join to Source
```

```
    Keepalive timeout: 196
    Downstream Neighbors:
        Interface: so-0/0/1.0
            192.168.296.50 State: Join   Flags: S    Timeout: 174
Instance: PIM.master Family: INET6
```

Router PE1 Status

```
user@PE1> show pim rps extensive
Instance: PIM.master
Family: INET
RP: 10.254.71.48
Learned via: static configuration
Time Active: 00:25:13
Holdtime: 0
Device Index: 34
Subunit: 32770
Interface: pe-1/1/0.32770
Group Ranges:
        224.0.0.0/4
Active groups using RP:
        239.1.1.1
        total 1 groups active
Register State for RP:
Group           Source          FirstHop        RP Address      State    Timeout
239.1.1.1       10.254.71.47    10.254.71.47    10.254.71.48    Suppress      42
Family: INET6

user@PE1> show pim rps extensive instance VPN-A
Instance: PIM.VPN-A
Family: INET
RP: 10.254.14.132
Learned via: static configuration
Time Active: 00:25:17
Holdtime: 0
Device Index: 34
Subunit: 32771
Interface: pe-1/1/0.32771
Group Ranges:
        224.0.0.0/4
Active groups using RP:
Register State for RP:
Group           Source          FirstHop        RP Address      State    Timeout
Family: INET6

user@PE1>  show pim join extensive
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.48
    Flags: sparse,rptree,wildcard
    Upstream interface: so-0/0/3.0
    Upstream State: Join to RP
    Downstream Neighbors:
        Interface: mt-1/1/0.32769
            0.0.0.0 State: Join   Flags: SRW  Timeout: Infinity
Group: 239.1.1.1
    Source: 10.254.71.47
```

```
    Flags: sparse
    Upstream interface: local
    Upstream State: Local Source, Prune to RP
    Keepalive timeout: 173
    Downstream Neighbors:
        Interface: so-0/0/3.0
            192.168.296.49 State: Join   Flags: S    Timeout: 199
Group: 239.1.1.1
    Source: 10.254.71.51
    Flags: sparse,spt-pending
    Upstream interface: so-0/0/3.0
    Upstream State: Join to Source
    Keepalive timeout: 173
    Downstream Neighbors:
        Interface: mt-1/1/0.32769
            0.0.0.0 State: Join   Flags: S    Timeout: Infinity
Instance: PIM.master Family: INET6

user@PE1>  show pim join extensive instance VPN-A
Instance: PIM.VPN-A Family: INET
Group: 229.1.1.1
    Source: 192.168.295.38
    Flags: dense
    Upstream interface: mt-1/1/0.32769
    Downstream interfaces:
        t1-1/0/0:0.0
Instance: PIM.VPN-A Family: INET6
```

Router CE1 Status

```
user@CE1> show pim rps extensive
Instance: PIM.master
Family: INET
RP: 10.254.14.132
Learned via: static configuration
Time Active: 00:28:22
Holdtime: 0
Device Index: 69
Subunit: 32768
Interface: pd-3/1/0.32768
Group Ranges:
        224.0.0.0/4
Active groups using RP:
Register State for RP:
Group           Source          FirstHop        RP Address      State    Timeout
Family: INET6

user@CE1>  show pim join extensive
Instance: PIM.master Family: INET
Group: 229.1.1.1
    Source: 192.168.295.38
    Flags: dense
    Upstream interface: t1-2/0/0:0.0
    Downstream interfaces:
        fe-2/2/0.0
Instance: PIM.master Family: INET6
```

For More Information

For additional information on multicast over Layer 3 VPNs, see the following resources:

- *JUNOS Multicast Protocols Configuration Guide*
- *JUNOS VPNs Configuration Guide*
- *RFC 2547, BGP/MPLS VPNs*
- *RFC 3618, Multicast Source Discovery Protocol (MSDP)*
- *RFC 4364, BGP/MPLS IP Virtual Private Networks (VPNs)*
- Internet draft draft-rosen-vpn-mcast-06.txt, *Multicast in MPLS/BGP VPNs* (expires April 2004)

Revision History

13 April 2009—9.5R1 Release. Roy Spencer.

15 January 2009—9.4R1 Release. Roy Spencer.

10 October 2008—9.3R1 Release. Roy Spencer.

10 July 2008—Added the -Step Multicast VPN Configuration Example. 9.2R1 Release. Roy Spencer.

10 April 2008—9.1R1 Release. Roy Spencer.

1 February 2008—9.0R1 Release. Fawn Damitio.

12 October 2007—Added point-to-multipoint TE provider tunnel support. 8.5R1 Release. Fawn Damitio.

29 June 2007—Added multiprotocol BGP multicast VPN: next-generation support. 8.4R1 Release. Fawn Damitio.

27 March 2007—8.3R1 Release. Fawn Damitio.

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

15 September 2006—Added RFC 4364, 8.1R1 Release. Richard Hendricks.

29 June 2006—8.0R1 Release. Richard Hendricks.

27 March 2006—7.6R1 Release. Richard Hendricks.

9 January 2006—7.5R1 Release. Richard Hendricks.

14 September 2005—7.4R1 Release. Richard Hendricks.

13 June 2005—7.3R1 Release. Richard Hendricks.

5 April 2005—Added support for MSDP in Layer 3 VPNs, 7.2R1 Release. Richard Hendricks.

2 February 2005—Added multicast distribution trees for data, 7.1R1 Release. Richard Hendricks.

6 October 2004—7.0R1 Release. Richard Hendricks.

6 July 2004—Added graceful restart information and mentioned that a PE router can now act as a VPN C-RP, 6.4R1 Release. Richard Hendricks.

5 April 2004—6.3R1 Release. Richard Hendricks.

22 December 2003—6.2R1 Release. Richard Hendricks.

22 September 2003—6.1R1 Release. Richard Hendricks.

30 June 2003—6.0R1 Release. Richard Hendricks.

2 April 2003—5.7R1 Release. Richard Hendricks.

27 December 2002—5.6R1 Release. Richard Hendricks.

30 September 2002—5.5R1 Release. Richard Hendricks.

27 August 2002—Reformatted the document in Feature Guide style. Richard Hendricks.

22 August 2002—Added PIM dense mode information. Bill Nowak.

8 February 2002—Initial 5.3 Quick Start Guide document. Bill Nowak.

Chapter 14

Translational Cross-Connect and Layer 2.5 VPNs

This feature guide covers these topics:

Overview

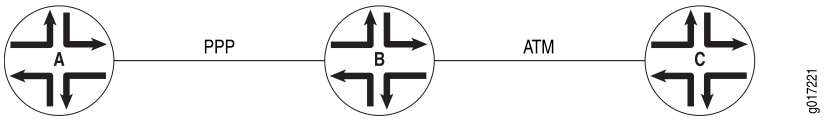
Translational cross-connect (TCC) is a switching concept that allows you to establish interconnections between a variety of Layer 2 protocols or circuits. It is similar to its predecessor, circuit cross-connect (CCC). However, while CCC requires the same Layer 2 encapsulations on both sides of a Juniper Networks router (such as PPP-to-PPP or Frame Relay-to-Frame Relay), TCC lets a network administrator connect different types of Layer 2 protocols interchangeably. With TCC, combinations such as PPP-to-ATM and Ethernet-to-Frame Relay cross-connections are possible. Also, TCC can be used to create Layer 2.5 VPNs and Layer 2.5 circuits.

The JUNOS software makes interworking between unlike protocols possible. The software strips off the Layer 2 header when frames enter the router and adds a different Layer 2 header before the frames exit the router. TCC supports these Layer 2 protocols:

- ATM
- Cisco HDLC
- Ethernet
- Extended VLAN
- Frame Relay
- PPP

In Figure 65 on page 642, the PPP header is stripped from frames arriving at Router B and an ATM header is added before the frames are sent to Router C. All Layer 2 negotiations are terminated at the interconnecting router (Router B). Examples include Link Control Protocol (LCP) and Network Control Protocol (NCP) for PPP, keepalives for Cisco HDLC, and Local Management Interface (LMI) for Frame Relay.

Figure 65: TCC Concept Example



TCC functionality is different from standard Layer 2 switching. TCC only swaps Layer 2 headers. No other processing, such as header checksums, time-to-live (TTL) decrementing, or protocol handling, is performed. Currently, TCC is supported in IPv4, ISO, and MPLS.

This guide shows you how to use the Layer 2 interworking nature of TCC for interface switching and for constructing Layer 2.5 virtual private networks (VPNs).

System Requirements

To implement TCC, your system must meet these minimum requirements:

- JUNOS Release 8.3 or later for support of IS-IS and MPLS traffic over Layer 2.5 VPNs.
- JUNOS Release 8.2 or later for support on M120 routers and MX-series routing platforms
- JUNOS Release 6.2 or later for CCC, TCC, and Layer 2.5 VPN support on M320 routers
- JUNOS Release 6.0 or later for CCC, TCC, and Layer 2.5 VPN support on T-series routing platforms
- JUNOS Release 5.6 or later for proxy Address Resolution Protocol (ARP) functionality in Ethernet TCC and extended VLAN TCC networks on M-series routers
- JUNOS Release 5.4 or later for Ethernet TCC and extended VLAN TCC interface encapsulation types on M-series routers
- JUNOS Release 5.2 or later for PPP TCC, Cisco HDLC TCC, ATM TCC, and Frame Relay TCC interface encapsulation types on M-series routers
- Ethernet-based PICs to support Ethernet TCC and extended VLAN TCC interface encapsulation types, with the following exceptions:
- All 8-port, 12-port, and 48-port Fast Ethernet PICs do not support TCC
- 4-port Gigabit Ethernet PICs do not support extended VLAN TCC
- Any combination of three Juniper Networks M-series or T-series routing platforms for TCC or five routing platforms for Layer 2.5 VPNs or Layer 2.5 circuits

Terms and Acronyms

C

circuit cross-connect (CCC) A method of exchanging frames between two router interfaces running the same Layer 2 protocol, such as ATM, Cisco HDLC, Ethernet, extended VLAN, Frame Relay, and PPP. For more information about CCC, see the *JUNOS Network Interfaces Configuration Guide*.

customer edge router (CE) Any router that connects a customer site to a PE router.

L

Layer 2.5 circuits A method of privately connecting sites across the Internet. Layer 2.5 circuits function like Layer 2 circuits, but connect two sites that use different Layer 2 protocols. For more information about Layer 2 circuits, see “Layer 2 Circuits” on page 513 or the *JUNOS VPNs Configuration Guide*.

Layer 2.5 VPNs A method of privately connecting sites across the Internet. Layer 2.5 VPNs function like Layer 2 VPNs, but connect two sites that use different Layer 2 protocols. For more information about Layer 2 VPNs, see the *JUNOS VPNs Configuration Guide*.

P

provider core router (P) Any router that lies between PE routers in the provider core network.

provider edge router (PE) Any router that connects customer edge (CE) routers to the provider core network.

T

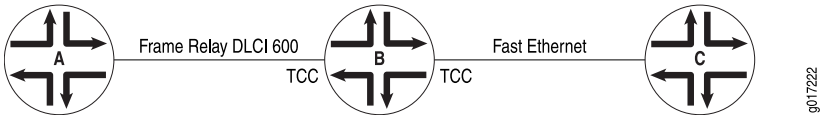
translational cross-connect (TCC) A Juniper Networks method of exchanging frames between two router interfaces running different Layer 2 protocols, such as ATM, Cisco HDLC, Ethernet, extended VLAN, Frame Relay, and PPP. For more information about TCC, see the *JUNOS Network Interfaces Configuration Guide*.

Configuring TCC Interface Switching

TCC joins disparate Layer 2 logical interfaces by means of a virtual Layer 2 switching technique.

Figure 66 on page 644 illustrates Layer 2 TCC switching. In this example, Router A is connected by Frame Relay to Router B and Router B is connected by Fast Ethernet to Router C. Router B is a Juniper Networks router. TCC allows you to configure Router B to act as a multiservice Layer 2 switch. To do this, you configure a connection from Router A to Router C that passes through Router B. This effectively establishes Router B as a Frame Relay-to-Fast Ethernet switch and allows the router to switch frames transparently between Router A and Router C without regard to the frames' contents or the Layer 3 protocols. Router B removes the Frame Relay header from frames arriving on data-link connection identifier (DLCI) 600 and adds an Ethernet header onto the frames before sending them to the Fast Ethernet link.

Figure 66: Layer 2 TCC Switching



You can configure Layer 2 TCC switching on ATM, Cisco HDLC, Ethernet, extended VLAN, Frame Relay, and PPP circuits. TCC enables unlike interfaces to be connected with a single cross-connect.

To configure Layer 2 TCC switching, perform the following tasks on the router that is acting as the switch (Router B in Figure 66 on page 644):

Defining the Encapsulation for Layer 2 TCC Switching

To begin implementation of Layer 2 TCC switching, configure TCC encapsulation on the desired interfaces of the router that is acting as the switch (Router B in Figure 66 on page 644).

NOTE: You cannot configure standard protocol families on TCC or CCC interfaces. Only the CCC family is allowed on CCC-encapsulated interfaces. Likewise, only the TCC family is allowed on TCC-encapsulated interfaces.

For ATM connections, specify the encapsulation type when configuring ATM virtual circuits (VCs) at the [edit interfaces *interface-name* unit *unit-number*] hierarchy level. For each VC, you configure whether it is a circuit or a regular logical interface. The default interface type is point-to-point.

```
[edit]
interfaces {
   at-fpc/pic/port {
      atm-options {
         vpi vpi-identifier  maximum-vcs maximum-vcs;
      }
      unit logical-unit-number {
         point-to-point; # This is the default interface type.
         encapsulation (atm-tcc-vc-mux | atm-tcc-snap);
         vci vpi-identifier.vci-identifier;
      }
   }
}
```

For Cisco HDLC and PPP circuits, specify the encapsulation in the encapsulation statement at the [edit interfaces *interface-name*] hierarchy level. This statement configures the entire physical device. Also, you must configure the logical interface unit 0 .

```
[edit]
interfaces {
   type-fpc/pic/port {
      encapsulation (cisco-hdlc-tcc | ppp-tcc);
```

```
        unit 0;
    }
}
```

You can specify the encapsulation for Frame Relay circuits at the [edit interfaces *interface-name*] hierarchy level and the [edit interfaces *interface-name* unit *unit-number*] hierarchy level. For TCC and CCC interfaces, the DLCI value must be configured in the range of 512 through 1022. The default interface type is point-to-point.

```
[edit]
interfaces {
    type-fpc/pic/port  {
        encapsulation frame-relay-tcc;
        unit logical-unit-number {
            point-to-point; # This is the default interface type.
            encapsulation frame-relay-tcc;
            dlci dlci-identifier;
        }
    }
}
```

Configuring Ethernet Encapsulation with Remote and Proxy ARP Addresses

For Ethernet TCC circuits, specify the encapsulation with the encapsulation ethernet-tcc statement. This statement configures the entire physical device.

To provide Address Resolution Protocol (ARP) functionality for an Ethernet-based neighbor, configure the remote statement at the [edit interfaces *interface-name* unit *unit-number* family tcc] hierarchy level and specify either the MAC address or IP address of the TCC router's Ethernet neighbor. To complete the setup of ARP, configure the proxy statement at the [edit interfaces *interface-name* unit *unit-number* family tcc] hierarchy level and specify the IP address of the TCC router's non-Ethernet neighbor.

```
[edit]
interfaces
EthernetType-fpc/pic/port {
    encapsulation ethernet-tcc;
    unit 0 {
        family tcc {
            remote { # Addresses associated with the Ethernet TCC neighbor.
                mac-address mac-address; # Select a MAC or IP address.
                inet-address inet-address;
            }
            proxy { # Addresses belonging to the non-Ethernet TCC neighbor.
                inet-address inet-address;
            }
        }
    }
}
```

Configuring Extended VLAN Encapsulation

with Remote and Proxy ARP Addresses

Specify the encapsulation for extended VLAN circuits with the encapsulation extended-vlan-tcc statement. This statement configures the entire physical device.

You must also enable VLAN tagging. Ethernet interfaces in VLAN mode can have multiple logical interfaces. For encapsulation type extended-vlan-tcc, all VLAN IDs from 0 through 4094 are valid, up to a maximum of 1024 VLANs.

To enable ARP functionality, configure the remote statement at the [edit interfaces *interface-name* unit *unit-number* family tcc] hierarchy level with either the MAC address or IP address of your Ethernet TCC neighbor. To complete the ARP setup, configure the proxy statement at the [edit interfaces *interface-name* unit *unit-number* family tcc] hierarchy level and specify the IP address of the non-Ethernet TCC neighbor.

```
[edit]
interfaces {
   EthernetType-fpc/pic/port {
      vlan-tagging;
      encapsulation extended-vlan-tcc;
      unit 0 {
         vlan-id 600;
         family tcc {
            remote { # Addresses associated with the Ethernet TCC neighbor.
               mac-address mac-address; # Select a MAC or IP address.
               inet-address inet-address;
            }
            proxy { # Addresses belonging to the non-Ethernet TCC neighbor.
               inet-address inet-address;
            }
         }
      }
   }
}
```

Option: Configuring Static ARP on the Ethernet Neighbor Instead of Proxy ARP

If you do not use the proxy statement in the Ethernet TCC and extended VLAN TCC encapsulation hierarchies shown earlier, you must use another method to allow ARP to continue to function. To retain the functionality of ARP for Ethernet networks, you must configure static ARP on the Ethernet neighbor. Use of static ARP assumes that you have already configured the remote statement on the TCC router (see "Configuring Ethernet Encapsulation with Remote and Proxy ARP Addresses" on page 646 and "Configuring Extended VLAN Encapsulation with Remote and Proxy ARP Addresses" on page 646).

You configure the arp statement on the Ethernet neighbor at the [edit interfaces *interface-number* unit *unit-number* family inet address *ip-address*] hierarchy level. Your static ARP statement must contain the IP address of the non-Ethernet neighbor on the opposite side of the TCC router and the Ethernet interface MAC address of the TCC router. This static ARP configuration enables return path ARP functionality and complements the remote statement previously set on the TCC router.

In Figure 68 on page 651, you would configure an ARP statement on the fe-0/0/0 interface of Router C. The ARP statement would contain the IP address for interface so-0/1/0.600 on Router A and the MAC address of the fe-1/0/0 interface of Router B.

Configure static ARP on an Ethernet neighbor at the [edit interfaces *interface-name* unit *unit-number* family inet address *ip-address*] hierarchy level.

```
[edit]
interfaces
EthernetType-fpc/pic/port {
   unit 0 {
      family inet {
         address ip-address { # The local IP address.
            arp ip-address mac mac-address; # IP address of the non-Ethernet
         }          # TCC neighbor and MAC address of the TCC
      }          # router's Ethernet interface.
   }
}
```

Defining the Connection for Layer 2 TCC Switching

The next step in configuring Layer 2 TCC switching is to define the connection between the two circuits. You configure this on the router acting as the TCC switch. When you specify the interface names, include the logical portion of the name, which corresponds to the logical unit number. The cross-connect is bidirectional, so packets received on the first interface are transmitted by the second interface, and those received on the second interface are transmitted by the first.

```
[edit]
protocols {
   connections {
      interface-switch connection-name {
         interface first-interface-name.unit-number;
         interface second-interface-name.unit-number;
      }
   }
}
```

Configuring MPLS

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

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

```
}
```

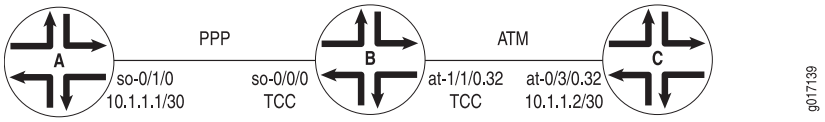
TCC Configuration Examples

This section contains configuration examples and commands you can issue to verify your Layer 2 TCC switching configuration:

- Example: PPP to ATM TCC Configuration on page 649
- Example: Frame Relay to Fast Ethernet TCC Configuration on page 651

Example: PPP to ATM TCC Configuration

Figure 67: TCC Interface Switching—PPP to ATM



In Figure 67 on page 649, Router A uses PPP to connect with Router B, while Router C connects with Router B through ATM. Router B acts as the Layer 2 virtual switch and transparently connects Router A to Router C.

On Router A, configure basic PPP encapsulation and any desired Layer 3 protocol families on the SONET/SDH interface.

Router A

```
[edit]
interfaces {
   so-0/1/0 {
      description "to Router B so-0/0/0";
      unit 0 {
         encapsulation ppp;
         family inet {
            address 10.1.1.1/30;
         }
      }
   }
}
```

Router B acts as the virtual Layer 2 switch. Here you configure the appropriate TCC encapsulations on the corresponding interfaces. In this case, encapsulation ppp-tcc is bound to physical interface so-0/0/0, and encapsulation atm-tcc-vc-mux is placed on VC 32 of interface at-1/1/0. Because the switching occurs at Layer 2, you cannot configure IP addresses or other Layer 3 family information on these interfaces.

You also need to configure MPLS and establish the cross-connect by adding the necessary interfaces to the interface-switch statement at the [edit protocols connections] hierarchy level.

Router B

```
[edit]
```

```
interfaces {
   so-0/0/0 {
      description "to Router A so-0/1/0";
      encapsulation ppp-tcc;
      unit 0 {
      }
      at-1/1/0 {
         description "to Router C at-0/3/0";
         atm-options {
            vpi 0 maximum-vcs 2000;
         }
         unit 32 {
            vci 32;
            encapsulation atm-tcc-vc-mux;
         }
      }
   }
   protocols {
      mpls {
         interface so-0/0/0.0;
         interface at-1/1/0.32;
      }
      connections {
         interface-switch PPP-to-ATM {
            interface so-0/0/0.0;
            interface at-1/1/0.32;
         }
      }
   }
}
```

On Router C, the encapsulation option used to connect to the TCC-encapsulated ATM interface on Router B is atm-vc-mux. Since this ATM connection is switched at Layer 2 to reach the PPP link, it is transparent to Layer 3 addressing. As a result, the IP address must be configured in the same address space as Router A's so-0/1/0 interface.

Router C

```
[edit]
interfaces {
   at-0/3/0 {
      description "to Router B at-1/1/0";
      atm-options {
         vpi 0 maximum-vcs 2000;
      }
      unit 32 {
         vci 32;
         encapsulation atm-vc-mux;
         family inet {
            address 10.1.1.2/30;
         }
      }
   }
}
```

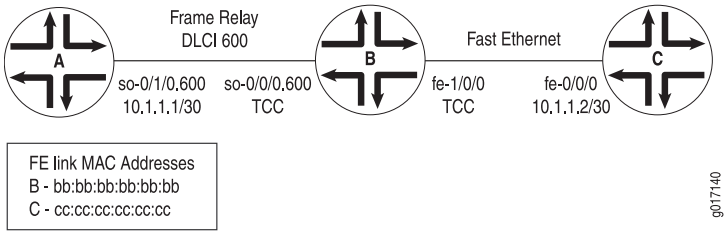
Verifying Your Work

To verify your TCC connection, use the show connections command on Router B:

```
user@router_b> show connections
CCC and TCC connections [Link Monitoring On]
  Legend for status (St)              Legend for connection types
  UN -- uninitialized                 if-sw:  interface switching
  NP -- not present                   rmt-if: remote interface switching
  WE -- wrong encapsulation           lsp-sw: LSP switching
  DS -- disabled
  Dn -- down                          Legend for circuit types
  -> -- only outbound conn is up      intf -- interface
  <- -- only inbound  conn is up      tlsp -- transmit LSP
  Up -- operational                   rlsp -- receive LSP
Connection/Circuit                  Type    St  Time last up     # Up trans
PPP-to-ATM                          if-sw   Up  Nov 30 08:57:53           1
  at-1/1/0.32                         intf  Up
  so-0/0/0.0                          intf  Up
```

Example: Frame Relay to Fast Ethernet TCC Configuration

Figure 68: TCC Interface Switching—Frame Relay to Fast Ethernet



In the configuration example in Figure 68 on page 651, Router A uses Frame Relay to connect with Router B, while Router C connects to Router B by using Fast Ethernet. Router B acts as the Layer 2 virtual switch and transparently connects Router A to Router C.

You must enable Frame Relay encapsulation on Router A at the physical interface level.

Router A

```
[edit]
interfaces {
   so-0/1/0 {
      description "to Router B so-0/0/0";
      encapsulation frame-relay;
      unit 600 {
         point-to-point;
         dlci 600;
         family inet {
            address 10.1.1.1/30;
         }
      }
   }
```

```
}
```

Router B acts as the virtual switch. Enable the appropriate TCC encapsulations on the corresponding interfaces. In this case, configure the encapsulation frame-relay-tcc option on the logical and physical interfaces of so-0/0/0.600. Next, add the ethernet-tcc encapsulation type to the physical interface of fe-1/0/0. To enable ARP, configure the remote MAC address or IP address of Router C's Fast Ethernet interface with the remote statement at the [edit interfaces *interface-name* unit 0 family tcc] hierarchy level. To enable proxy ARP, include the proxy statement at the [edit interfaces *interface-name* unit 0 family tcc] hierarchy level and specify the IP address of Router A.

After configuring the correct interface encapsulations, complete your cross-connect by adding both interfaces into your MPLS configuration. Include the same interfaces in the interface-switch statement at the [edit protocols connections] hierarchy level.

Router B

```
[edit]
interfaces {
   so-0/0/0 {
      description "to Router A so-0/1/0";
      dce;
      encapsulation frame-relay-tcc;
      unit 600 {
         point-to-point;
         encapsulation frame-relay-tcc;
         dlci 600;
      }
   }
   fe-1/0/0 {
      description "to Router C fe-0/0/0";
      encapsulation ethernet-tcc;
      unit 0 {
         family tcc {
            protocol inet
            remote { # Addresses associated with the Ethernet TCC neighbor Router C.
               mac-address cc:cc:cc:cc:cc:cc; # Or, specify Router C's IP address here.
            }
            proxy { # Addresses associated with the other TCC neighbor—Router A.
               inet-address 10.1.1.1;
            }
         }
      }
   }
}
protocols {
   mpls {
      interface so-0/0/0.600;
      interface fe-1/0/0.0;
   }
   connections {
      interface-switch FR-to-Ether {
         interface so-0/0/0.600;
         interface fe-1/0/0.0;
      }
   }
}
```

Ethernet encapsulation is the default for Router C. Because the Fast Ethernet connection is switched at Layer 2 to reach the Frame Relay link, it is transparent to Layer 3 addressing. As a result, you must configure the IP address for the fe-0/0/0 interface in the same address space as Router A's so-0/1/0.600 interface.

Optionally, configure static ARP on the fe-0/0/0 interface if you omit the proxy statement on Router B. The arp statement must contain the IP address from interface so-0/1/0.600 on Router A and the MAC address of the Fast Ethernet interface on Router B.

Router C

```
[edit]
interfaces
fe-0/0/0 {
   description "to Router B fe-1/0/0";
   unit 0 {
      family inet {
         address 10.1.1.2/30 {
            arp 10.1.1.1 mac bb:bb:bb:bb:bb:bb; # Configure this only if you did not
         }
      }
   }
}
```

Verifying Your Work

To verify the operational status of your TCC connection, use the show connections command on Router B:

```
user@router_b> show connections
CCC and TCC connections [Link Monitoring On]
  Legend for status (St)              Legend for connection types
  UN -- uninitialized                 if-sw:  interface switching
  NP -- not present                   rmt-if: remote interface switching
  WE -- wrong encapsulation           lsp-sw: LSP switching
  DS -- disabled
  Dn -- down                          Legend for circuit types
  -> -- only outbound conn is up      intf -- interface
  <- -- only inbound  conn is up      tlsp -- transmit LSP
  Up -- operational                   rlsp -- receive LSP
Connection/Circuit                  Type    St  Time last up     # Up trans
FR-to-Ether                         if-sw   Up  Dec 30 09:57:23           1
  so-0/0/0.600                      intf    Up
  fe-1/0/0.0                        intf    Up
```

Configuring Layer 2.5 VPNs

Layer 2.5 VPNs are an extension to Layer 2 VPNs. The main difference is that Layer 2.5 VPNs are not required to have the same media on both ends of the connection. In general, Layer 2.5 VPNs combine the capabilities of TCC with Layer 2 VPNs.

Layer 2.5 VPNs support the same media types as TCC: ATM, Cisco HDLC, Ethernet, extended VLAN, Frame Relay, and PPP. They support IPv4, IS-IS, and MPLS traffic types, but do not support IPv6 traffic.

Although not covered in this manual, you can also use TCC in conjunction with Layer 2 circuits to connect two CE sites that use different Layer 2 protocols. You can use these so-called Layer 2.5 circuits as an alternative to Layer 2.5 VPNs. For more information about Layer 2 circuits, see "Layer 2 Circuits" on page 513 or the *JUNOS VPNs Configuration Guide*.

To configure Layer 2.5 VPNs, complete the following tasks:

Configuring the Encapsulation on Interfaces Participating in the Layer 2.5 VPN

The encapsulation types used for Layer 2.5 VPNs are parallel to CCC encapsulations and identical to the TCC encapsulations explained in "Configuring TCC Interface Switching" on page 644. The encapsulation types are:

- atm-tcc-vc-mux
- atm-tcc-snap
- cisco-hdlc-tcc
- ethernet-tcc
- extended-vlan-tcc
- frame-relay-tcc
- ppp-tcc

When you configure a TCC encapsulation type, some modifications are needed to handle VPN connections over unlike Layer 2 links and to terminate the Layer 2 protocol locally. The router performs the following media-specific changes:

- PPP TCC—Both Link Control Protocol (LCP) and Network Control Protocol (NCP) are terminated on the router. Internet Protocol Control Protocol (IPCP) IP address negotiation is not supported. The JUNOS software strips all PPP encapsulation data from incoming frames before forwarding them. For frames destined to a PPP-connected neighbor, PPP encapsulation is added.
- Cisco HDLC TCC—Keepalive processing is terminated on the router. The JUNOS software strips all Cisco HDLC encapsulation data from incoming frames before forwarding them. Cisco-HDLC encapsulation is added onto frames that are destined to a Cisco HDLC-connected neighbor.
- Frame Relay TCC—All Local Management Interface (LMI) processing is terminated on the router. The JUNOS software strips all Frame Relay encapsulation data

from incoming frames before forwarding them. For frames destined to a Frame Relay-connected neighbor, Frame Relay encapsulation is added.

- ATM—Operation, Administration, and Maintenance (OAM) and Interim Local Management Interface (ILMI) processing is terminated at the router. Cell relay is not supported. The JUNOS software strips all ATM encapsulation data from incoming frames before forwarding them. ATM encapsulation is added onto frames that are destined to an ATM-connected neighbor.

Configuring the Layer 2.5 VPN

Layer 2.5 VPNs use essentially the same configuration format as Layer 2 VPNs. You should already have experience configuring Layer 2 VPNs before you attempt to configure a Layer 2.5 VPN. The major steps required to create a Layer 2 VPN are:

- Enable MPLS on interfaces pointing toward the core and the edge on your provider edge (PE) and provider core (P) routers.
- Configure Label Distribution Protocol (LDP) on all P and PE routers for traffic traveling from the PEs, through the core, and to the remote PEs.
- Establish an internal BGP (IBGP) Layer 2 VPN peering relationship between PE routers.
- Set up policies on your PE routers that will set a private community tag on outbound BGP traffic heading to the remote PEs and accept incoming traffic that matches similar community traffic from the remote PEs.
- Build VPN routing and forwarding (VRF) instances on your PE routers and apply the previously configured policies to deliver private traffic to the customer edge (CE) routers.
- In addition to the above configurations procedures, you can specify a Layer 3 protocol family for a Layer 2.5 VPN on T-series and M320 routers. See the “Option: Configuring ISO or MPLS Traffic on T-series and M320 Routers” on page 655 section for configuration information.

Option: Configuring ISO or MPLS Traffic on T-series and M320 Routers

Layer 2.5 VPNs on T-series and M320 routers support IPv4, IS-IS, and MPLS traffic types. Layer 2.5 VPNs running on M-series routers support only IPv4 traffic. Layer 2.5 VPNs do not support IPv6.

By default, IPv4 traffic runs on T-series and M320 routers. To configure IS-IS (ISO traffic) or MPLS traffic on Layer 2.5 VPNs, you must configure both ends of the VPN with the protocol configuration.

The same type of Layer 3 traffic must be configured on both ends of a TCC connection. By default, TCC connections carry IPv4 traffic. To specify which traffic can run over a TCC interface, include the [inet | mpls | iso] statement at the [edit interfaces *interface-name* encapsulation *encapsulation-type* logical-unit-number family tcc protocol] hierarchy level:

[edit]

```
interfaces {
   interface so-2/0/0  {
      encapsulation ppp-tcc;
         unit 0  {
         family tcc {
            protocols [iso | inet | mpls];
         }
      }
   }
}
```

When enabling ISO over a Layer 2.5 VPN that is configured on a CE Ethernet interface, include the [point-to-point] statement at the [edit protocols isis interface *interface-name*] hierarchy level:

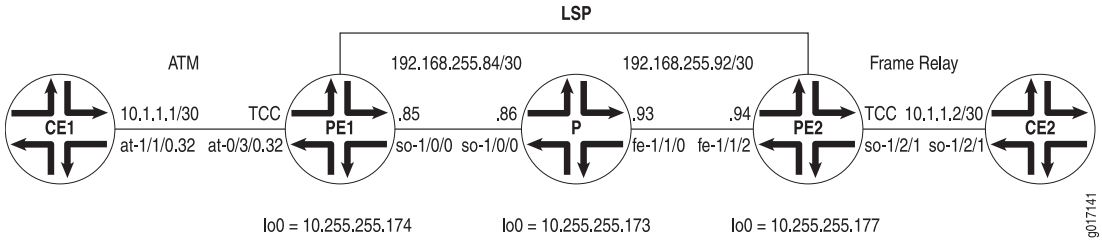
```
[edit]
protocols {
   isis {
      interface fe-1/0/0.0 {
         point-to-point;
      }
   }
}
```

For a full explanation of Layer 2 VPNs and configuration samples, see the *JUNOS VPNs Configuration Guide*.

Example: Layer 2.5 VPN Configuration

This section contains a configuration example and commands you can issue to verify your Layer 2.5 VPN configuration:

Figure 69: Layer 2.5 VPN Topology Diagram



In Figure 69 on page 656, ATM is configured between CE1 and PE1 and Frame Relay is configured between PE2 and CE2. To begin the Layer 2 VPN configuration, enable ATM and the corresponding encapsulation on CE1.

Router CE1

```
[edit]
interfaces
at-1/1/0 {
   description "to PE1 at-0/3/0";
   atm-options {
      vpi 0 maximum-vcs 2000;
   }
```

```
    unit 32 {
      vci 32;
      encapsulation atm-vc-mux;
      family inet {
        address 10.1.1.1/30;
      }
    }
  }
```

The first provider edge (PE1) router uses ATM TCC encapsulation on the ATM VC connecting to CE1. After this, standard Layer 2 VPN design rules apply. You use MPLS on interfaces pointing toward the core and the edge, establish a Layer 2 VPN BGP peer relationship with PE2, use LDP or Resource Reservation Protocol (RSVP) for traffic traveling through the core, and configure the proper VRF instance. Finally, you create policies for PE1 that will set a private community tag on outbound BGP traffic heading to PE2 and accept incoming traffic that matches similar community traffic from PE2.

Router PE1

```
[edit]
interfaces {
  at-0/3/0 {
    description "to CE1 at-1/1/0";
    atm-options {
      vpi 0 maximum-vcs 2000;
    }
    unit 32 {
      encapsulation atm-tcc-vc-mux;
      vci 32;
    }
  }
  so-1/0/0 {
    description "to P so-1/0/0";
    unit 0 {
      family inet {
        address 192.168.255.86/30;
      }
      family mpls;
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.255.255.174/32;
      }
    }
  }
}
protocols {
  mpls {
    interface at-0/3/0.32;
    interface so-1/0/0.0;
  }
  bgp {
    group my-internal-peers {
      type internal;
```

```
            local-address 10.255.255.174;
            family l2vpn {
               signaling;
            }
            neighbor 10.255.255.177;
         }
      }
      ldp {
         interface so-1/0/0.0;
      }
   }
   policy-options {
      policy-statement companyA-import {
         term T1 {
            from {
               protocol bgp;
               community companyA;
            }
            then accept;
         }
         term Final {
            then reject;
         }
      }
      policy-statement companyA-export {
         term T1 {
            then {
               community add companyA;
               accept;
            }
         }
         term Final {
            then reject;
         }
      }
      community companyA members target:100:1;
   }
   routing-instances {
      companyA {
         instance-type l2vpn;
         interface at-0/3/0.32;
         route-distinguisher 10.255.255.174:1;
         vrf-import companyA-import;
         vrf-export companyA-export;
         protocols {
            l2vpn {
               encapsulation-type interworking;
               site Denver {
                  site-identifier 1;
                  interface at-0/3/0.32 {
                     remote-site-id 2;
                  }
               }
            }
         }
      }
```

```
}
```

On the provider core router (P), you need only enable MPLS and LDP on the interfaces that bridge the gap between the PE routers.

Router P

```
[edit]
interfaces {
   so-1/0/0 {
      description "to PE1 so-1/0/0";
      unit 0 {
         family inet {
            address 192.168.255.85/30;
         }
         family mpls;
      }
   }
   fe-1/1/0 {
      description "to PE2 fe-1/1/2";
      unit 0 {
         family inet {
            address 192.168.255.93/30;
         }
         family mpls;
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.255.255.173/32;
         }
      }
   }
}
protocols {
   mpls {
      interface so-1/0/0.0;
      interface fe-1/1/0.0;
   }
   ldp {
      interface so-1/0/0.0;
      interface fe-1/1/0.0;
   }
}
```

The PE2 router uses Frame Relay TCC encapsulation on the Frame Relay DLCI connecting to CE2. To establish the Layer 2.5 VPN, follow the same steps you used to configure PE1. You use MPLS on interfaces pointing toward the core and the edge, establish a Layer 2 VPN BGP peer relationship with PE1, use LDP or RSVP for traffic traveling through the core, and configure the proper VRF instance. Finally, you create policies on PE2 that will set a private community tag on outbound BGP traffic heading to PE1 and accept incoming traffic that matches similar community traffic from PE1.

Router PE2

```
[edit]
interfaces {
   fe-1/1/2 {
```

```
        description "to P fe-1/1/0";
        unit 0 {
            family inet {
                address 192.168.255.94/30;
            }
            family mpls;
        }
    }
    so-1/2/1 {
        description "to CE2 so-1/2/1";
        dce;
        encapsulation frame-relay-tcc;
        unit 600 {
            encapsulation frame-relay-tcc;
            dlci 600;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.255.177/32;
            }
        }
    }
}
protocols {
    mpls {
        interface fe-1/1/2.0;
        interface so-1/2/1.600;
    }
    bgp {
        group my-internal-peers {
            type internal;
            local-address 10.255.255.177;
            family l2vpn {
                signaling;
            }
            neighbor 10.255.255.174;
        }
    }
}
ldp {
    interface fe-1/1/2.0;
}
policy-options {
    policy-statement companyA-import {
        term T1 {
            from {
                protocol bgp;
                community companyA;
            }
            then accept;
        }
        term Final {
            then reject;
        }
```

```
    }
    policy-statement companyA-export {
        term T1 {
            then {
                community add companyA;
                accept;
            }
        }
        term Final {
            then reject;
        }
    }
    community companyA members target:100:1;
}
routing-instances {
    companyA {
        instance-type l2vpn;
        interface so-1/2/1.600;
        route-distinguisher 10.255.255.177:1;
        vrf-import companyA-import;
        vrf-export companyA-export;
        protocols {
            l2vpn {
                encapsulation-type interworking;
                site NewYork {
                    site-identifier 2;
                    interface so-1/2/1.600 {
                        remote-site-id 1;
                    }
                }
            }
        }
    }
}
```

To complete the Layer 2.5 VPN configuration, enable Frame Relay encapsulation on CE2.

Router CE2

```
[edit]
interfaces
so-1/2/1 {
    description "to PE2 so-1/2/1";
    encapsulation frame-relay;
    unit 600 {
        dlci 600;
        family inet {
            address 10.1.1.2/30;
        }
    }
}
```

Verifying Your Work

To verify the operational status of your Layer 2.5 VPN, use the following commands:

- show route forwarding-table
- show ldp database
- show l2vpn connections
- show bgp summary
- show route

To view sample output of these commands as used with the configuration example, see the following:

- Router PE1 Status on page 662
- Router PE2 Status on page 664
- Router P Status on page 665

Router PE1 Status

```
user@PE1> show route forwarding-table
<snip>
Routing table:: ccc
MPLS:
Interface.Label    Type RtRef Nexthop           Type Index NhRef Netif
default            perm     0                  dscd    10     1
0                  user     0                  recv    12     2
1                  user     0                  recv    12     2
100128             user     0                  Pop           so-1/0/0.0
100128(S=0)        user     0                  Pop           so-1/0/0.0
100129             user     0                  Swap  100000  so-1/0/0.0
800001             user     0                  ucst   137     1 at-0/3/0.32
at-0/3/0. (CCC)    user     0                  indr   133     2
                                               Push 800000, Push 100000(top)
so-1/0/0.0
<snip>

user@PE1> show ldp database
Input label database, 10.255.255.174:0-10.255.255.173:0
  Label     Prefix
 100002     10.255.255.174/32
 100000     10.255.255.177/32
      3     10.255.255.173/32
Output label database, 10.255.255.174:0-10.255.255.173:0
  Label     Prefix
 100128     10.255.255.173/32
 100129     10.255.255.177/32
      3     10.255.255.174/32

user@PE1> show l2vpn connections
L2VPN Connections:
Legend for connection status (St)   Legend for interface status
OR -- out of range                  up -- operational
```

```
EI -- encapsulation invalid         Dn -- down
EM -- encapsulation mismatch        NP -- no present
CN -- circuit not present           DS -- disabled
OL -- no outgoing label             WE -- wrong encapsulation
Dn -- down                          UN -- uninitialized
VC-Dn -- Virtual circuit down
WE -- intf encaps != instance encaps
-> -- only outbound conn is up
<- -- only inbound  conn is up
UP -- operational
XX -- unknown
Instance: companyA
Local site: Denver (1)
    connection-site           Type  St     Time last up          # Up trans
    2                         rmt   Up     Nov 30 08:21:07 2001           1
      Local interface: at-0/3/0.32, Status: Up, Encapsulation: INTERWORKING
      Remote PE: 10.255.255.177
      Incoming label: 800001, Outgoing label: 800000

user@PE1>  show bgp summary
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 0          0          0          0          0          0
bgp.l2vpn.0            1          1          0          0          0          0
Peer               AS      InPkt     OutPkt    OutQ   Flaps Last
Up/DwnState|#Active/Received/Damped...
10.255.255.177      69         49         45       0       1       19:16 Establ
  bgp.l2vpn.0: 1/1/0
  companyA.l2vpn.0: 1/1/0

user@PE1> show route
<snip>
mpls.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
0                  *[MPLS/0] 1d 18:54:24, metric 1
                      Receive
1                  *[MPLS/0] 1d 18:54:24, metric 1
                      Receive
100128             *[LDP/9] 00:24:03, metric 1
                    > via so-1/0/0.0, Pop
100128(S=0)        *[LDP/9] 00:24:03, metric 1
                    > via so-1/0/0.0, Pop
100129             *[LDP/9] 00:24:03, metric 1
                    > via so-1/0/0.0, Swap 100000
800001             *[L2VPN/7] 00:10:35
                    > via at-0/3/0.32, Pop      [0]
at-0/3/0.32        *[L2VPN/7] 00:10:35
                    > via so-1/0/0.0, Push 800000, Push 100000(top)
companyA.l2vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
1:1:1:1        /96
                   *[L2VPN/7] 00:19:55
                      Discard
1:1:2:1        /96
                   *[BGP/170] 00:06:46, localpref 100, from 10.255.255.177
                      AS path: I
                    > via so-1/0/0.0, Push 100000
bgp.l2vpn.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
1:1:2:1        /96
                   *[BGP/170] 00:10:35, localpref 100, from 10.255.255.177
```

```
                    AS path: I
                  > via so-1/0/0.0, Push 100000
<snip>
```

Router PE2 Status

```
user@vpn07> show route forwarding-table
<snip>
Routing table:: ccc
MPLS:
Interface.Label    Type RtRef Nexthop           Type Index NhRef Netif
default            perm     0                   dscd     8     1
0                  user     0                   recv    10     2
1                  user     0                   recv    10     2
100002             user     0                   Pop           fe-1/1/2.0
100002(S=0)        user     0                   Pop           fe-1/1/2.0
100003             user     0                   Swap  100002  fe-1/1/2.0
800000             user     0                   ucst    60     1 so-1/2/1.0
so-1/2/1. (CCC)    user     0                   indr    59     2
<snip>

user@vpn07>  show ldp database
Input label database, 10.255.255.177:0-10.255.255.173:0
  Label     Prefix
 100000     10.255.255.177/32
      3     10.255.255.173/32
 100002     10.255.255.174/32
Output label database, 10.255.255.177:0-10.255.255.173:0
  Label     Prefix
 100002     10.255.255.173/32
      3     10.255.255.177/32
 100003     10.255.255.174/32

user@vpn07>  show l2vpn connections
L2VPN Connections:
Legend for connection status (St)   Legend for interface status
OR -- out of range                  up -- operational
EI -- encapsulation invalid         Dn -- down
EM -- encapsulation mismatch        NP -- no present
CN -- circuit not present           DS -- disabled
OL -- no outgoing label             WE -- wrong encapsulation
Dn -- down                          UN -- uninitialized
VC-Dn -- Virtual circuit down
WE -- intf encaps != instance encaps
-> -- only outbound conn is up
<- -- only inbound  conn is up
UP -- operational
XX -- unknown
Instance: companyA
Local site: NewYork (2)
    connection-site           Type  St     Time last up          # Up trans
    1                         rmt   Up     Nov 30 08:21:01 2001           1
      Local interface: so-1/2/1.0, Status: Up, Encapsulation: INTERWORKING
      Remote PE: 10.255.255.174
      Incoming label: 800000, Outgoing label: 800001

user@vpn07>  show bgp summary
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
```

```
bgp.l2vpn.0            1          1          0          0          0          0
inet.0                 0          0          0          0          0          0
Peer               AS      InPkt     OutPkt    OutQ   Flaps Last
Up/DwnState|#Active/Received/Damped...
10.255.255.174      69         45         52       0       0       20:20 Establ
  bgp.l2vpn.0: 1/1/0
  companyA.l2vpn.0: 1/1/0

user@vpn07>  show route
<snip>
mpls.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
0                  *[MPLS/0] 02:34:04, metric 1
                      Receive
1                  *[MPLS/0] 02:34:04, metric 1
                      Receive
100002             *[LDP/9] 00:25:39, metric 1
                    > via fe-1/1/2.0, Pop
100002(S=0)        *[LDP/9] 00:25:39, metric 1
                    > via fe-1/1/2.0, Pop
100003             *[LDP/9] 00:25:01, metric 1
                    > via fe-1/1/2.0, Swap 100002
800000             *[L2VPN/7] 00:07:50
                    > via so-1/2/1.0, Pop      [0]
so-1/2/1.0         *[L2VPN/7] 00:07:50
                    > via fe-1/1/2.0, Push 800001, Push 100002(top)
companyA.l2vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
1:1:1:1        /96
                   *[BGP/170] 00:04:59, localpref 100, from 10.255.255.174
                      AS path: I
                    > via fe-1/1/2.0, Push 100002
1:1:2:1        /96
                   *[L2VPN/7] 00:11:34
                      Discard
bgp.l2vpn.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
1:1:1:1        /96
                   *[BGP/170] 00:11:38, localpref 100, from 10.255.255.174
                      AS path: I
                    > via fe-1/1/2.0, Push 100002
<snip>
```

Router P Status

```
user@P> show ldp database
Input label database, 10.255.255.173:0-10.255.255.174:0
  Label     Prefix
 100128     10.255.255.173/32
 100129     10.255.255.177/32
      3     10.255.255.174/32
Output label database, 10.255.255.173:0-10.255.255.174:0
  Label     Prefix
      3     10.255.255.173/32
 100000     10.255.255.177/32
 100002     10.255.255.174/32
Input label database, 10.255.255.173:0-10.255.255.177:0
  Label     Prefix
```

```
      3     10.255.255.177/32
 100002     10.255.255.173/32
 100003     10.255.255.174/32
Output label database, 10.255.255.173:0-10.255.255.177:0
  Label     Prefix
      3     10.255.255.173/32
 100000     10.255.255.177/32
 100002     10.255.255.174/32
```

For More Information

For additional information on TCC or Layer 2.5 VPNs, see the following documents:

- *JUNOS VPNs Configuration Guide*
- *JUNOS MPLS Applications Configuration Guide*
- *JUNOS Network Interfaces Configuration Guide*

Revision History

13 April 2009—9.5R1 Release. Roy Spencer.

15 January 2009—9.4R1 Release. Roy Spencer.

10 October 2008—9.3R1 Release. Roy Spencer.

10 July 2008—9.2R1 Release. Roy Spencer.

10 April 2008—9.1R1 Release. Roy Spencer.

1 February 2008—9.0R1 Release. Fawn Damitio.

29 June 2007—Added support for MX-series routers. 8.4R1 Release.

27 March 2007—Added support for MPLS and ISIS traffic over Layer 2.5 VPN. 8.3R1 Release. Fawn Damitio.

12 January 2007—Added support for M120 router and MX960 router. 8.2R1 Release. Fawn Damitio.

15 September 2006—8.1R1 Release. Richard Hendricks.

29 June 2006—8.0R1 Release. Richard Hendricks.

27 March 2006—7.6R1 Release. Richard Hendricks.

9 January 2006—7.5R1 Release. Richard Hendricks.

14 September 2005—7.4R1 Release. Richard Hendricks.

13 June 2005—7.3R1 Release. Richard Hendricks.

5 April 2005—7.2R1 Release. Richard Hendricks.

2 February 2005—7.1R1 Release. Richard Hendricks.

6 October 2004—7.0R1 Release. Richard Hendricks.

6 July 2004—6.4R1 Release. Richard Hendricks.

5 April 2004—6.3R1 Release. Richard Hendricks.

22 December 2003—6.2R1 Release. Richard Hendricks.

22 September 2003—Mentioned Layer 2.5 circuits, 6.1R1 Release. Richard Hendricks.

30 June 2003—Added T-series support for TCC and Layer 2.5 VPNs, 6.0R1 Release. Elizabeth Lichtenberg.

2 April 2003—5.7R1 Release. Richard Hendricks.

27 December 2002—Added proxy ARP, 5.6R1 Release. Richard Hendricks.

30 September 2002—5.5R1 Release. Richard Hendricks.

19 July 2002—5.4R1 Release. Richard Hendricks.

6 June 2002—Document reformatted, added Ethernet TCC. Richard Hendricks.

21 November 2001—Initial document created. Gary Matthews.

Chapter 15

Virtual Private LAN Service

This feature guide covers these topics:

Overview

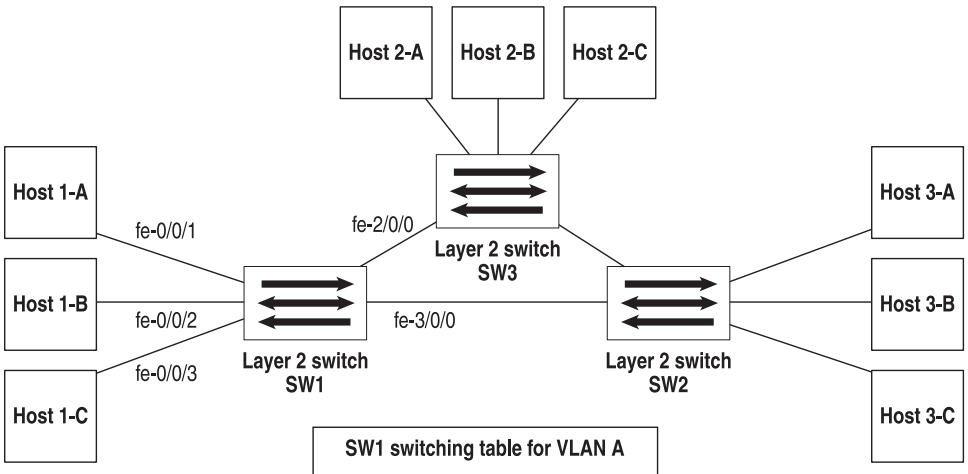
Ethernet is an increasingly important component of a service provider's slate of service offerings. Many customers are requesting the ability to connect local area network (LAN) locations across the country and around the world. To fulfill customer desire, service providers have had to set up complex point-to-point Layer 2 virtual private networks (VPNs) or connect expensive Layer 2 switches to handle traffic.

Virtual private LAN service (VPLS) meets the growing Ethernet needs of service providers and their customers. VPLS is an Ethernet-based multipoint-to-multipoint Layer 2 VPN. With VPLS, multiple Ethernet LAN sites can be connected to each other across an MPLS backbone. To the customer, all sites interconnected by VPLS appear to be on the same Ethernet LAN (even though traffic travels across a service provider network).

This guide explains the background knowledge you need to understand VPLS and provides detailed steps for you to follow to implement it in your network.

Before VPLS, the only way you could connect Ethernet LAN sites together was to set up a non-VPLS Layer 2 VPN or install multiple Layer 2 Ethernet switches. Figure 70 on page 671 shows how three switches can be connected to each other.

Figure 70: Ethernet Switching Example



SW1 switching table for VLAN A

Port	MAC Address
fe-0/0/1	1a1a.1a1a.1a1a
fe-2/0/0	2a2a.2a2a.2a2a
fe-3/0/0	3a3a.3a3a.3a3a

SW1 switching table for VLAN B

Port	MAC Address
fe-0/0/2	1b1b.1b1b.1b1b
fe-2/0/0	2b2b.2b2b.2b2b
fe-3/0/0	3b3b.3b3b.3b3b

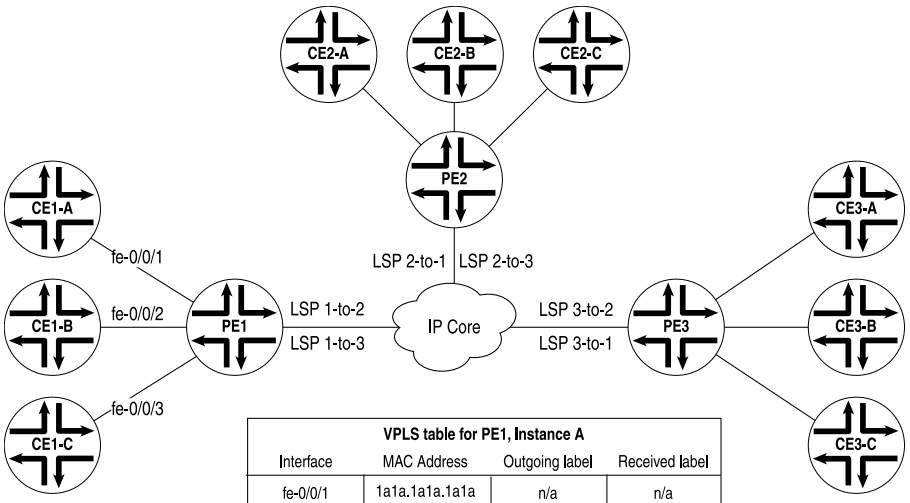
SW1 switching table for VLAN C

Port	MAC Address
fe-0/0/3	1c1c.1c1c.1c1c
fe-2/0/0	2c2c.2c2c.2c2c
fe-3/0/0	3c3c.3c3c.3c3c

A typical switch builds its Layer 2 switching table with MAC address and interface information learned from traffic received from other switches. If a switch does not know how to reach a particular destination, it floods traffic for that destination to all ports except the one where the traffic originated. When information about a previously unknown destination is received, this information is added to the switching table. If a destination is known, the switch sends the traffic directly to the intended recipient through the associated port listed in the switching table.

Figure 71 on page 672 shows a VPLS network comparable to the switch example and explains how VPLS functions similarly to Ethernet switches (assuming a Spanning Tree Protocol is configured).

Figure 71: VPLS Introductory Example



VPLS table for PE1, Instance A

Interface	MAC Address	Outgoing label	Received label
fe-0/0/1	1a1a.1a1a.1a1a	n/a	n/a
vt-0/3/0.32770	2a2a.2a2a.2a2a	800000	800002
vt-0/3/0.32773	3a3a.3a3a.3a3a	800012	800014

VPLS table for PE1, Instance B

Interface	MAC Address	Outgoing label	Received label
fe-0/0/2	1b1b.1b1b.1b1b	n/a	n/a
vt-0/3/0.32771	2b2b.2b2b.2b2b	800004	800006
vt-0/3/0.32774	3b3b.3b3b.3b3b	800016	800018

VPLS table for PE1, Instance C

Interface	MAC Address	Outgoing label	Received label
fe-0/0/3	1c1c.1c1c.1c1c	n/a	n/a
vt-0/3/0.32772	2c2c.2c2c.2c2c	800008	800010
vt-0/3/0.32775	3c3c.3c3c.3c3c	800020	800022

Notice that Layer 2 information gathered by a switch (for example, MAC addresses and interface ports) is included in the VPLS instance table. However, instead of requiring all VPLS interfaces to be physical switch ports, the router allows remote traffic for a VPLS instance to be delivered across an MPLS label-switched path (LSP) and arrive on a virtual port. The virtual port emulates a local, physical port. Traffic can be learned, forwarded, or flooded to the virtual port almost identically to the way traffic is sent to a local port.

The VPLS table learns MAC address and interface information for both physical and virtual ports. If no activity is seen for a particular MAC address, it is purged from the table over time.

As shown in Figure 71 on page 672, the main difference between a physical port and a virtual port is that the router captures additional information from a virtual port—an outgoing MPLS label used to reach the remote site and an incoming MPLS label for VPLS traffic received from the remote site.

When you configure VPLS on a routing platform, a virtual port is generated as a logical interface on a virtual loopback tunnel (vt) interface or a label-switched interface (LSI). On M-series and T-series routers, virtual ports are created dynamically on vt interfaces if you install a Physical Interface Card (PIC) that supports virtual tunnels. With VPLS, you must install at least one Tunnel Services, Link Services, or Adaptive Services PIC in each VPLS provider edge (PE) router. On MX-series routers, virtual ports are created dynamically on vt interfaces if you configure tunnel services on

one of the four Packet Forwarding Engines (PFEs) included in each DPC. If your routing platform does not offer tunnel services through a PIC or PFE, you can configure VPLS to create virtual ports on LSI logical interfaces.

One property of flooding behavior in VPLS is that traffic received from remote PE routers is never forwarded to other PE routers. This restriction helps prevent loops in the core network. If a customer edge (CE) Ethernet switch has two connections or more to the same PE router, you must enable the Spanning Tree Protocol to prevent loops. For more information on configuring the Spanning Tree Protocol, see the *JUNOS Routing Protocols Configuration Guide*.

The paths carrying VPLS traffic between each PE router participating in a routing instance are called pseudowires. The pseudowires are signaled using either BGP or LDP.

System Requirements

To implement VPLS, your system must meet these minimum requirements:

- JUNOS Release 9.1 or later for nonstop active routing (NSR), VPLS ping on M120, M320, and MX-series routers, and automatic site selection for BGP-signaled VPLS
- JUNOS Release 9.0 or later for Virtual Spanning Tree Protocol (VSTP) support, 802.1p classification in Bridged Ethernet over ATM mode support, interworking between LDP and BGP signaling in a VPLS instance, and Layer 2 VPLS filters for MX960 routers
- JUNOS Release 8.4 or later for VPLS with LDP signaling. Also, integrated routing and bridging (IRB) is supported starting in this release
- JUNOS Release 8.3 or later for point-to-multipoint LSP support on VPLS
- JUNOS Release 8.2 or later for VPLS support on MX-series routing platform, VPLS graceful Routing Engine switchover (GRES) support, and VPLS support on Gigabit Ethernet IQ2 aggregated Ethernet interfaces
- JUNOS Release 7.6 or later for VPLS support on LSI logical interfaces
- JUNOS Release 7.5 or later for multihoming a CE router to multiple PE routers
- JUNOS Release 7.3 or later for VPLS per-packet load balancing, support for limiting MAC address learning per interface in a VPLS domain, and migration to the VPLS and Layer 2 VPN signaling statement at the [edit protocols bgp groups *group-name* family l2vpn] hierarchy level
- JUNOS Release 6.4 or later to implement Ethernet VPLS over ATM LLC interface encapsulation on T-series and M320 routing platforms, to select the tunnel-enabled PICs that provide virtual ports for VPLS operation, and to issue the show vpls statistics command
- JUNOS Release 6.3 or later to clear MAC addresses from the VPLS table and to modify VPLS table timeout intervals
- JUNOS Release 6.2 or later for VPLS class of service (CoS), VPLS graceful restart, VPLS interinstance bridging and routing, VPLS source and destination MAC address accounting, VPLS virtual port support on the Adaptive Services PIC for M-series routers, and general VPLS support for T-series and M320 routing platforms

- JUNOS Release 6.1 or later for VPLS policers and filters
- JUNOS Release 6.0 or later for Ethernet VPLS over ATM LLC interface encapsulation on M-series routers
- JUNOS Release 5.7 or later for VPLS with BGP signaling and Ethernet VPLS, VLAN VPLS, and extended VLAN VPLS interface encapsulations
- Two Juniper Networks M-series (except the M160 router), MX-series, T-series, or TX Matrix routing platforms for the provider edge (PE)
- On M-series and T-series routing platforms, one Adaptive Services PIC, Link Services PIC, or Tunnel Services PIC per routing platform to create VPLS virtual ports on vt interfaces
- On M-series and T-series routing platforms, one Fast Ethernet or Gigabit Ethernet PIC per routing platform (from this list):
  - 4-port Fast Ethernet PIC with 10/100 Base-TX interfaces
  - 1-port, 2-port, or 10-port Gigabit Ethernet PIC
  - 4-port, quad-wide Gigabit Ethernet PIC
  - 1- and 2-port Gigabit Ethernet Intelligent Queuing (IQ) PIC
  - 4- and 8-port Gigabit Ethernet IQ2 PIC with small form-factor pluggable transceivers (SFPs)
  - 1-, 2-, and 4-port Gigabit Ethernet PIC with SFPs
  - 1-port 10-Gigabit Ethernet PIC

Terms and Acronyms

V

virtual port A special logical interface that is generated dynamically when you configure VPLS on a PE router. Virtual ports send and receive VPLS traffic for remote PE routers as if the remote VPLS sites had Ethernet-based interfaces directly connected to the local PE router. To generate virtual ports, VPLS PE routing platforms use logical interfaces on a vt interface (that is generated by the Tunnel Services PIC, Link Services PIC, Adaptive Services PIC, an LSI interface, or a tunnel services interface configured on MX-series routers).

virtual private LAN service (VPLS) An Ethernet-based multipoint-to-multipoint Layer 2 VPN service used for interconnecting multiple Ethernet LANs across an MPLS backbone. BGP-based VPLS is based on the Internet Engineering Task Force (IETF) Internet draft draft-ietf-l2vpn-vpls-bgp-08.txt, *Virtual Private LAN Service (VPLS) Using BGP for Auto-discovery and Signaling* (expires December 2006). LDP-based VPLS is specified in the IETF draft *Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling*. For more information about VPLS, see the *JUNOS VPNs Configuration Guide*.

Configuring VPLS

The following sections show the configuration steps necessary to implement VPLS :

Configuring Routing Protocols on the PE and Core Routers

At a fundamental level, VPLS is a type of Layer 2 VPN. All forms of Layer 2 VPNs require you to configure network protocols to handle *intradomain routing* (an interior

gateway protocol [IGP], such as Open Shortest Path First [OSPF] or Intermediate System-to-Intermediate System [IS-IS]), *interdomain routing* (Border Gateway Protocol [BGP]), *label switching* (Multiprotocol Label Switching [MPLS]), and *path signaling* (Resource Reservation Protocol [RSVP] or Label Distribution Protocol [LDP]). For more information about these protocols and examples of how to configure these protocols to support a Layer 2 VPN, see the *JUNOS VPNs Configuration Guide.*

NOTE: The 8-port, 12-port, and 48-port dense Fast Ethernet Physical Interface Cards (PICs) cannot push more than two labels onto an MPLS packet. Because of this, we do not recommend that you configure these PICs as core-facing or equivalent interfaces.

Configuring VPLS Encapsulation on CE-Facing Interfaces

There are four types of VPLS interface encapsulation: Ethernet VPLS, Ethernet VPLS over ATM LLC, VLAN VPLS, and extended VLAN VPLS. When one of these encapsulations is applied to an interface, a family type of VPLS is enabled by default. The encapsulation types are:

- ether-vpls-over-atm-llc—Use Ethernet VPLS over ATM LLC encapsulation on ATM2 IQ logical interfaces. Use this encapsulation type to support IEEE 802.1p classification binding on ATM VCs. This encapsulation type enables a VPLS instance to support bridging between Ethernet interfaces and ATM interfaces, as described in RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*. When you use this encapsulation type, you configure it on logical interfaces only and you cannot configure multipoint interfaces.
- extended-vlan-vpls—Use extended VLAN VPLS encapsulation on Ethernet interfaces that have VLAN 802.1Q tagging and VPLS enabled and that must accept packets carrying TPIDs 0x8100, 0x9100, and 0x9901.

NOTE: The built-in Gigabit Ethernet PIC on the M7i router does not support MPLS.

- ethernet-vpls—Use Ethernet VPLS encapsulation on Ethernet interfaces that have VPLS enabled and must accept packets carrying standard Tag Protocol ID (TPID) values.
- vlan-vpls—Use VLAN VPLS encapsulation on Ethernet interfaces with VLAN tagging enabled. VLAN VPLS encapsulation supports TPID 0x8100 only. You must configure this encapsulation type on both the physical interface and the logical interface.
- flexible-ethernet-services—Use flexible Ethernet services encapsulation when you want to configure multiple per-unit Ethernet encapsulations. This encapsulation type allows you to configure any combination of route, TCC, CCC, and VPLS encapsulations on a single physical port. Aggregated Ethernet bundles cannot use this encapsulation type.

Use the following guidelines to configure a VPLS interface:

- For encapsulation type vlan-vpls, VLAN IDs 1 through 511 are reserved for normal Ethernet VLANs, IDs 512 through 1023 are reserved for VPLS VLANs on Fast Ethernet interfaces, and IDs 512 through 4094 are reserved for VPLS VLANs on Gigabit Ethernet interfaces. For encapsulation type extended-vlan-vpls, all VLAN IDs from 1 through 1023 are valid for VPLS VLANs on Fast Ethernet interfaces, and all VLAN IDs from 1 through 4094 are valid for VPLS VLANs on Gigabit Ethernet interfaces. VLAN ID 0 is reserved for priority tagging. For encapsulation type flexible-ethernet-services, VLAN IDs from 1 through 511 are no longer reserved for normal VLANs.

To configure VPLS interface encapsulation for an Ethernet interface, include the encapsulation statement at the [edit interfaces *interface-fpc/pic/port*] hierarchy level and select ethernet-vpls, vlan-vpls, extended-vlan-vpls, flexible-ethernet-services or ether-vpls-over-atm-llc as the encapsulation type. If you select the VLAN VPLS encapsulation, also include the vlan-vpls statement at the [edit interfaces *ethernet-interface-fpc/pic/port* unit *unit-number* encapsulation] logical interface hierarchy level. When using either VLAN VPLS or extended VLAN VPLS encapsulations, include the vlan-tagging statement at the [edit interfaces *ethernet-interface-fpc/pic/port*] hierarchy level.

To configure VPLS interface encapsulation for an ATM2 IQ interface, include the encapsulation statement at the [edit interfaces at-*fpc/pic/port*] hierarchy level and select ether-vpls-over-atm-llc as the encapsulation type. To configure VPLS interface encapsulation for a gigabit Ethernet IQ interface or gigabit Ethernet PICs with small form-factor pluggable transceivers (SFPs), include the encapsulation statement at the [edit interfaces ge-*fe/pic/port*] hierarchy level and select flexible-ethernet-services as the encapsulation type.

```
[edit]
interfaces {
   ge-0/1/0 {
      vlan-tagging;
      encapsulation vlan-vpls;
      unit 0 {
         encapsulation vlan-vpls;
         vlan-id 600;
      }
   }
   at-0/2/0 {
      encapsulation ether-vpls-over-atm-llc;
   }
}
```

Configuring LDP Signaling for VPLS

Like other Layer 2 VPNs, you must enable a VPLS instance to isolate VPLS traffic from other network traffic. An important element of a VPLS instance is the signaling protocol. You can configure BGP signaling, LDP signaling, or both BGP and LDP signaling in a VPLS instance.

To configure LDP signaling, you must first enable a VPLS instance to isolate VPLS traffic from other network traffic. To configure, include the instance-type vpls statement

at the [edit routing-instances *instance-name*] hierarchy level. To configure LDP signaling within the instance, identify the virtual circuit with the vpls-id statement and specify the PE routers participating in the instance with the neighbor statement:

```
[edit]
routing-instances {
   instance-name {
      instance-type vpls;
      interface ge-0/1/0.0;
      protocols {
         vpls {
            vpls-id id-name;
            neighbor neighbor-id; # The neighbor-id should be the loopback address of
               # the remote PE router.
         }
      }
   }
}
```

To fully enable LDP signaling on a PE in a VPLS instance, you must also enable ldp on the loopback interface of the router. To configure, include the interface lo0.0 statement at the [edit protocols ldp] hierarchy level:

```
[edit]
protocols {
   ldp {
      interface lo0.0;
   }
}
```

For LDP signaling within a VPLS routing instance, the JUNOS software supports the following values only:

- FEC—FEC 128
- Control bit—0
- Ethernet pseudowire type—hexadecimal 0x0005

Configuring a VPLS Instance with BGP Signaling

Like other Layer 2 VPNs, you must enable a VPLS instance to isolate VPLS traffic from other network traffic. An important element of a VPLS instance is the signaling protocol. You can configure BGP signaling, LDP signaling, or both BGP and LDP signaling in a VPLS instance.

You must enable a VPLS instance to isolate VPLS traffic from other network traffic. To configure, include the instance-type vpls statement at the [edit routing-instances *instance-name*] hierarchy level.

Within the instance, you can define the maximum number of sites that can participate in this VPLS instance, a local site name, and a local site identifier. To configure the maximum number of sites, include the site-range statement at the [edit

routing-instances *instance-name* protocols vpls] hierarchy level. The maximum number of sites is 65,535.

NOTE: The site ID must be less than the site range. If you specify a site ID that is greater than the site range, your connections do not come up, even though the commit operation succeeds.

To configure a site name, include the site statement at the [edit routing-instances *instance-name* protocols vpls] hierarchy level. To configure the site ID, include the site-identifier statement at the [edit routing-instances *instance-name* protocols vpls site *name*] hierarchy level.

```
[edit]
routing-instances;
   instance-name {
      instance-type vpls;
      interface ge-0/1/0.0;
      route-distinguisher 10.245.14.218:1;
      vrf-target target:11111:1;
      protocols {
         vpls {
            site-range 10;
            site greenPE1 {
               site-identifier 1;
            }
         }
      }
   }
```

To complete the configuration, you must configure the Layer 2 VPN family for BGP by including the signaling statement at the [edit protocols bgp family l2vpn] hierarchy level:

```
[edit]
protocols {
   bgp {
      family l2vpn;
         signaling;
      }
   }
}
```

Configuring Interworking between BGP Signaling and LDP Signaling in VPLS Instances

If you want to configure a VPLS instance with both BGP and LDP-signaled pseudowires, you must configure a VPLS border router. Without a VPLS border router, LDP-signaled PEs and BGP-signaled PEs will be unaware of one another and the VPLS instance will not be fully meshed.

NOTE: Interworking between BGP signaling and LDP signaling in VPLS instances is supported only on MX-series and M320 routers.

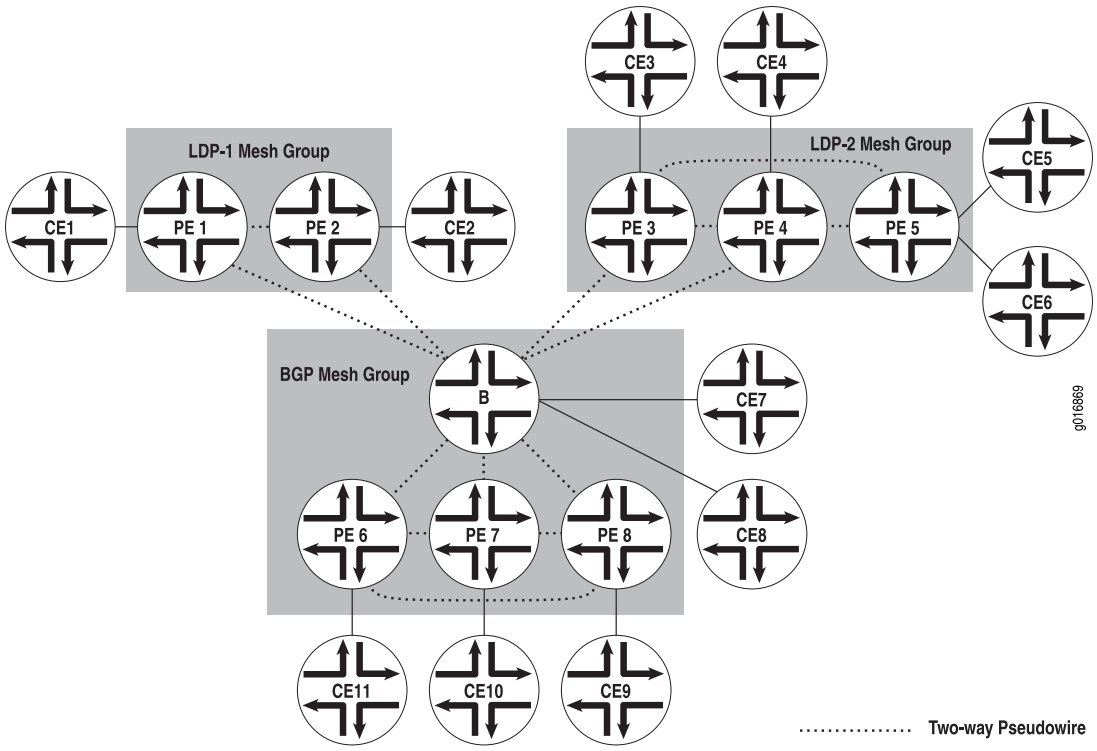
To enable interworking between BGP-signaled PE routers and LDP-signaled PE routers, you configure a border router to interconnect both sets of routers within the same VPLS routing instance. You also need to configure mesh groups on the border router to group the sets of PE routers that are fully meshed and which share the same signaling protocol, either BGP or LDP. You can configure multiple mesh groups to map each fully meshed LDP-signaled or BGP-signaled VPLS domain to a mesh group. In the data plane, the border router maintains a common MAC table used to forward traffic between the LDP-signaled and BGP-signaled mesh groups. When forwarding any VPLS traffic received over a PE router pseudowire, the border router ensures that traffic is not forwarded back to the PE routers, which are in same mesh group as the originating PE router.

There is always just one BGP mesh group in a VPLS instance, and it is created automatically when you configure BGP signaling for that instance. You can configure one or more LDP mesh groups. MX-series routers support up to 15 PE mesh groups (including the default BGP mesh group), and M-series and T-series routers support up to 127 PE mesh groups (including the default BGP mesh group).

In Figure 72 on page 681, Routers PE1 and PE2 are in the LDP-signaled mesh group “LDP-1”. Routers PE3, PE4, and PE5 are in the LDP-signaled mesh group “LDP-2”. Routers PE6, PE7, and PE8 are in the BGP-signaled mesh group “BGP-default”. As you can see, the border router is acting as a traditional PE (by connecting to CEs) in addition to being a border router. Every router shown in the topology below is in the same VPLS instance, bgp-ldp-mesh1.

When router CE1 sends a frame whose destination MAC address is CE9, PE1 receives the frame and performs a MAC address lookup. The MAC address is not in the PE1 MAC table and so PE1 floods the frame to the other PEs in the LDP-1 mesh group (PE2 and Router B), which from the perspective of PE1, are the only members of the VPLS network. When Router B receives the data from PE1, it does not find the MAC address in its MAC table and so it floods the frame to PE3, PE4, PE5, PE6, PE7, and PE8, but not back to PE1 or PE2. The PE routers then perform a MAC table lookup and flood the data to their CE routers.

Figure 72: Topology for BGP/LDP Interworking in a VPLS Instance



In this topology, you configure routers PE6, PE7, and PE8 as you traditionally configure BGP-signaled VPLS routers. You configure routers PE1, PE2, PE3, PE4, and PE5 as you traditionally configure LDP-signaled VPLS routers. In addition, you create the mesh group LDP-1 for Routers PE1 and PE2 and mesh group LDP-2 for Routers PE3, PE4, and PE5 by including the mesh-group *mesh-group-name* statement at the [edit routing-instances *routing-instance-name* protocols vpls] hierarchy level.

NOTE: The border router can act as a normal PE in addition to being a border router and can support local CE-facing interfaces.

To enable interworking between VPLS mesh groups, configure the border router by including the site *site-name* statement at the [edit routing instances *routing-instance-name* protocols] hierarchy level:

```
[edit]
routing-instances {
   bgp-ldp-mesh1 {
      instance-type vpls;
      route-distinguisher 10.245.14.218:1;
      interface fe-1/3/1.0; # these interfaces are CE interfaces. In this case,
         # the router is acting as both a border router and a regular PE.
      interface fe-1/3/2.0;
   vrf-target target:10:100;
}
protocols {
```

```
    vpls {
    site green {
        site-identifier 1;
    }
}
```

Configure LDP signaling with the vpls-id and neighbor *neighbor-id* statements. You can configure mesh groups LDP-1 and LDP-2 by including the mesh-group statement at the [edit routing instances *routing-instance-name* protocols vpls vpls *vpls-id* and including the neighbor *neighbor-id* statement at the [edit routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name*] hierarchy level:

```
[edit routing-instances bgp-ldp-mesh1 protocols vpls]
vpls-id 100;
mesh-group LDP-1 {
    neighbor 10.1.1.1;
    neighbor 20.1.1.1;
}
mesh-group LDP-2 {
    neighbor 30.1.1.1;
    neighbor 40.1.1.1;
    neighbor 10.1.1.1;
}
```

NOTE: When you configure BGP signaling to interoperate with LDP signaling in a VPLS network, the following features are not supported:

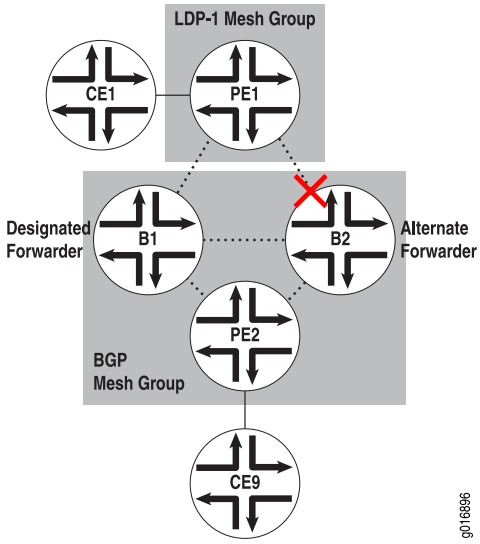
- Point-to-multipoint VPLS
- Integrated routing and bridging

Configuring Multihoming on a VPLS Border Router

Configuring multihoming on VPLS border routers ensures that if one border router is unreachable, BGP/LDP PE connectivity is maintained through the other VPLS border router. With multihoming, one border router is chosen as the designated forwarder for each mesh group. The designated forwarder is chosen through either the BGP or VPLS path-selection procedure. If the designated forwarder loses connectivity with a mesh group, the alternate border router then takes over as designated forwarder for that mesh group. A VPLS instance must be configured with BGP signaling in order for multihoming to work.

Figure 73 on page 683 shows a simplified example of how multihoming works with VPLS border routers. In this example, B1 is the designated forwarder and B2 is the alternate forwarder. If CE1 wanted to send data to CE9, the data would travel from CE1 to PE1, which is part of the LDP-1 mesh group. PE1 would then flood the data to B1 (the designated forwarder) which would forward the data to PE2. It would not send the data to Router B2. PE2 would then send the data to its destination, CE9. If B1 lost connectivity with the LDP-1 mesh group then B2 would become the designated forwarder. In this case, PE1 would send the data through B2, not through B1.

Figure 73: Multihoming for Border Area Routers



You configure multihoming on border routers by including the site-identifier and multi-homing statement at the [edit routing-instances *routing-instance-name* protocols] hierarchy level. The designated forwarder and alternate forwarder must be configured with the same site identifier.

Router B1

```
[edit routing-instances example protocols]
vpls {
    site mult-home-ldp-1 {
        site-identifier 1;
        mesh-group ldp-1;
        multi-homing;
    }
```

Router B2

```
[edit routing-instances example protocols]
vpls {
    site mult-home-ldp-1 {
        site-identifier 1;
        mesh-group ldp-1;
        multi-homing;
    }
```

For more information on multihoming, see “Option: Configuring VPLS Multihoming with BGP Signaling” on page 684.

Option: Selecting an LSP for the VPLS Routing Instance to Traverse

If you have two or more equal-cost-path LSPs between your VPLS PE router sites, you can select an LSP over which the VPLS traffic will travel. To select an LSP for VPLS traffic, assign the VPLS instance to a BGP community, define a policy that directs community traffic over a specified LSP, and then apply the policy to the forwarding table.

To configure a BGP community, include the community *community-name* statement at the [edit policy-options] hierarchy level. Be sure to specify the vrf-export or vrf-target values from the VPLS routing instance as community identifiers with the members *community-ids* statement at the [edit policy-options community *community-name*] hierarchy level.

To create a policy that sends community traffic over a specific LSP, include the community *community-name* statement at the [edit policy-options policy-statement *policy-name* term *term-name* from] hierarchy level and the install-nexthop lsp *lsp-name* statement at the [edit policy-options policy-statement *policy-name* term *term-name* then] hierarchy level. To apply the policy to the forwarding table, include the export *policy-name* statement at the [edit routing-options forwarding-table] hierarchy level.

```
[edit]
routing-options {
    autonomous-system 69;
    forwarding-table {
        export LSP-policy;
    }
    policy-options {
        policy-statement LSP-policy {
            term a {
                from community gold;
                then {
                    install-nexthop lsp pe1-to-pe2;
                    accept;
                }
            }
        }
        community gold members target:11111:1;
    }
}
```

Option: Configuring VPLS Multihoming with BGP Signaling

With VPLS multihoming, you can connect multiple PE router interfaces to one customer site. This feature provides VPLS redundancy should a PE router or PE router interface fail.

To configure multihoming, you must configure the same site IDs on all PE routers and router interfaces that are connected to the same customer site, You must also specify on each PE router which interfaces are connected to the customer site. We recommend that you configure distinct route distinguishers for each multihomed router. Configuring distinct route distinguishers helps with faster convergence when the connection to a primary router goes down. It also requires the other PE routers to maintain additional state information.

To configure a route distinguisher, include the route-distinguisher statement at the [edit routing-instances *instance-name*] hierarchy level. To assign a site ID, include the site-identifier statement at the [edit routing-instances *instance-name* protocols vpls site *name*] hierarchy level. To specify the interfaces associated with a site, include the interface statement at the [edit routing-instances *instance-name* protocols vpls site *name*] hierarchy level.

To connect multiple PE routers to one customer site, you must configure multihoming on each PE router connected to that site. This will prevent routing loops should BGP connectivity fail. BGP automatically determines the primary and backup routers. Alternatively, you can statically configure a primary PE router and backup PE routers for a customer site by specifying the preference value. BGP uses preference values to determine routing paths.

NOTE: Multihoming relies on full BGP connectivity to all other PEs. Configure a dual router reflector topology to provide redundant PE-to-PE BGP connectivity.

To configure multihoming, include the multi-homing statement at the [edit routing-instances *instance-name* protocols vpls site *name*] hierarchy level. To configure preference value, include the preference-value statement at the [edit routing-instances *instance-name* protocols vpls *site name*] hierarchy level. You can configure the preference value as primary or backup, or you can specify a preference number. When specifying preference numbers, configure the primary interface with a preference value of 65,535 and any backup interfaces with a number from 1 to 65,534.

When multiple PE router interfaces on a single PE router are connected to one customer site, you must configure an active interface. All traffic will pass through the active interface unless this interface fails, in which case a backup interface will become the active interface.

To specify a multihomed interface as the primary interface for a site, include the active-interface statement at the [edit routing-instances *instance-name* protocols vpls site *name*] hierarchy level. The interface that you specify is called the primary interface. If the primary interface goes down, an alternate interface becomes the active interface. Once the primary interface comes back up, the primary interface becomes the active interface once again and the alternate interface becomes inactive.

If you do not want to specify a primary multihomed interface, you can use the any option. With the any option, the router dynamically chooses an active interface. If the active interface goes down, an alternate interface becomes the active interface. Once the down interface comes back up, it stays inactive.

If no active interfaces are configured at the site level, it is assumed that all traffic for a VPLS site travels through a single, nonmultihomed PE router.

NOTE: If you add a direct connection between CE devices that are multihomed to the same VPLS site on different PE routers, traffic loops and loss of connectivity might occur. We do not recommend this topology.

The following example shows a multihoming configuration with two PE routers that are connected to a single customer site. Note in the configuration that PE1 is the primary router and PE2 is the backup router.

Router PE1

```
[edit]
routing-instances {
   green {
      instance-type vpls;
```

```
        interface fe-0/1/3.0;
        route-distinguisher 10.255.14.218:1;
        vrf-target target:11111:1;
        protocols {
            vpls {
                site-range 10;
                site green4 {
                    site-identifier 4;
                    multi-homing; # Ensures that BGP is established before forwarding on the
                        # site member interfaces.
                    preference value 65535;
                    interface fe-1/1/3.0;
                }
            }
        }
    }
}
```

Router PE2

```
[edit]
routing-instances {
    green {
        instance-type vpls;
        interface fe-0/1/0.0;
        route-distinguisher 10.255.14.219:1;
        vrf-target target:11111:1;
        protocols {
            vpls {
                site-range 10;
                site green4 {
                    site-identifier 4;
                    multi-homing;
                    preference value 1;
                    interface fe-0/1/0.0;
                }
            }
        }
    }
}
```

The following example shows a multihoming configuration with one PE router with multiple interfaces that are connected to a single customer site.

Router PE3

```
[edit]
routing-instances {
    green {
        instance-type vpls;
        interface fe-1/1/0.0;
        interface fe-1/2/0.0;
        interface fe-1/3/0.0;
        route-distinguisher 10.255.14.218:1;
        vrf-target target:11111:1;
        protocols {
            vpls {
                site-range 10;
                site green4 {
```

```
                site-identifier 4;
                active-interface any;
                interface fe-1/1/0.0;
                interface fe-1/2/0.0;
                interface fe-1/3/0.0;
            }
        }
    }
  }
}
```

For more information on VPLS multihoming, see the *JUNOS VPNs Configuration Guide*.

Option: Configuring VPLS Traffic Flooding over a Point-to-Multipoint LSP

In each VPLS routing instance, you can configure a dedicated point-to-multipoint LSP to carry all unknown unicast, broadcast, and multicast traffic. Enabling this feature increases the efficiency of your network because duplicate copies of flooded traffic do not have to be created for each PE router in the VPLS routing instance. Figure 74 on page 687 shows how flooded traffic reaches PE routers in a VPLS routing instance when a point-to-multipoint LSP is not configured for flooding. Figure 75 on page 688 shows an example of a VPLS routing instance configured with point-to-multipoint LSP flooding.

NOTE: You cannot configure point-to-multipoint LSP flooding if your VPLS network is configured for interoperability between BGP and LDP signaling.

Figure 74: Traditional Flooding in a VPLS Routing Instance

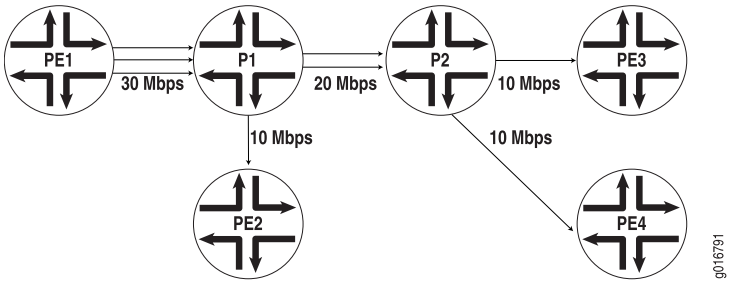
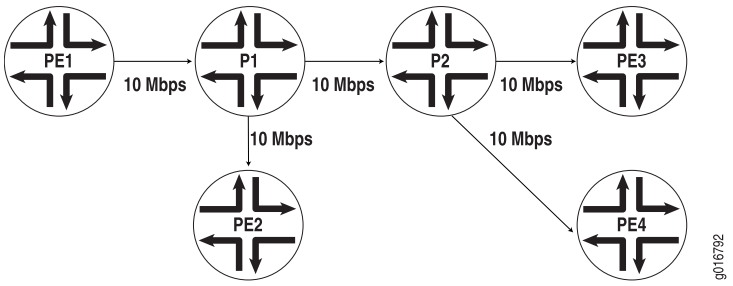


Figure 75: VPLS Routing Instance with Point-to-Multipoint LSP Flooding



You have three options when configuring a point-to-multipoint LSP for flooding:

- Static point-to-multipoint LSP—Configure this option to control which path each PE sub-LSP takes. When using this option, ensure that all PEs within the VPLS routing instance are part of the static point-to-multipoint LSP. When you add PEs to the VPLS routing instance, you must configure a sub-LSP for the new PE and add the sub-LSP to the static point-to-multipoint LSP. To configure a static point-to-multipoint LSP, include the label-switched-path *path-name* statement at the [edit protocols mpls] hierarchy level.
- Dynamic point-to-multipoint LSP with a preconfigured template—Configure this option to create a dynamic point-to-multipoint LSP with specific parameters such as link protection and optimized time. With this option, newly added PEs are automatically added to the point-to-multipoint LSP. To configure the preconfigured template, include the template statement at the [edit protocols mpls label-switch-path *path-name*] hierarchy level.
- Dynamic point-to-multipoint LSP with a default template—Configure this option to automatically create a dynamic point-to-multipoint LSP with default parameters. With this option, newly added PEs are automatically added to the point-to-multipoint LSP. To configure a default template, include the default-template statement at the [edit routing-instances *routing-instance-name* provider-tunnel rsvp-te label-switched-path-template] hierarchy level.

To define the parameters for a static point-to-multipoint LSP, include the label-switched-path *path-name* statement at the [edit protocols mpls] hierarchy level:

```
[edit]
protocols {
    mpls {
        label-switch-path vpls-bar-p2mp-s21_lsp_a {
            to 192.168.1.1
            p2mp vpls-bar-p2mp-lsp;
        }
        label-switched-path vpls-bar-p2mp-s21_lsp_b {
            to 192.168.1.2
            p2mp vpls-bar-p2mp-lsp;
        }
    }
}
```

To add a new PE router to the static p2mp LSP, include the label-switched-path *sub-path-name* statement at the [edit protocols mpls] hierarchy level:

```
[edit]
protocols {
   mpls {
      label-switched-path added-PE3 {
         to 1.1.1.1
         p2mp vpls-bar-p2mp-lsp;
      }
   }
}
```

For more information on configuring static and dynamic point-to-multipoint LSPs, see the *JUNOS MPLS Applications Configuration Guide*.

To enable this feature, configure either the static or label-switched-path-template options for the rsvp-te statement at the [edit routing-instance *routing-instance-name* provider-tunnel] hierarchy level:

```
[edit]
routing-instance foo {
   provider-tunnel {
      rsvp-te {
         static-lsp vpls-bar-p2mp-lsp;
      }
   }
}
```

To verify your work, enter the show vpls connection extensive command:

```
Router_1# show vpls connection extensive
....
status-vector: BF
connection-site Type St Time last up # Up trans
2 rmtUpJan 31 10:14:37 2007 1
Local interface: lsi.32768, Status: Up, Encapsulation: VPLS
Description: Intf -vpls VPLS-A local site 1 remote site 2
Remote PE: 10.255.164.2, Negotiated control-word: No
Incoming label: 262153, Outgoing label: 800000
RSVP-TE P2MP lsp:
Ingress branch LSP: 13:vpls:10.255.164.1:BPLS-A, State: Up
Egress branch LSP: 4:vpls:10.255.164.2:VPLS-A, Statue: Up
TimeEventInterface/Lb1/PE
Jan 31 10:14:37 2007 status update timer
Ingress RSVP-TE P2MP LSP: 11:vpls:10.255.164.1:VPLS-A, Flood next-hop ID: 476
```

Option: Configuring Automatic Site Selection

You can configure BGP-signaled VPLS instances to automatically specify the site IDs for the routers participating in the VPLS domain. Site IDs help to minimize label usage in VPLS instances with numerous PE routers.

The automatic-site-id statement includes the following options:

- startup-wait-time—Time to wait at startup to receive all VPLS information for configured route targets from other PE routers.
- new-site-wait-time—Time to wait to receive VPLS information from a newly configured routing instance or a new site. Effectively, it is the time to wait before a site makes an attempt to locate an unused site ID for its claim advertisement.
- collision-detect-time—Time to wait after issuing a claim advertisement before the PE router can start using the site ID if it does not receive a competing claim. If the PE router receives a competing claim within this time interval, it runs a collision resolution procedure. Explicitly configured site IDs always take precedence over automatically generated site IDs.
- reclaim-wait-time—Time to wait before attempting to claim a site ID after a collision. There are default values for all of these options, so they do not need to be explicitly configured.

To configure VPLS automatic site ID, include the automatic-site-id statement at the [edit routing-instances *routing-instance-name* protocols vpls site *site-name*] hierarchy level:

```
[edit]
routing-instances {
    vpls instance 1 {
        protocols {
            vpls {
                site vpls instance 1 {
                    automatic-site-id;
                }
            }
        }
    }
}
```

Option: Configuring VPLS to Use LSI Interfaces

On M-series and T-series routing platforms, VPLS uses tunnel-based PICs to create virtual ports on vt interfaces. If you do not have a tunnel-based PIC installed on your M-series or T-series routing platform, you can still configure VPLS by using label-switched interfaces (LSIs) to support the virtual ports. Use of LSI interfaces requires the use of Ethernet-based PICs installed in an Enhanced FPC.

NOTE: On MX-series routers, when using VPLS with an LSI interface, for traffic flowing from the core to the egress CE, classification does not work on the egress PE.

To use LSI interfaces for VPLS instead of vt interfaces, include the no-tunnel-services statement at the [edit routing-instances *instance-name* protocols vpls] hierarchy level.

```
[edit routing-instances]
instance-name  {
    protocols {
```

```
        vpls {
            no-tunnel-services;
        }
    }
}
```

NOTE: The following interface types do not support the use of LSI interfaces with VPLS:

- Aggregated SONET/SDH interfaces (cannot be used as the core-facing interface)
- Channelized interfaces (cannot be used as the core-facing interface)
- ATM1 interfaces

Option: Configuring Tunnel Services on MX-series Routers

MX-series routers use Dense Port Concentrators (DPCs) with built-in physical ports, which means that you do not insert PICs on the router. Instead, you configure tunnel interfaces on one of the four Packet Forwarding Engines (PFEs) that are on each DPC.

To create tunnel interfaces on a MX-series router, include the tunnel-services statement at the [edit chassis fpc *slot-number* pic *number*] hierarchy level. To configure the bandwidth for a tunnel interface, include the bandwidth statement at the [edit chassis fpc *slot-number* pic *number*] hierarchy level.

The following example shows a tunnel interface with 1 Gbps of bandwidth configured on PFE 1 of the DPC installed in slot 4 of an MX-series router:

```
[edit chassis]
fpc 4;
pic 1 {
    tunnel services {
        bandwidth 1g;
    }
}
```

Once you have configured a tunnel interface on a PFE, you can treat this interface as a standard tunnel interface and proceed with a standard VPLS configuration. For more information, see the *JUNOS System Basics Configuration Guide*.

Configuring Integrated Routing and Bridging in a VPLS Instance (MX-series Routers Only)

Integrated routing and bridging (IRB) over VPLS cannot be used in conjunction with the vlan-id all statement. One or more Layer 2 logical interfaces must be configured inside the instance in order for IRB to function properly.

To configure IRB within a VPLS instance, include the routing-interface *irb-interface-name* statement at the [edit routing-instances *routing-instance-name* instance-type vpls] hierarchy level:

```
[edit]
routing-instances {
    marketing {
        instance-type vpls;
        route-distinguisher 11.11.11.11:10;
        vrf-target target:100:100;
        interface ae0.100;
        interface ae0.200;
        routing-interface irb.1234;
    }
}
```

Configuring VLAN IDs in a VPLS Instance (MX-series Routers Only)

You can configure VLAN identifiers for a VPLS instance in the following ways:

- By using the input-vlan-map and the output-vlan-map statements at the [edit interfaces] hierarchy level. For more information, see the *JUNOS Network Interfaces Configuration Guide* and *JUNOS Class of Service Configuration Guide*.
- By using the vlan-id or vlan-tags statements at the [edit routing-instances *routing-instance-name* instance-type vpls] hierarchy level.

The vlan-id and vlan-tags statements are used to perform the following functions:

- Translate, or normalize, the VLAN tags of received packets received into a learn VLAN identifier.
- Create multiple learning domains that each contain a learn VLAN identifier. A learning domain is a MAC address database to which MAC addresses are added based on the learn VLAN identifier.

For more information about how VLAN tags are processed and translated, see the *JUNOS MX-series Layer 2 Configuration Guide*.

To configure VLAN identifiers for a VPLS instance, include the vlan-id or vlan-tags statement at the [edit routing-instances *routing-instance-name* instance-type vpls] hierarchy level.

NOTE: You cannot configure VLAN mapping using the input-vlan-map and output-vlan-map statements if you configure a learn VLAN identifier for a VPLS instance using the vlan-id or vlan-tags statements.

```
[edit]
routing-instances {
    marketing {
        instance-type vpls;
        vlan-id 401;
```

```
        route-distinguisher 11.11.11.11:10;
        vrf-target target:100:100;
        interface ae0.100;
        interface ae0.200;
    }
}
```

Defining a VPLS Firewall Policer

You can configure filters, policers, and broadcast/unknown filters to determine which kind of traffic is allowed into and out of a VPLS domain. You can apply these filters and policers to CE-facing interfaces only.

To process traffic as it enters a VPLS domain, you can define a firewall policer and apply it to the input interface. To define policer characteristics for incoming VPLS traffic, include the bandwidth-limit and burst-size-limit statements at the [edit firewall policer *policer-name* if-exceeding] hierarchy level. Then, specify statements to implement the desired action (for example, discard) for the policed traffic at the [edit firewall policer *policer-name* then] hierarchy level. To apply the policer to a CE-facing interface, include the input or output statements and the name of the policer at the [edit interfaces *interface-name* unit *unit-number* family vpls policer] hierarchy level.

```
[edit]
interfaces {
    ge-2/1/0 {
        vlan-tagging;
        mtu 1544;
        encapsulation vlan-vpls;
        unit 0 {
            encapsulation vlan-vpls;
            vlan-id 600;
            family vpls {
                policer {
                    input vpls-policer;
                }
            }
        }
    }
}
firewall {
    policer {
        vpls-policer {
            if-exceeding {
                bandwidth-limit 5m;
                burst-size-limit 1m;
            }
            then discard;
        }
    }
}
```

Defining a VPLS Firewall Filter

You can configure filters, policers, and broadcast/unknown filters to determine which kind of traffic is allowed into and out of a VPLS domain. You can apply these filters and policers to CE-facing interfaces only.

To process traffic as it exits a VPLS domain, you can define a firewall filter and apply it to the output interface. To configure match conditions for a firewall filter, include the interface-group, source-mac-address, destination-mac-address, ethernet-type, or vlan-ethernet-type statements at the [edit firewall family vpls filter *filter-name* term *term-name* from] hierarchy level. Then, implement the desired action (for example, discard) for the traffic at the [edit firewall family vpls filter *filter-name* term *term-name* then] hierarchy level. To apply the filter to a CE-facing interface, include the input, output, or group statements at the [edit interfaces *interface-name* unit *unit-number* family vpls filter] hierarchy level.

```
[edit]
interfaces {
    fe-2/1/1 {
        vlan-tagging;
        mtu 1544;
        encapsulation vlan-vpls;
        unit 0 {
            encapsulation vlan-vpls;
            vlan-id 600;
            family vpls {
                filter {
                    output vpls-out-filter;
                }
            }
        }
    }
}
firewall {
    family vpls {
        filter vpls-out-filter {
            interface-specific;
            term 1 {
                from {
                    source-mac-address {
                        00.10.10.10.11.18/48;
                    }
                }
                then {
                    count count.ce2;
                    accept;
                }
            }
            term 2 {
                then accept;
            }
        }
    }
```

```
}
```

NOTE:

- Output filters do not work for broadcast, multicast, and unknown unicast traffic.
- If an IRB interface is configured as part of a VPLS routing instance, VPLS filters might not filter packets that are destined to the IRB interface. This can be configured by installing filters that match Layer 3 fields for the the IRB interface.
- If you apply a firewall filter to discard a source MAC address, the MAC address is not deleted from the MAC address table.

Restricting Broadcast Packets in VPLS

You can configure filters, policers, and broadcast/unknown filters to determine which kind of traffic is allowed into and out of a VPLS domain. You can apply these filters and policers to CE-facing interfaces only.

To restrict the flow of broadcast and unknown unicast packets into a VPLS domain, you must create a firewall filter and apply the filter to one of the forwarding tables of the VPLS routing instance. When you apply a filter in this way, the filter processes traffic from all interfaces in the instance, including vt interfaces. To configure match conditions for a VPLS-based firewall filter, include the source-mac-address, destination-mac-address, interface-group, ethernet-type, or vlan-ethernet-type statements at the [edit firewall family vpls filter *filter-name* term *term-name* from] hierarchy level. Then, specify statements to activate the desired action (for example, discard) for the matched packets at the [edit firewall family vpls filter *filter-name* term *term-name* then] hierarchy level.

To apply the filter to the broadcast and unknown unicast table of a VPLS routing instance, include the input statement and the name of the filter at the [edit routing-instances *instance-name* forwarding-options family vpls flood] hierarchy level. To apply the filter to the destination MAC address table of a VPLS routing instance, include the input statement and the name of the filter at the [edit routing-instances *instance-name* forwarding-options family vpls filter] hierarchy level.

```
[edit]
firewall {
   family vpls {
      filter vpls-flood {
         term 1 {
            from {
               destination-mac-address (broadcast | multicast | unknown-unicast) {
                  # The broadcast, multicast,
                  # and unknown-unicast options apply to MX-series
                  # routers only.
                  00.90.69.dc.95.3b/48;
               }
            }
            then discard;
         }
```

```
            term 2 {
                then accept;
            }
        }
    }
}
routing-instances {
    green {
        forwarding-options {
            family vpls {
                (flood | filter) {
                    input vpls-flood;
                }
            }
        }
    }
}
```

When you configure VPLS, a priority filter for Spanning Tree Protocol (STP) bridge protocol data units (BPDUs) is enabled by default. This BPDU filter matches on the well-known STP MAC address of 01:80:c2:00:00:00/24 and applies high priority to this traffic.

For more information on VPLS policers and filters, see the *JUNOS Policy Framework Configuration Guide* and the *JUNOS VPNs Configuration Guide.*

Option: Enabling VPLS Class of Service

For JUNOS Release 6.2 or later, you can configure class of service (CoS) for all interfaces in the VPLS domain. CoS information is sent across the MPLS backbone and is preserved for all VPLS traffic processed by local interfaces, virtual ports, and remote interfaces.

For more information on configuring CoS, see the *JUNOS Class of Service Configuration Guide*.

Option: Enabling VPLS Graceful Restart

VPLS graceful restart allows you to continue forwarding VPLS traffic across the core MPLS network even if one of the routers in the forwarding path restarts. Graceful restart for VPLS functions the same way as Layer 2 VPN graceful restart. To configure graceful restart for VPLS, include the graceful-restart statement at the [edit routing-options] hierarchy level on all PE and core routers.

```
[edit]
routing-options {
    graceful-restart;
}
```

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

Configuring the VPLS MAC Address Timeout

You can fine-tune your VPLS domain by clearing MAC address entries from the VPLS table or modifying the default timeout interval for the VPLS table.

NOTE: On MX-series routers running JUNOS Release 8.4 and later, you can set the expiration time of entries in the MAC table only for the entire router, not for specific VPLS routing instances. To set the expiration for the entire router, include the mac-table-aging-time seconds statement at the [edit protocols l2-learning] hierarchy level. Do not include the mac-table-aging-time statement at the [edit routing-instances routing-instance-name protocols vpls] hierarchy level on MX-series routers running JUNOS Release 8.4 and later.

To clear all MAC address entries from the VPLS table, issue the clear vpls mac-address command. Add the logical-system *logical-system-name* option to clear entries within a logical system and include the instance *instance-name* option to clear entries in a specific VPLS instance. Use the *mac-address* option to remove individual MAC addresses.

To configure the VPLS table timeout interval, include the mac-table-aging-time statement at the [edit routing-instances *instance-name* protocols vpls] hierarchy level. The default interval is 300 seconds, with a minimum of 10 seconds and a maximum of 1 million seconds. As a general rule, you can configure longer values for small, stable VPLS networks and shorter values for large, dynamic VPLS networks. If no traffic is received for a specific MAC, M-series and T-series routers wait one additional interval before automatically clearing MAC address entries from the VPLS table. MX-series routers do not wait this interval.

```
[edit]
routing-instances {
   instance-name {
      protocols {
         vpls {
            mac-table-aging-time seconds;
         }
      }
   }
}
```

Option: Configuring VPLS Interinstance Bridging and Routing

To deliver interinstance traffic between two or more VPLS instances, or between a VPLS instance and a Layer 3 VPN routing instance, you must use a logical tunnel interface. Originally designed to interconnect logical systems, the logical tunnel interface acts as a point-to-point connection between instances. A logical tunnel interface can be generated by a Tunnel Services PIC installed on an Enhanced FPC in your routing platform, an integrated Adaptive Services Module installed in an M7i router, or a tunnel services interface configured on MX-series routers. To configure a logical tunnel interface, include the lt-*fpc/pic/*0 statement at the [edit interfaces] hierarchy level. Keep in mind these rules when you connect instances:

- You need to configure both endpoints of the logical tunnel. Configure the first logical tunnel interface in the VPLS instance and the second within the instance you want to interconnect to the VPLS domain.
- Choose one of several interface encapsulation types for your logical tunnel interface peers. Your choices are Ethernet, Ethernet circuit cross-connect (CCC), Ethernet VPLS, Frame Relay, Frame Relay CCC, VLAN, VLAN CCC, and VLAN VPLS. Include one of these choices with the encapsulation statement at the [edit interfaces *lt-fpc/pic/*0 unit *unit-number*] hierarchy level.
- Depending on the encapsulation type you select, specify a corresponding data-link connection identifier (DLCI) number for Frame Relay or a VLAN identifier for VLAN encapsulations on your logical tunnel interface peers. To configure, include the dlci or vlan-id statement at the [edit interfaces lt-*fpc/pic/*0 unit *unit-number*] hierarchy level.
- Your choice of protocol family for the logical tunnel interface also is determined by your selection of an encapsulation type. For Ethernet VPLS and VLAN VPLS, family vpls is assigned by default. For all other Ethernet and VLAN encapsulation types, include the mpls or inet statement at the [edit interfaces lt-*fpc/pic/*0 unit *unit-number* family] hierarchy level. For Frame Relay encapsulation types, you can configure any of the available protocol families: ccc, inet, inet6, iso, mpls, or tcc.
- Be sure to match the logical interface unit numbers of the peering logical tunnel interfaces. To configure, include the peer-unit statement at the [edit interfaces lt-*fpc/pic/*0 unit *unit-number*] hierarchy level.

```
[edit]
interfaces {
   lt-fpc/pic/0 {
      unit unit-number {
         encapsulation (ethernet | ethernet-ccc | ethernet-vpls | frame-relay |
            frame-relay-ccc | vlan | vlan-ccc | vlan-vpls);
         peer-unit number; # The logical unit number of the peering lt interface.
         dlci dlci-number;
         vlan-id vlan-number;
         family (ccc | inet | inet6 | iso | mpls | tcc);
      }
   }
}
routing-instances {
```

```
    vpls-instance-name {
        interface ge-fpc/pic/port.unit-number;
        interface lt-0/0/0.1;
        ...
        second-instance-name {
            interface at-fpc pic/port.unit-number;
            interface lt-0/0/0.2;
            ...
        }
    }
}
```

Option: Selecting Interfaces to Process VPLS Traffic

On M-series and T-series routing platforms, the PICs that can create VPLS virtual ports dynamically from vt interfaces include the Tunnel Services PIC, the Link Services PIC, and the Adaptive Services PIC. On MX-series routers, logical tunnel interfaces configured by including the tunnel-services statement at the [edit chassis fpc *slot-number pic number*] hierarchy level can create VPLS virtual ports dynamically from vt interfaces.

By default, the JUNOS software automatically and randomly selects vt interfaces to act as VPLS virtual ports in a round-robin fashion. However, if your routing platform contains two or more of these tunnel-enabled interfaces, you can manually select which interfaces process traffic for each VPLS domain.

You can select an interface to be the primary device responsible for VPLS traffic processing. You can also select a group of interfaces to share responsibility for VPLS traffic processing. When the primary interface is operating normally, it handles all VPLS-related tasks. If the primary device is not available, any interfaces included in the VPLS interface group assume responsibility.

To select an interface to be the primary device responsible for VPLS traffic processing, include the primary statement at the [edit routing-instances *instance-name* protocols vpls tunnel-services] hierarchy level. To select a group of interfaces to share responsibility for VPLS traffic processing, include the devices statement at the [edit routing-instances *instance-name* protocols vpls tunnel-services] hierarchy level.

```
[edit]
routing-instances {
    instance-name {
        protocols {
            vpls {
                tunnel-services {
                    devices [vt-0/0/0 vt-1/0/0 vt-2/0/0];
                    primary vt-0/0/0;
                }
            }
        }
    }
}
```

Option: Limiting the Number of MAC Addresses Learned on an Interface

There are three main levels where you can configure MAC address limits:

- interface-mac-limit—This statement allows you to specify a limit for MAC addresses at an interface level. For VPLS, you can include the interface-mac-limit statement at the [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls], [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls site *site-name* interfaces *interface-name*], [edit routing-instances *routing-instance-name* protocols vpls], or [edit routing-instances *routing-instance-name* protocols vpls site *site-name* interfaces *interface-name*] hierarchy level. For MX-series routers only, you can specify what the router does with additional MACs once the MAC limit is reached. The default behavior is for the router to flood the MACs, but you can alternatively include the packet-action drop option to have the router drop the MACs. The default MAC address table size for each interface is 1024 addresses.
- mac-table-size—This statement allows you to specify a limit for MAC addresses at a domain level. For VPLS, you can include the mac-table-size statement at the [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls] or [edit routing-instances *routing-instance-name* protocols vpls] hierarchy level. The default MAC address table size for each domain is 5120 addresses.
- global-mac-limit (MX-series routers only)—This statement allows you to specify a limit for MAC addresses for all interfaces and all domains for the entire router. You can include the global-mac-limit statement at the [edit protocols l2-learning] hierarchy level. The default MAC address table size for the entire system is 393,215 addresses.

NOTE: If you manually configure a MAC address limit, you must ensure that values for interface limits (such as the interface-mac-limit) are set lower than domain limits (such as mac-table-size), and the domain limits are set lower than global limits (such as global-mac-limit). If a value for a more specific limit is set higher than a more global limit, the commit will fail.

The range of values for the interface-mac-limit statement is 16 through 65,536. The output of the show vpls statistics command displays the results of configuring interface-level MAC address limitations.

```
[edit]
routing-instances {
   instance-name {
      protocols {
         vpls {
            interface-mac-limit number;
            site site-name {
               interface interface-name {
                  interface-mac-limit number;
               }
            }
```

```
        }
      }
    }
  }
```

Option: Optimizing VPLS Traffic Flows

To improve the performance of VPLS traffic processing in your routing platform, you can implement the following features:

- To optimize VPLS traffic flows across multiple paths, you can enable per-packet load balancing. To configure, include the load-balance per-packet statement at the [edit policy-options policy-statement *policy-name* term *term-name* then] hierarchy level and apply the policy to the forwarding table with the export *policy-name* statement at the [edit routing-options forwarding-table] hierarchy level.
- To optimize hashing of source and destination MAC addresses within VPLS traffic flows, include the source-mac and destination-mac statements at the [edit forwarding-options hash-key family multiservice] hierarchy level.

For more information on load balancing and hash keys, see the *JUNOS Policy Framework Configuration Guide*.

Option: Aggregated Interfaces for VPLS

You can configure aggregated Ethernet interfaces between CE devices and PE routers for VPLS routing instances. Traffic is load balanced across all of the links in the aggregated interface. If one or more links in the aggregated interface fails, the traffic is switched to the remaining links.

In the example below, 0 is the interface instance number that completes the link association. This number can be from 0 through 127, for a total of 128 aggregated interfaces. The VPLS encapsulation types supported on aggregated Ethernet interfaces are ethernet-vpls, vlan-vpls, or extended-vlan-vpls.

```
[edit]
interfaces ae0
vlan-tagging;
encapsulation vlan-vpls;
unit 0 {
   vlan-id 100;
}
```

The aggregated Ethernet interface must also be configured for a VPLS routing instance. Use the standard VPLS routing instance configuration on aggregated Ethernet interfaces.

For more information on how to configure aggregated Ethernet interfaces, see the *JUNOS Network Interfaces Configuration Guide*.

Option: Configuring VPLS Graceful Routing Engine Switchover

Graceful Routing Engine switchover (GRES) allows a routing platform with dual Routing Engines to switch over from a master Routing Engine to a backup Routing Engine without causing an interruption to packet forwarding. Graceful Routing Engine switchover is supported with VPLS, meaning that should the master Routing Engine go down, the router can switch to the backup Routing Engine without affecting VPLS traffic.

To configure graceful Routing Engine switchover, include the graceful-switchover statement at the [edit chassis redundancy] hierarchy level. To disable graceful Routing Engine switchover, remove the graceful-switchover statement at the [edit chassis redundancy] hierarchy level.

```
[edit]
chassis redundancy {
   graceful-switchover;
}
```

Option: Configuring VPLS Nonstop Active Routing

Nonstop active routing (NSR) enables a routing platform with redundant Routing Engines to switch from a primary Routing Engine to a backup Routing Engine without alerting peer nodes that a change has occurred. Nonstop active routing uses the same infrastructure as graceful Routing Engine switchover to preserve interface and kernel information. However, nonstop active routing also preserves routing information and protocol sessions by running the routing protocol process (rpd) on both Routing Engines. The logical interface, next-hop router, and both advertised and received labels are preserved. In addition, nonstop active routing preserves TCP connections maintained in the kernel.

Configuring Nonstop Active Routing

Nonstop active routing requires you to configure graceful Routing Engine switchover (GRES). To enable graceful Routing Engine switchover, include the graceful-switchover statement at the [edit chassis redundancy] hierarchy level:

```
[edit chassis redundancy]
graceful-switchover;
```

By default, nonstop active routing is disabled. To enable nonstop active routing, include the nonstop-routing statement at the [edit routing-options] hierarchy level:

```
[edit routing-options]
nonstop-routing;
```

To disable nonstop active routing, remove the nonstop-routing statement from the [edit routing-options] hierarchy level.

To enable the routing platform to switch over to the backup Routing Engine when the routing protocol process (rpd) fails rapidly three times in succession, include the other-routing-engine statement at the [edit system processes routing failover] hierarchy level.

The following example enables graceful Routing Engine switchover, nonstop active routing, and nonstop active routing trace options for VPLS.

```
[edit]
system commit {
   synchronize;
}
chassis {
   redundancy {
      graceful-switchover; # This enables graceful Routing Engine switchover on the
         # routing platform.
   }
}
routing-options {
   nonstop-routing; # This enables nonstop active routing on the routing platform.
   traceoptions {
      flag nsr-synchronization;
   }
}
```

For more information about the other-routing-engine statement, see the *JUNOS System Basics Configuration Guide*.

Synchronizing the Routing Engine Configuration

When you configure nonstop active routing, you must also include the commit synchronize statement at the [edit system] hierarchy level so that configuration changes are synchronized on both Routing Engines:

```
[edit system]
commit synchronize;
```

If you try to commit the nonstop active routing configuration without including the commit synchronize statement, the commit operation fails.

If you issue the commit synchronize command at the [edit] hierarchy level on the backup Routing Engine, the JUNOS system software displays a warning and commits the candidate configuration.

NOTE: A newly inserted backup Routing Engine automatically synchronizes its configuration with the master Routing Engine configuration.

When you configure nonstop active routing, you can bring the backup Routing Engine online after the master Routing Engine is already running. There is no requirement to start the two Routing Engines simultaneously.

Verifying VPLS Nonstop Active Routing Operation

To see whether or not nonstop active routing is enabled, issue the show task replication command.

NOTE: You must issue the show task replication command on the master Routing Engine. This command is not supported on the backup Routing Engine.

For more information on this command, see the *JUNOS System Basics and Services Command Reference*.

Tracing VPLS Nonstop Active Routing Synchronization Events

To trace the label and logical interface association that VPLS receives from the kernel replication state, include the nsr-synchronization statement at the [edit routing-options traceoptions flag] hierarchy level. This flag also traces the Layer 2 VPN signaling state replicated from routes advertised by BGP.

```
[edit routing-options]
traceoptions {
   flag nsr-synchronization;
}
```

Option: Configuring the Spanning Tree Protocol and VPLS on MX-series Routers

If multiple routers on a customer site are connected the same PE, you should configure a version of the Spanning Tree Protocol on that PE. To configure RSTP or MSTP and VPLS simultaneously, include the rstp or mstp statement at the [edit instance-type layer2–control] hierarchy level:

```
[edit]
instance-type layer2-control;
protocols {
   rstp {
      interface interface name;
      force-version stp; # To run STP instead of RSTP
   }
}
```

The Per-VLAN Spanning Tree (PVST) protocol maintains a separate spanning-tree instance for each VLAN. To enable PVST for a specific VLAN ID, there should be a VPLS instance with that VLAN ID and all of the logical interfaces assigned to that instance should have the same matching VLAN ID. To configure PVST with VPLS, include the vstp statement at the [edit instance-type layer2-control] hierarchy level:

```
[edit]
instance-type layer2-control;
protocols {
   vstp {
      interface interface name;
      vlan vlan-id;
   }
}
```

If you only want STP to run on a device, you can configure STP by including the force-version stp statement at the [edit protocols rstp] or [edit protocols vstp] hierarchy level:

```
[edit]
protocols {
   rstp {
      force-version stp;
   }
}
```

For more information about the Spanning Tree Protocol (VSTP, MSTP, RSTP, or STP), see the *MX-series Solutions Guide* and the *JUNOS Routing Protocols Configuration Guide*.

Filtering Layer 2 Packets in a VPLS Instance (MX-series Routers Only)

You can match the learn-vlan-id, user-vlan-id, and traffic-type terms for a VPLS instance on the MX-series platform. Packets entering or exiting the VPLS instance have a single VLAN tag. This VLAN tag is the same as what was received from the network. This VLAN tag corresponds to the one VLAN ID on a singly tagged logical interface or inner VLAN tag for the doubly tagged logical interface. The VLAN ID is used to qualify learned MAC addresses.

To configure a firewall filter for a VPLS instance, specify the conditions that the packet must match at the [edit firewall family vpls filter *filter-name* term *term-name* from] hierarchy level. To apply a firewall filter to a VPLS routing instance, include the input filter-name statement at [edit routing-instances *routing-instance-name* forwarding-options family vpls filter] hierarchy level. For more information, see the *JUNOS Policy Framework Configuration Guide*.

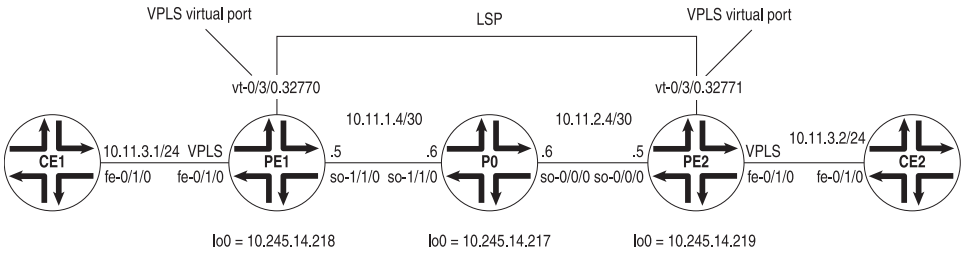
VPLS Configuration Examples

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

- Example: VPLS Configuration (BGP Signaling) on page 706

Example: VPLS Configuration (BGP Signaling)

Figure 76: VPLS Topology Diagram



VPLS table for PE1			
Interface	MAC Address	Outgoing label	Received label
fe-0/1/0	aaaa.aaaa.aaaa	n/a	n/a
vt-0/3/0.32770	bbbb.bbbb.bbbb	800000	800002

g017142

In Figure 76 on page 706, a simple VPLS topology is enabled between routers PE1 and PE2. CE routers CE1 and CE2 use Ethernet-based interfaces to connect VLAN 600 to their local PE router. The PE routers PE1 and PE2 are connected to one another by LSPs enabled across a service provider backbone running MPLS, BGP, RSVP, and OSPF.

In a VPLS routing instance named green, PE1 has a local interface fe-0/1/0 and a virtual port of vt-0/3/0.32770 (the virtual port is created dynamically on the Tunnel Services PIC when VPLS is configured). PE2 has a local interface fe-0/1/0 and a virtual port of vt-0/3/0.32771 in the same green instance. As a result, routers CE1 and CE2 can send Ethernet traffic to one another as if they are physically connected to each other on a LAN.

On Router CE1, the only item you need to configure is the Fast Ethernet interface that connects to PE1. Be sure to write down the VLAN identifier and IP address, so you can match them later on CE2.

Router CE1

```
[edit]
interfaces {
   fe-0/1/0 {
      vlan-tagging;# Configure VLAN tagging for VLAN VPLS or extended VLAN VPLS.
      unit 0 {
         vlan-id 600;# The Ethernet interface on CE2 must use the same VLAN ID.
         family inet {
            address 10.11.3.1/24; # The interface on CE2 must use the same prefix.
         }
      }
   }
}
```

On Router PE1, prepare the router for VPLS by configuring BGP, MPLS, OSPF, and RSVP. (These protocols are the basis for most Layer 2 VPN-related applications,

including VPLS.) Include the signaling statement at the [edit protocols bgp group *group-name* family l2vpn] hierarchy level, because VPLS uses the same infrastructure for internal BGP as Layer 2 VPNs.

NOTE: In JUNOS Release 7.3 and later, the signaling statement replaces the unicast statement at the [edit protocols bgp group *group-name* family l2vpn] hierarchy level. You must use the signaling statement if you wish to configure VPLS domains and Layer 2 VPNs simultaneously.

Next, configure VLAN tagging on the Fast Ethernet interface connected to Router CE1. Include VLAN VPLS encapsulation at both the physical and logical interface levels. Be sure to use the same VLAN ID for all Ethernet interfaces that are part of a single VPLS instance. Finally, add the Fast Ethernet interface into a VPLS routing instance and specify the site range, site ID number, and site name.

Router PE1

```
[edit]
interfaces {
   fe-0/1/0 {
      vlan-tagging;# Configure VLAN tagging for VLAN VPLS or extended VLAN VPLS.
      encapsulation vlan-vpls; # Configure VPLS encapsulation on both the
      unit 0 { # physical interface and the logical interface.
         encapsulation vlan-vpls;
         vlan-id 600;# The VLAN ID is the same one used by the CE routers.
      }
   }
   so-1/1/0 {
      unit 0 {
         family inet {
            address 10.11.1.5/30;
         }
         family mpls;
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.245.14.218/32;
         }
      }
   }
}
routing-options {
   autonomous-system 69;
   forwarding-table {
      export exp-to-fwd;# Applies a policy that selects an LSP for the VPLS instance.
   }
}
protocols {
   rsvp {
      interface all {
         aggregate;
      }
   }
```

```
    mpls {
      label-switched-path pe1-to-pe2 { # Configure an LSP to reach other VPLS PEs.
        to 10.245.14.219;
      }
      interface all;
    }
    bgp {
      group vpls-pe {
        type internal;
        local-address 10.245.14.218;
        family l2vpn { # VPLS uses the same infrastructure as Layer 2 VPNs
          signaling; # for internal BGP.
        }
        neighbor 10.245.14.217;
        neighbor 10.245.14.219;
      }
    }
    ospf {
      traffic-engineering;
      area 0.0.0.0 {
        interface so-1/1/0.0 {
          metric 11;
        }
        interface lo0.0 {
          passive;
        }
      }
    }
}
policy-options {
    policy-statement exp-to-fwd {
      term a {
        from community grn-com; # Matches the community in the VPLS instance.
        then {
          install-nexthop lsp pe1-to-pe2; # If there are multiple LSPs that exist
          accept; # between VPLS PE routers, this statement sends VPLS traffic
        }
      }
    }
    community grn-com members target:11111:1; # Adds the instance to a BGP
      community.
}
routing-instances {
    green {
      instance-type vpls; # Configure a VPLS routing instance.
      interface fe-0/1/0.0;
      route-distinguisher 10.245.14.218:1;
      vrf-target target:11111:1; # This value is important to the BGP community.
      protocols {
        vpls {# Configure a VPLS site range, site name, and site identifier.
          site-range 10;
          site greenPE1 {
            site-identifier 1;
          }
        }
      }
```

```
    }
  }
```

On Router P0, configure BGP, MPLS, OSPF, and RSVP to interconnect PE1 and PE2.

Router P0

```
[edit]
interfaces {
  so-0/0/0 {
    unit 0 {
      family inet {
        address 10.11.2.6/30;
      }
      family mpls;
    }
  }
  so-1/1/0 {
    unit 0 {
      family inet {
        address 10.11.1.6/30;
      }
      family mpls;
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.245.14.217/32;
      }
    }
  }
}
protocols {
  rsvp {
    interface all {
      aggregate;
    }
  }
  mpls {
    interface all;
  }
  bgp {
    group vpls-pe {
      type internal;
      local-address 10.245.14.217;
      family l2vpn { # VPLS uses the same infrastructure as Layer 2 VPNs
        signaling; #for internal BGP.
      }
      neighbor 10.245.14.218;
      neighbor 10.245.14.219;
    }
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface so-1/1/0.0 {
        metric 11;
```

```
        }
        interface so-0/0/0.0 {
            metric 15;
        }
        interface lo0.0 {
            passive;
        }
    }
  }
}
```

On Router PE2, configure BGP, MPLS, OSPF, and RSVP to complement the configuration on PE1. Next, configure VLAN tagging on the Fast Ethernet interface connected to Router CE2. Include VLAN VPLS encapsulation at both the physical and logical interface levels. Be sure to use the same VLAN ID for all Ethernet interfaces that are part of a single VPLS instance. Finally, add the Fast Ethernet interface into a VPLS routing instance and specify the site range, site ID number, and site name.

Router PE2

```
[edit]
interfaces {
  fe-0/1/0 {
    vlan-tagging; # Configure VLAN tagging for VLAN VPLS or extended VLAN VPLS.
    encapsulation vlan-vpls; # Configure VPLS encapsulation on both the
    unit 0 { # physical interface and logical interface.
      encapsulation vlan-vpls;
      vlan-id 600;# The VLAN ID is the same one used by the CE routers.
    }
  }
  so-0/0/0 {
    unit 0 {
      family inet {
        address 10.11.2.5/30;
      }
      family mpls;
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.245.14.219/32;
      }
    }
  }
}
routing-options {
  autonomous-system 69;
  forwarding-table {
    export exp-to-fwd; # Applies a policy that selects an LSP for the VPLS instance.
  }
}
protocols {
  rsvp {
    interface all {
      aggregate;
    }
```

```
    }
    mpls {
      label-switched-path pe2-to-pe1 { # Configure an LSP to other VPLS PE routers.
        to 10.245.14.218;
      }
      interface all;
    }
    bgp {
      group vpls-pe {
        type internal;
        local-address 10.245.14.219;
        family l2vpn { # VPLS uses the same infrastructure as Layer 2 VPNs
          signaling; # for internal BGP.
        }
        neighbor 10.245.14.217;
        neighbor 10.245.14.218;
      }
    }
    ospf {
      traffic-engineering;
      area 0.0.0.0 {
        interface so-0/0/0.0 {
          metric 15;
        }
        interface lo0.0 {
          passive;
        }
      }
    }
}
policy-options {
    policy-statement exp-to-fwd {
      term a {
        from community grn-com; # Matches the community with the VPLS instance.
        then {
          install-nexthop lsp pe2-to-pe1; # If there are multiple LSPs that exist
          accept; # between VPLS PE routers, this statement sends VPLS traffic
        }
      }
    }
    community grn-com members target:11111:1; # This adds the instance into a BGP
      community.
}
routing-instances {
    green {
      instance-type vpls; # Configure a VPLS routing instance.
      interface fe-0/1/0.0;
      route-distinguisher 10.245.14.219:1;
      vrf-target target:11111:1; # This value is important for the BGP community.
      protocols {
        vpls { # Configure a VPLS site range, site name, and site identifier.
          site-range 10;
          site greenPE2 {
            site-identifier 2;
          }
        }
```

```
        }
    }
}
```

On Router CE2, complete your VPLS network by configuring the Fast Ethernet interface that connects to PE2. Use the same VLAN identifier and IP address prefix used on Router CE1.

Router CE2

```
[edit]
interfaces {
    fe-0/1/0 {
        vlan-tagging; # Configure VLAN tagging for VLAN VPLS or extended VLAN VPLS.
        unit 0 {
            vlan-id 600; # The Ethernet interface on CE1 must use the same VLAN ID.
            family inet {
                address 10.11.3.2/24; # The interface on CE1 must use the same prefix.
            }
        }
    }
}
```

Verifying Your Work

To verify proper operation of VPLS, use the following commands:

- clear vpls mac-address instance *instance-name*
- show interfaces terse
- show route forwarding-table family mpls
- show route forwarding-table family vpls (destination | extensive | matching | table)
- show route instance (detail)
- show system statistics vpls
- show vpls connections
- show vpls statistics

The following section shows the output of these commands on Router PE1 as a result of the configuration example:

```
user@PE1> show interfaces terse
Interface               Admin Link Proto Local                 Remote
so-1/1/0                up    up
so-1/1/0.0              up    up   inet  10.11.1.5/30
                                  mpls
so-1/1/1                up    up
so-1/1/2                up    up
so-1/1/3                up    up
fe-0/1/0                up    up
fe-0/1/0.0              up    up   vpls   # This is the local Fast Ethernet
# interface.
fe-0/1/1                up    up
fe-0/1/2                up    up
```

```
fe-0/1/3                up    up
gr-0/3/0                up    up
ip-0/3/0                up    up
mt-0/3/0                up    up
pd-0/3/0                up    up
pe-0/3/0                up    up
vt-0/3/0                up    up
vt-0/3/0.32770          up    up  # This is the dynamically generated virtual
port.
dsc                     up    up
fxp0                    up    up
fxp0.0                  up    up   inet  192.186.14.218/24
fxp1                    up    up
fxp1.0                  up    up   tnp   4
gre                     up    up
ipip                    up    up
lo0                     up    up
lo0.0                   up    up   inet  10.245.14.218       --> 0/0
                                         127.0.0.1           --> 0/0
                                   inet6 fe80::2a0:a5ff:fe28:13e0
                                         feee::10:245:14:218
lsi                     up    up
mtun                    up    up
pimd                    up    up
pime                    up    up
tap                     up    up

user@PE1> show system statistics vpls
vpls:
        0 total packets received
        0 with size smaller than minimum
        0 with incorrect version number
        0 packets for this host
        0 packets with no logical interface
        0 packets with no family
        0 packets with no route table
        0 packets with no auxiliary table
        0 packets with no corefacing entry
        0 packets with no CE-facing entry
         6 mac route learning requests # This indicates that VPLS is working.
        6 mac routes learnt
        0 mac routes aged
        0 mac routes moved
```

To display VPLS source and destination MAC address accounting information, use the destination, extensive, matching, or table option with the show route forwarding-table family vpls command. When you analyze the display output, keep in mind the following:

- VPLS MAC address accounting is handled on a per-MAC address basis for each VPLS instance. All information is retrieved from MAC address entries in the MAC address table. VPLS MAC address accounting is performed only on local CE routers.
- The VPLS counters for source and destination MAC addresses increment continuously until the oldest MAC address entries are removed from the memory buffer, either when the entries time out or if the VPLS instance is restarted.

```
user@PE1> show route forwarding-table family vpls extensive
```

```
Routing table: green.vpls [Index 2]
VPLS:

Destination:  default
  Route type: dynamic             Route reference: 0
  Flags: sent to PFE
  Next-hop type: flood                 Index: 353      Reference: 1

Destination:  default
  Route type: permanent           Route reference: 0
  Flags: none
  Next-hop type: discard               Index: 298      Reference: 1

Destination:  fe-0/1/0.0
  Route type: dynamic             Route reference: 0
  Flags: sent to PFE
  Next-hop type: flood                 Index: 355      Reference: 1

Destination:  bb:bb:bb:bb:bb:bb/48 # This MAC address belongs to remote
CE2.
  Route type: dynamic             Route reference: 0
  Flags: sent to PFE, prefix load balance
  Next-hop type: indirect              Index: 351      Reference: 4
  Next-hop type: Push 800000, Push 100002(top)
  Next-hop interface: so-1/1/0.0

Destination:  aa:aa:aa:aa:aa:aa/48 # This MAC address belongs to local
CE1.
  Route type: dynamic             Route reference: 0
  Flags: sent to PFE, prefix load balance
  Next-hop type: unicast               Index: 354      Reference: 2
  Next-hop interface: fe-0/1/0.0

user@PE1> show route forwarding-table family vpls

Routing table: green.vpls
VPLS:
Destination        Type RtRef Next hop          Type Index NhRef Netif
default            dynm     0                   flood  353     1
default            perm     0                   dscd   298     1
fe-0/1/0.0         dynm     0                   flood  355     1
bb:bb:bb:bb:bb:bb/48 # This MAC address belongs to remote CE2.
                   dynm     0                   indr   351     4
                                                Push 800000, Push
100002(top)
so-1/1/0.0
aa:aa:aa:aa:aa:aa/48 # This MAC address belongs to local CE1.
                  dynm     0                  ucst   354     2 fe-0/1/0.0


user@PE1> show route forwarding-table family mpls

Routing table: mpls
MPLS:
Destination        Type RtRef Next hop          Type Index NhRef Netif
default            perm     0                   dscd    19     1
0                  user     0                   recv    18     3
1                  user     0                   recv    18     3
2                  user     0                   recv    18     3
100000            user     0 10.11.1.6        swap  100001    so-1/1/0.0
```

```
800002             user     0                    Pop
vt-0/3/0.32770
vt-0/3/0.32770 (VPLS)
                   user     0                    indr   351     4
                                                 Push 800000, Push
100002(top) so-1/1/0.0


user@PE1> show route instance green detail

green:
  Router ID: 0.0.0.0
  Type: vpls              State: Active
  Interfaces:
    fe-0/1/0.0  # This is the local Fast Ethernet interface.
   vt-0/3/0.32770  # This is the dynamically generated VPLS virtual port.

  Route-distinguisher: 10.245.14.218:1
  Vrf-import: [ __vrf-import-green-internal__ ]
  Vrf-export: [ __vrf-export-green-internal__ ]
  Vrf-import-target: [ target:11111:1 ]
  Vrf-export-target: [ target:11111:1 ]
  Tables:
    green.l2vpn.0          : 2 routes (2 active, 0 holddown, 0 hidden)


user@PE1> show vpls connections

L2VPN Connections:
Legend for connection status (St)
OR -- out of range              WE -- intf encaps != instance encaps
EI -- encapsulation invalid     Dn -- down
EM -- encapsulation mismatch    VC-Dn -- Virtual circuit down
CM -- control-word mismatch     -> -- only outbound conn is up
CN -- circuit not present       <- -- only inbound  conn is up
OL -- no outgoing label         Up -- operational
NC -- intf encaps not CCC/TCC   XX -- unknown
NP -- interface not present
Legend for interface status
Up -- operational
Dn -- down
Instance: green
Local site: greenPE1 (1)
    connection-site           Type  St     Time last up          # Up
trans
    2                         rmt   Up     Jan 24 06:26:49 2003
  1
      Local interface: vt-0/3/0.32770, Status: Up, Encapsulation: VPLS
      Remote PE: 10.245.14.219, Negotiated control-word: No
      Incoming label: 800002, Outgoing label: 800000


user@PE1> show system statistics vpls

vpls:
        0 total packets received
        0 with size smaller than minimum
        0 with incorrect version number
        0 packets for this host
        0 packets with no logical interface
```

```
        0 packets with no family
        0 packets with no route table
        0 packets with no auxiliary table
        0 packets with no corefacing entry
        0 packets with no CE-facing entry
        7 mac route learning requests
        7 mac routes learnt
        0 mac routes aged
        0 mac routes moved


user@PE1> show route instance green detail

green:
  Router ID: 0.0.0.0
  Type: vpls              State: Active
   Interfaces:
    fe-0/1/0.0
    vt-0/3/0.32770
  Route-distinguisher: 10.245.14.218:1
  Vrf-import: [ __vrf-import-green-internal__ ]
  Vrf-export: [ __vrf-export-green-internal__ ]
  Vrf-import-target: [ target:11111:1 ]
  Vrf-export-target: [ target:11111:1 ]
  Tables:
    green.l2vpn.0          : 2 routes (2 active, 0 holddown, 0 hidden)

user@PE1> show vpls statistics

Layer-2 VPN Statistics:
Instance: green
   Local interface: fe-0/1/0.0, Index: 351
   Remote provider edge router: 10.245.14.219
     Multicast packets:                   363
     Multicast bytes  :                 30956
     Flood packets    :                     0
     Flood bytes      :                     0
   Local interface: vt-0/3/0.32770, Index: 354
   Remote provider edge router: 10.245.14.219
     Multicast packets:                   135
     Multicast bytes  :                 12014
     Flood packets    :                   135
     Flood bytes      :                 12014
```

To clear all MAC address entries for a VPLS instance from the VPLS table, issue the clear vpls mac-address instance *instance-name* command. Add the logical-system *logical-system-name* option to clear entries in a VPLS instance within a logical system. Use the *mac-address* option to remove individual MAC addresses.

Example: VPLS Configuration (BGP and LDP Interworking)

Figure 77: Topology for VPLS Configuration Example

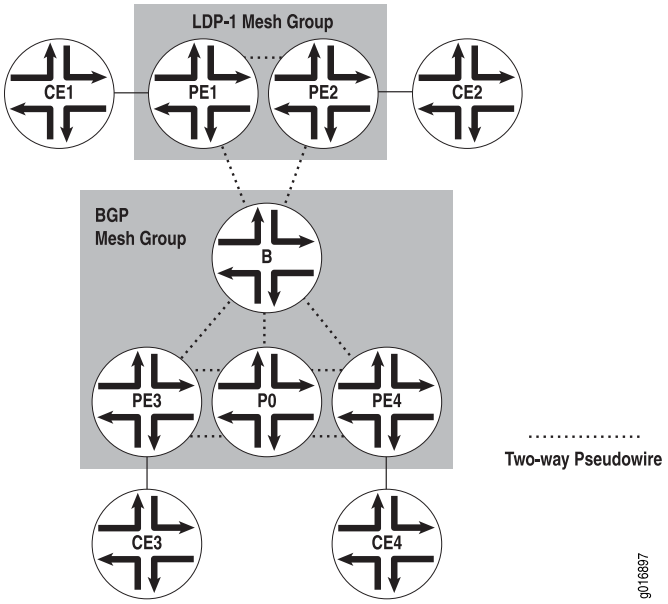


Figure 77 on page 717, shows two VPLS mesh groups: LDP-1 and the default BGP mesh group. The VPLS instance is named vi in the configuration. Table 44 on page 717 shows the addresses for the router interfaces in the example topology.

Table 44: Router Interface Addresses for VPLS Configuration Example

Router	Interface	Address
CE1	fe-0/0/3 (link to Router PE1)	10.12.31.1
	loopback	10.12.53.1
CE2	fe-0/0/1 (link to Router PE2)	10.12.31.2
	loopback	10.12.53.2
PE1	t1-1/1/1 (link to Router PE2)	10.12.100.17
	t1-0/1/0 (link to Router B)	10.12.100.2
	loopback	10.255.170.106
PE2	t1-0/1/1 (link to Router PE1)	10.12.100.18
	t1–0/1/3 (link to Router B)	10.12.100.6
	loopback	10.255.170.104

Table 44: Router Interface Addresses for VPLS Configuration Example *(continued)*

Router	Interface	Address
B	t1-0/1/2 (link to Router PE1)	10.12.100.1
	t1-0/1/3 (link to Router PE2)	10.12.100.5
	so-0/2/2 (link to Router PE3)	10.12.100.9
	fe-0/0/3 (link to Router PE4)	10.12.100.13
	loopback	10.255.170.98
PE3	s0-0/2/1 (link to Router B)	10.12.100.10
	so-0/2/2 (link to Router P0)	10.12.100.21
	loopback	10.255.170.96
P0	so-0/2/1 (link to Router PE3)	10.12.100.22
	t1-0/1/3 (link to Router PE4)	10.12.100.25
	loopback	10.255.170.100
PE4	fe-0/0/3 (link to Router B)	10.12.100.14
	t1-0/1/3 (link to Router P0)	10.12.100.26
	loopback	10.255.170.102
CE3	ge-1/2/1 (link to PE3)	10.12.31.3
	loopback	10.12.53.3
CE4	fe-0/0/2 (link to PE4)	10.12.31.4
	loopback	10.12.53.4

On Router CE3, the only item you need to configure is the Gigabit Ethernet interface that connects to PE3.

Router CE3

```
[edit]
interfaces {
    ge-1/2/1 {
        unit 0 {
            family inet {
                address 10.12.31.1/24;
            }
        }
    }
}
```

On Router PE3, prepare the router for VPLS by configuring BGP, MPLS, OSPF, and LDP. (These protocols are the basis for most Layer 2 VPN-related applications,

including VPLS.) Include the signaling statement at the [edit protocols bgp group *group-name* family l2vpn] hierarchy level, because VPLS uses the same infrastructure for internal BGP as Layer 2 VPNs.

NOTE: In JUNOS Release 7.3 and later, the signaling statement replaces the unicast statement at the [edit protocols bgp group *group-name* family l2vpn] hierarchy level. You must use the signaling statement if you wish to configure VPLS domains and Layer 2 VPNs simultaneously.

Next, configure VLAN tagging on the Gigabit Ethernet interface connected to Router CE3. Finally, add the Gigabit Ethernet interface into a VPLS routing instance and specify the site range, site ID number, and site name.

Router PE3

```
[edit]
interfaces {
    so-0/2/1 {
        unit 0 {
            family inet {
                address 10.12.100.10/30;
            }
            family mpls;
        }
    so-0/2/2 {
        unit 0 {
            family inet {
                address 10.12.100.21/30;
            }
            family mpls;
        }
    }
        ge-1/3/1 {
            encapsulation ethernet-vpls;
            unit 0 {
                family vpls;
            }
        }
    }
}
protocols {
    mpls {
        interface all;
    }
    bgp {
        log-updown;
        group int {
            type internal;
            local-address 10.255.170.96;
            family l2vpn {
                signaling;
            }
            neighbor 10.255.170.98;
            neighbor 10.255.170.102;
        }
```

```
    }
    ospf {
      area 0.0.0.0 {
        interface so-0/2/1.0;
        interface so-0/2/2.0;
        interface lo0.0 {
          passive;
        }
      }
    }
    ldp {
      interface so-0/2/1.0;
      interface so-0/2/2.0;
    }
  }
  routing-instances {
    v1 {
      instance-type vpls;
      interface ge-1/3/1.0;
      route-distinguisher 10.255.170.96:1;
      vrf-target target:1:2;
      protocols {
        vpls {
          site-range 10;
          site 1 {
            site-identifier 3;
          }
        }
      }
    }
```

On Router P0, configure BGP, MPLS, OSPF, and RSVP to interconnect PE3 and PE4.

Router P0

```
[edit]
interfaces {
  t1-0/1/3 {
    unit 0 {
      family inet {
        address 10.12.100.25/30;
      }
      family mpls;
    }
  so-0/2/1 {
    unit 0 {
      family inet {
        address 10.12.100.22/30;
      }
      family mpls;
    }
  }
}
protocols {
  mpls {
    interface all;
  }
  ospf {
```

```
        area 0.0.0.0 {
            interface so-0/2/1.0;
            interface t1-0/1/3.0;
            interface lo0.0 {
                passive;
            }
        }
    }
    ldp {
        interface t1-0/1/3.0;
        interface so-0/2/1.0;
    }
}
```

On Router PE4, configure BGP, MPLS, OSPF, and LDP to complement the configuration on PE3. Next, configure VLAN tagging on the Fast Ethernet interface connected to Router CE4. Include VLAN VPLS encapsulation at both the physical and logical interface levels. Finally, add the Fast Ethernet interface into a VPLS routing instance and specify the site range, site ID number, and site name.

Router PE4

```
[edit]
interfaces {
    fe-0/0/2 {
    encapsulation ethernet-vpls;
        unit 0 {
            family vpls;
        }
    fe-0/0/3 {
        unit 0 {
            family inet {
                address 10.12.100.14/30;
            }
            family mpls;
        }
    }
    t1-0/1/3 {
        unit 0 {
            family inet {
                address 10.12.100.26/30;
            }
            family mpls;
        }
    }
}
protocols {
    mpls {
        interface all;
    }
    bgp {
        log-updown;
        group int {
            type internal;
            local-address 10.255.170.102;
            family l2vpn {
                signaling;
            }
```

```
            neighbor 10.255.170.96;
            neighbor 10.255.170.98;
        }
        }
    }
    ospf {
        area 0.0.0.0 {
            interface fe-0/0/3.0;
            interface t1-0/1/3.0;
            interface lo0.0 {
                passive;
            }
        }
    }
    ldp {
        interface fe-0/0/3.0;
        interface t1-0/1/3.0;
        interface lo0.0;
    }
}
```

On Router CE4, configure the Fast Ethernet interface that connects to PE4.

Router CE4

```
[edit]
interfaces {
    fe-0/0/2 {
        unit 0 {
            family inet {
                address 10.12.31.4/24;
            }
        }
    }
}
```

On Router B, the area border router, configure the interfaces. Next, configure BGP, MPLS, OSPF, and LDP. Be sure to include the loopback interface in the LDP configuration by including the **interface lo0.0** statement at the [**edit protocols ldp**] hierarchy level. For BGP, include the **signaling** statement at the [**edit bgp group** *group-name* **family l2vpn**] hierarchy level. Last, configure the vpls instance with both bgp and ldp signaling. Configure the LDP-1 mesh group by including the **mesh-group ldp1** statement at the [**edit routing-instances v1 protocols vpls**] hierarchy level.

Router B

```
[edit]
interfaces {
    fe-0/0/3 {
        unit 0 {
            family inet {
                address 10.12.100.13/30;
            }
            family mpls;
        }
    }
    t1-0/1/2 {
        unit 0 {
            family inet {
```

```
            address 10.12.100.1/30;
          }
          family mpls;
        }
      }
      t1-0/1/3 {
        unit 0 {
          family inet {
            address 10.12.100.5/30;
          }
          family mpls;
        }
      }
      so-0/2/2 {
        unit 0 {
          family inet {
            address 10.12.100.9/30;
          }
          family mpls;
        }
      }
    }
    protocols {
      mpls {
        interface all;
      }
      bgp {
        log-updown;
        group int {
          type internal;
          local-address 10.255.170.98;
          family l2vpn {
            signaling;
          }
          neighbor 10.255.170.96;
          neighbor 10.255.170.102;
        }
      }
      ospf {
        area 0.0.0.0 {
          interface t1-0/1/2.0;
          interface t1-0/1/3.0;
          interface so-0/2/2.0;
          interface fe-0/0/3.0;
          interface lo0.0 {
            passive;
          }
        }
      }
      ldp {
        interface fe-0/0/3.0;
        interface t1-0/1/2.0;
        interface t1-0/1/3.0;
        interface so-0/2/2.0;
        interface lo0.0;
      }
```

```
}
routing-instances {
   v1 {
      instance-type vpls;
      route-distinguisher 10.255.170.98:1;
      vrf-target target:1:2;
      protocols {
         vpls {
            site-range 10;
            site 1 {
               site-identifier 1;
            }
            vpls-id 101;
            mesh-group ldp-1 {
               neighbor 10.255.170.106;
               neighbor 10.255.170.104;
            }
         }
      }
   }
}
```

Finally, configure the LDP PE routers. On Router PE1, prepare the router for VPLS by configuring LDP, MPLS, and OSPF. Next, configure VPLS encapsulation on the Fast Ethernet interface connected to CE1. Finally, add the Fast Ethernet interface to the routing instance, specifying the VPLS ID and the neighboring routers' loopback addresses.

Router PE1

```
[edit]
interfaces {
   fe-0/0/3 {
      encapsulation ethernet-vpls;
      unit 0 {
         family vpls;
      }
   }
   t1-0/1/0 {
      unit 0 {
         family inet {
            address 10.12.100.2/30;
         }
         family mpls;
      }
   }
   t1-1/1/1 {
      unit 0 {
         family inet {
            address 10.12.100.17/30;
         }
         family mpls;
      }
   }
}
protocols {
   mpls {
```

```
        interface all;
    }
    ospf {
        area 0.0.0.0 {
            interface t1-0/1/0.0;
            interface t1-1/1/1.0;
            interface lo0.0 {
                passive;
            }
        }
    }
    ldp {
        interface t1-0/1/0.0;
        interface t1-1/1/1.0;
        interface lo0.0;
    }
}
routing-instances {
    v1 {
        instance-type vpls;
        interface fe-0/0/3.0;
        protocols {
            vpls {
                vpls-id 101;
                neighbor 10.255.170.98;
                neighbor 10.255.170.104;
            }
        }
    }
}
```

Next, configure the Fast Ethernet interface on router CE1 that connects to router PE1.

Router CE1

```
[edit]
interfaces {
    fe-0/0/3 {
        unit 0 {
            family inet {
                address 10.12.31.1/24;
            }
        }
    }
}
```

On Router PE2, prepare the router for VPLS by configuring LDP, MPLS, and OSPF. Next, configure VPLS encapsulation on the Fast Ethernet interface connected to router CE1. Finally, add the Fast Ethernet interface to the routing instance, specifying the VPLS ID and the neighboring routers' loopback addresses.

Router PE2

```
[edit]
interfaces {
    t1-0/1/1 {
        unit 0 {
            family inet {
```

```
            address 10.12.100.18/30;
          }
          family mpls;
        }
      t1-0/1/3 {
        unit 0 {
          family inet {
            address 10.12.100.6/30;
          }
          family mpls;
        }
      }
      fe-1/0/2 {
        encapsulation ethernet-vpls;
        unit 0 {
          family vpls;
        }
      }
    }
    protocols {
      mpls {
        interface all;
      }
      ospf {
        area 0.0.0.0 {
          interface t1-0/1/3.0;
          interface t1-0/1/1.0;
          interface lo0.0 {
            passive;
          }
        }
      }
      ldp {
        interface t1-0/1/1.0;
        interface t1-0/1/3.0;
        interface lo0.0;
      }
    }
    routing-instances {
      v1 {
        instance-type vpls;
        interface fe-1/0/2.0;
        protocols {
          vpls {
            vpls-id 101;
            neighbor 10.255.170.98;
            neighbor 10.255.170.106;
          }
        }
      }
    }
```

Finally, on Router CE2 configure the Fast Ethernet interface connected to PE2:

Router CE2 [edit]

```
interfaces {
   fe-0/0/1 {
      unit 0 {
         family inet {
            address 10.12.31.2/24;
         }
      }
   }
}
```

Verifying Your Work

To verify proper operation of VPLS, use the following commands:

- show bgp summary
- show ldp neighbor
- show vpls connections
- show route forwarding-table family vpls (destination | extensive | matching | table)
- show interfaces vt* terse
- show vpls flood extensive
- show vpls statistics

The following section shows the output of some of these commands on Router B as a result of the configuration example.

Use the show bgp summary command to verify BGP signaling for VPLS is up.

```
user@PB> show bgp summary
Groups: 1 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l2vpn.0            2          2          0          0          0          0
Peer               AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Damped...
10.255.170.96   65000        124        125       0       0       54:26 Establ
  bgp.l2vpn.0: 1/1/0
  v1.l2vpn.0: 1/1/0
10.255.170.102  65000        122        124       0       0       54:18 Establ
  bgp.l2vpn.0: 1/1/0
  v1.l2vpn.0: 1/1/0
```

Use the show ldp neighbors command to verify ldp signaling for VPLS is up.

```
user@B> show ldp neighbors
Address            Interface          Label space ID         Hold time
10.255.170.104     lo0.0              10.255.170.104:0         41
10.255.170.106     lo0.0              10.255.170.106:0         38
10.12.100.14       fe-0/0/3.0         10.255.170.102:0         12
10.12.100.10       so-0/2/2.0         10.255.170.96:0          14
10.12.100.2        t1-0/1/2.0         10.255.170.106:0         14
10.12.100.6        t1-0/1/3.0         10.255.170.104:0         13
```

To verify that the VPLS connections are up, use the show vpls connections command.

```
user@B>show vpls connections
Layer-2 VPN connections:

Legend for connection status (St)
EI -- encapsulation invalid      NC -- interface encapsulation not CCC/TCC/VPLS
EM -- encapsulation mismatch     WE -- interface and instance encaps not same
VC-Dn -- Virtual circuit down    NP -- interface hardware not present
CM -- control-word mismatch      -> -- only outbound connection is up
CN -- circuit not provisioned    <- -- only inbound connection is up
OR -- out of range               Up -- operational
OL -- no outgoing label          Dn -- down
LD -- local site signaled down   CF -- call admission control failure
RD -- remote site signaled down  SC -- local and remote site ID collision LN --
local site not designated  LM -- local site ID not minimum designated RN -- remote
 site not designated RM -- remote site ID not minimum designated XX -- unknown
connection status  IL -- no incoming label
MM -- MTU mismatch               MI -- Mesh-Group ID not availble

Legend for interface status
Up -- operational
Dn -- down

Instance: v1
BGP-VPLS State
  Local site: 1 (1)
    connection-site           Type  St     Time last up          # Up trans
    3                         rmt   Up     Jan 22 16:38:47 2008           1
      Local interface: vt-0/3/0.1048834, Status: Up, Encapsulation: VPLS
        Description: Intf - vpls v1 local site 1 remote site 3
      Remote PE: 10.255.170.96, Negotiated control-word: No
      Incoming label: 800258, Outgoing label: 800000
    4                         rmt   Up     Jan 22 16:38:54 2008           1
      Local interface: vt-0/3/0.1048835, Status: Up, Encapsulation: VPLS
        Description: Intf - vpls v1 local site 1 remote site 4
      Remote PE: 10.255.170.102, Negotiated control-word: No
      Incoming label: 800259, Outgoing label: 800000 LDP-VPLS State
VPLS-id: 101
  Mesh-group connections: m1
    Neighbor                  Type  St     Time last up          # Up trans
    10.255.170.104(vpls-id 101) rmt Up     Jan 22 16:38:40 2008           1
      Local interface: vt-0/3/0.1048833, Status: Up, Encapsulation: ETHERNET
        Description: Intf - vpls v1 neighbor 10.255.170.104 vpls-id 101
      Remote PE: 10.255.170.104, Negotiated control-word: No
      Incoming label: 800001, Outgoing label: 800000
    10.255.170.106(vpls-id 101) rmt Up     Jan 22 16:38:39 2008           1
      Local interface: vt-0/3/0.1048832, Status: Up, Encapsulation: ETHERNET
        Description: Intf - vpls v1 neighbor 10.255.170.106 vpls-id 101
      Remote PE: 10.255.170.106, Negotiated control-word: No
      Incoming label: 800000, Outgoing label: 800000
```

To display VPLS routes (MAC addresses) in the vpls forwarding table, use the show route forwarding-table family vpls command.

```
user@B> show route forwarding-table family vpls
Routing table: v1.vpls
VPLS:
Destination        Type RtRef Next hop           Type Index NhRef Netif
default            perm     0                    rjct   540     1
vt-0/3/0.1048832   user     0                    comp   587     3
```

```
vt-0/3/0.1048833   user     0                    comp   587     3
vt-0/3/0.1048834   user     0                    comp   589     3
vt-0/3/0.1048835   user     0                    comp   589     3
00:17:cb:c2:10:01/48
                   dynm     0                    indr 262143     4
                                                Push 800000   580     2
t1-0/1/3.0
00:17:cb:c2:10:02/48
                   dynm     0                    indr 262145     4
                              10.12.100.14      Push 800000   594     2
fe-0/0/3.0
00:17:cb:c2:10:03/48
                   dynm     0                    indr 262142     4
                                                Push 800000   576     2
t1-0/1/2.0
00:17:cb:c2:10:bd/48
                   dynm     0                    indr 262144     4
                                                Push 800000   585     2
so-0/2/2.0
```

To display VPLS source and destination MAC address accounting information, use the destination, extensive, matching, or table option with the show route forwarding-table family vpls command. When you analyze the display output, keep in mind the following:

- VPLS MAC address accounting is handled on a per-MAC address basis for each VPLS instance. All information is retrieved from MAC address entries in the MAC address table. VPLS MAC address accounting is performed only on local CE routers.
- The VPLS counters for source and destination MAC addresses increment continuously until the oldest MAC address entries are removed from the memory buffer, either when the entries time out or if the VPLS instance is restarted.

To display status information about Virtual Loopback Tunnel interfaces in the VPLS instance, use the show interfaces vt* terse command.

```
user@B> show interfaces vt* terse
Interface               Admin Link Proto    Local                 Remote
vt-0/3/0                up    up
vt-0/3/0.1048832        up    up   vpls
vt-0/3/0.1048833        up    up   vpls
vt-0/3/0.1048834        up    up   vpls
vt-0/3/0.1048835        up    up   vpls
```

To display VPLS route information related to the flood process, use the show vpls flood extensive command.

```
user@B> show vpls flood extensive
Name: v1
CEs: 0
VEs: 4
  Flood route prefix: 0x4a/32
  Flood route type: IFF_FLOOD
  Flood route owner: vt-0/3/0.1048834
  Flood group name: __ves__
  Flood group index: 0
```

```
Nexthop type: comp
Nexthop index: 589
  Flooding to:
  Name             Type          NhType          Index
  m1               Group          comp            588
      Composition: flood-to-all
      Flooding to:
      Name             Type          NhType          Index
      vt-0/3/0.1048832 VE             indr            262142
      vt-0/3/0.1048833 VE             indr            262143

Flood route prefix: 0x4b/32
Flood route type: IFF_FLOOD
Flood route owner: vt-0/3/0.1048835
Flood group name: __ves__
Flood group index: 0
Nexthop type: comp
Nexthop index: 589
  Flooding to:
  Name             Type          NhType          Index
  m1               Group          comp            588
      Composition: flood-to-all
      Flooding to:
      Name             Type          NhType          Index
      vt-0/3/0.1048832 VE             indr            262142
      vt-0/3/0.1048833 VE             indr            262143

Flood route prefix: 0x48/32
Flood route type: IFF_FLOOD
Flood route owner: vt-0/3/0.1048832
Flood group name: m1
Flood group index: 2
Nexthop type: comp
Nexthop index: 587
  Flooding to:
  Name             Type          NhType          Index
  __ves__          Group          comp            586
      Composition: flood-to-all
      Flooding to:
      Name             Type          NhType          Index
      vt-0/3/0.1048834 VE             indr            262144
      vt-0/3/0.1048835 VE             indr            262145

Flood route prefix: 0x49/32
Flood route type: IFF_FLOOD
Flood route owner: vt-0/3/0.1048833
Flood group name: m1
Flood group index: 2
Nexthop type: comp
Nexthop index: 587
  Flooding to:
  Name             Type          NhType          Index
  __ves__          Group          comp            586
      Composition: flood-to-all
      Flooding to:
      Name             Type          NhType          Index
      vt-0/3/0.1048834 VE             indr            262144
      vt-0/3/0.1048835 VE             indr            262145
```

To view packet flow statistics for the VPLS instance, use the show vpls statistics command:

```
user@B> show vpls statistics
Instance: v1
   Local interface: vt-0/3/0.1048832, Index: 72
   Remote PE: 10.255.170.106
     Multicast packets:                    6
     Multicast bytes  :                  360
     Flooded packets  :                   16
     Flooded bytes    :                 1188
     Current MAC count:                    1
   Local interface: vt-0/3/0.1048833, Index: 73
   Remote PE: 10.255.170.104
     Multicast packets:                    4
     Multicast bytes  :                  240
     Flooded packets  :                    6
     Flooded bytes    :                  398
     Current MAC count:                    1
   Local interface: vt-0/3/0.1048834, Index: 74
   Remote PE: 10.255.170.96
     Multicast packets:                    2
     Multicast bytes  :                  120
     Flooded packets  :                    4
     Flooded bytes    :                  278
     Current MAC count:                    1
   Local interface: vt-0/3/0.1048835, Index: 75
   Remote PE: 10.255.170.102
     Multicast packets:                    1
     Multicast bytes  :                   60
     Flooded packets  :                    2
     Flooded bytes    :                  158
     Current MAC count:                    1
```

Example: Configuring Nonstop Active Routing

The following example enables graceful Routing Engine switchover, nonstop active routing, and nonstop active routing trace options for VPLS.

```
[edit]
system commit {
   synchronize;
}
chassis {
   redundancy {
      graceful-switchover; # This enables graceful Routing Engine switchover on the
         # routing platform.
   }
}
routing-options {
   nonstop-routing; # This enables nonstop active routing on the routing platform.
   traceoptions {
      flag nsr-synchronization;
   }
}
```

Example: Configuring Inter-AS VPLS with MAC Processing at the ASBR

This example describes how to configure inter-AS Virtual Private LAN Service (VPLS) with MAC processing between BGP-signaled VPLS and LDP-signaled VPLS. This feature is described in RFC 4761 as multi-AS VPLS option E or method E .

This example is organized in the following sections:

Requirements

To support inter-AS VPLS between BGP-signaled VPLS and LDP-signaled VPLS, your network must meet the following hardware and software requirements:

- MX-series or M320 routers for the ASBRs.
- JUNOS software release 9.3 or higher.
- Gigabit Ethernet or 10–Gigabit Ethernet interfaces.

Overview and Topology

VPLS is a key enabler for delivering multipoint Ethernet service. Major service providers have implemented IP and MPLS backbones and offer VPLS services to large enterprises. Growing demand requires the VPLS network to scale to support many VPLS customers with multiple sites spread across geographically dispersed regions. BGP-signaled VPLS signaling offers scaling advantages over LDP-signaled VPLS. In some environments there is a need for BGP-signaled VPLS to interoperate with existing LDP-signaled VPLS.

This example shows one way to configure BGP-signaled VPLS interworking with an existing LDP-signaled VPLS network.

The advantages of the configuration are:

- You can interconnect customer sites that are spread across different autonomous systems (ASs).
- LDP-signaled VPLS and BGP-signaled VPLS interworking is supported.
- Because the ASBR supports MAC operations, customer sites can be connected directly to the ASBR.
- The inter-AS link is not restricted to Ethernet interfaces.
- Additional configuration for multihoming is relatively straightforward.

Traffic from the interworking virtual private LAN services is switched at the ASBR. The ASBR does all the data plane operations: flooding, MAC learning, aging, and MAC

forwarding for each AS to switch traffic among any customer facing interfaces and between the fully meshed pseudowires in the AS. A single pseudowire is created between the ASBRs across the inter-AS link and the ASBRs forward traffic from the pseudowires in each AS to the peer ASBR.

Each ASBR performs VPLS operations within its own AS and performs VPLS operations with the ASBR in the other AS. The ASBR treats the other AS as a BGP-signaled VPLS site. To establish VPLS pseudowires, VPLS NLRI messages are exchanged across the EBGP sessions on the inter-AS links between the ASBRs.

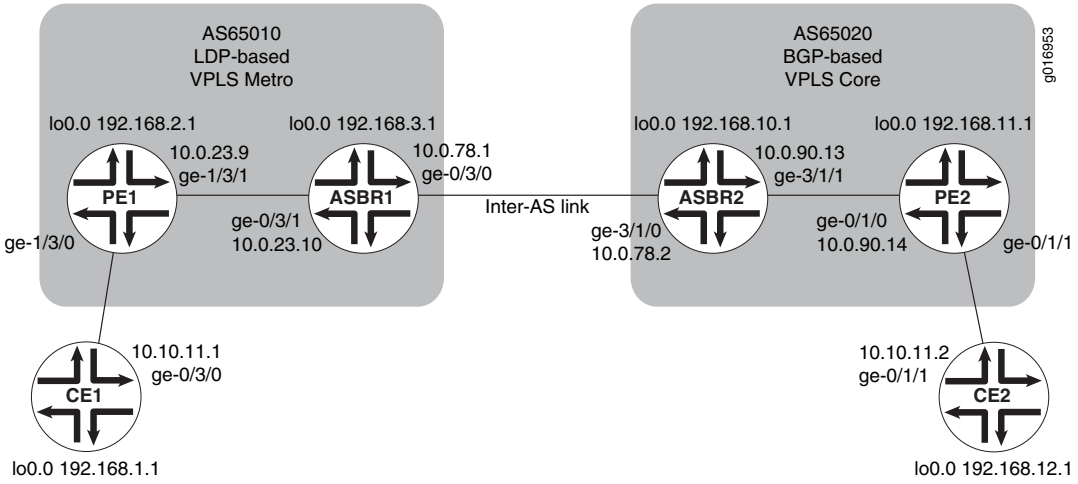
The example metro network is configured for LDP-signaled VPLS. The core network is configured for BGP-signaled VPLS.

The first part of the example shows the basic configuration steps to configure the logical interfaces, OSPF, internal BGP, LDP, and MPLS. This part of the configuration is the same as other VPLS configurations for LDP-signaled VPLS and BGP-signaled VPLS.

The unique part of the example is configured in the VPLS routing instances, external BGP, and the policy that populates the BGP route table with routes learned from direct routes and OSPF routes. Additional details about the configuration statements are included in the step-by-step procedure.

Figure 78 on page 733 shows the topology used in this example.

Figure 78: Inter-AS VPLS with MAC Operations Example Topology



Configuration

To configure inter-AS VPLS between BGP-signaled VPLS and LDP-signaled VPLS, perform these tasks:

NOTE: In any configuration session it is a good practice to periodically use the commit check command to verify that the configuration can be committed.

Configuring Interfaces

Step-by-Step Procedure To configure interfaces:

1. On each router, configure an IP address on the loopback logical interface 0 (lo0.0):

   user@CE1# **set interfaces lo0 unit 0 family inet address 192.168.1.1/32 primary**

   user@PE1# **set interfaces lo0 unit 0 family inet address 192.168.2.1/32 primary**

   user@ASBR1# **set interfaces lo0 unit 0 family inet address 192.168.3.1/32 primary**

   user@ASBR2# **set interfaces lo0 unit 0 family inet address 192.168.10.1/32 primary**

   user@PE2# **set interfaces lo0 unit 0 family inet address 192.168.11.1/32 primary**

   user@CE2# **set interfaces lo0 unit 0 family inet address 192.168.12.1/32 primary**

2. On each router, commit the configuration:

   ```
   user@host> commit
   ```

3. On each router, display the interface information for lo0 and verify that the correct IP address is configured:

```
user@host> show interfaces lo0

Physical interface: lo0, Enabled, Physical link is Up
  Interface index: 6, SNMP ifIndex: 6
  Type: Loopback, MTU: Unlimited
  Device flags   : Present Running Loopback
  Interface flags: SNMP-Traps
  Link flags     : None
  Last flapped   : Never
    Input packets : 0
    Output packets: 0

  Logical interface lo0.0 (Index 75) (SNMP ifIndex 16)
    Flags: SNMP-Traps Encapsulation: Unspecified
    Input packets : 0
    Output packets: 0
    Protocol inet, MTU: Unlimited
      Flags: None
      Addresses
        Local: 127.0.0.1
      Addresses, Flags: Primary Is-Default Is-Primary
        Local: 192.168.3.1
Logical interface lo0.16384 (Index 64) (SNMP ifIndex 21)
    Flags: SNMP-Traps Encapsulation: Unspecified
    Input packets : 0
    Output packets: 0
    Protocol inet, MTU: Unlimited
      Flags: None
      Addresses
        Local: 127.0.0.1

  Logical interface lo0.16385 (Index 65) (SNMP ifIndex 22)
    Flags: SNMP-Traps Encapsulation: Unspecified
    Input packets : 0
    Output packets: 0
    Protocol inet, MTU: Unlimited
      Flags: None
```

In the example above notice that the primary lo0 local address for the inet protocol family on ASBR1 is 192:168:3:1.

4. On each router, configure an IP address and protocol family on the Gigabit Ethernet interfaces. Specify the inet protocol family.

```
user@CE1# set interfaces ge-0/3/0 unit 0 family inet address 10.10.11.1/24

user@PE1# set interfaces ge-1/3/1 unit 0 family inet address 10.0.23.9/30

user@ASBR1# set interfaces ge-0/3/1 unit 0 family inet address
  10.0.23.10/30
user@ASBR1# set interfaces ge-0/3/0 unit 0 family inet address 10.0.78.1/30

user@ASBR2# set interfaces ge-3/1/0 unit 0 family inet address 10.0.78.2/30
```

```
user@ASBR2# set interfaces ge-3/1/1 unit 0 family inet address
  10.0.90.13/30

user@PE2# set interfaces ge-0/1/0 unit 0 family inet address 10.0.90.14/30

user@CE2# set interfaces ge-0/1/1 unit 0 family inet address 10.10.11.2/24
```

5. On each router, commit the configuration:

```
user@host> commit
```

6. Display information for Gigabit Ethernet interfaces and verify that the IP address and protocol family are configured correctly.

```
user@ASBR2> show interfaces ge-* terse

Interface               Admin Link Proto    Local             Remote
ge-3/1/0                up    up
ge-3/1/0.0              up    up   inet     10.0.78.2/30
                                  multiservice
ge-3/1/1                up    up
ge-3/1/1.0              up    up   inet     10.0.90.13/30
                                  multiservice
ge-3/1/2                up    down
ge-3/1/3                up    down
```

Configuring OSPF

Step-by-Step Procedure To configure OSPF:

1. On the PE and ASBR routers, configure the provider instance of OSPF. Configure OSPF traffic engineering support. Specify area 0.0.0.1 in the LDP-signaled VPLS network and area 0.0.0.0 in the BGP-signaled network. Specify the Gigabit Ethernet logical interfaces between the PE and ASBR routers. Specify lo0.0 as a passive interface.

```
user@PE1# set protocols ospf traffic-engineering
user@PE1# set protocols ospf area 0.0.0.1 interface ge-1/3/1.0
user@PE1# set protocols ospf area 0.0.0.1 interface lo0.0 passive

user@ASBR1# set protocols ospf traffic-engineering
user@ASBR1# set protocols ospf area 0.0.0.1 interface ge-0/3/1.0
user@ASBR1# set protocols ospf area 0.0.0.1 interface lo0.0 passive

user@ASBR2# set protocols ospf traffic-engineering
user@ASBR2# set protocols ospf area 0.0.0.0 interface ge-3/1/1.0
user@ASBR2# set protocols ospf area 0.0.0.0 interface lo0.0 passive

user@PE2# set protocols ospf traffic-engineering
user@PE2# set protocols ospf area 0.0.0.0 interface ge-0/1/0.0
```

```
user@PE2# set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

2. On each router, commit the configuration:

```
user@PE1> commit
```

3. Display OSPF neighbor information and verify that the PE routers form adjacencies with the ASBR router in the same area:

```
user@host> show ospf neighbor

Address          Interface         State     ID            Pri  Dead
10.0.23.10       ge-1/3/1.0        Full      192.168.3.1   128  31
```

Notice that the neighbor state is full.

Configuring the Internal BGP Peer Group

Step-by-Step Procedure

The purpose of configuring an internal BGP peer group is to create a full mesh of BGP LSPs among the PE routers in the BGP-signaled AS including the ASBRs .To configure the internal BGP peer group:

1. The purpose of this step is to create a full mesh of IBGP peers between the PE routers, including the ASBRs, within the BGP-signaled AS.

   On ASBR2, configure internal BGP. Specify the BGP type as internal. Specify the local address as the local lo0 IP address.

   Specify the inet protocol family. Specify the labeled-unicast statement and the resolve-vpn option. The labeled-unicast statement causes the router to advertise labeled routes out of the IPv4 inet.0 route table and places labeled routes into the inet.0 route table. The resolve-vpn option puts labeled routes in the MPLS inet.3 route table The inet.3 route table is used to resolve routes for the PE router located in the other AS.

   Specify the l2vpn family to indicate to the router that this is a VPLS. Specify the signaling option to configure BGP as the signaling protocol. This enables BGP to carry Layer 2 VPLS NLRI messages for this peer group.

   Specify the lo0 interface IP address of the PE as the neighbor. Configure an autonomous system identifier.

```
user@ASBR2# set protocols bgp group core-ibgp type internal
user@ASBR2# set protocols bgp group core-ibgp local-address 192.168.10.1
user@ASBR2# set protocols bgp group core-ibgp family inet labeled-unicast
  resolve-vpn
user@ASBR2# set protocols bgp group core-ibgp family l2vpn signaling
user@ASBR2# set protocols bgp group core-ibgp neighbor 192.168.11.1
user@ASBR2# set routing-options autonomous-system 0.65020
```

2. On PE2, configure internal BGP. Specify the BGP type as internal. Specify the local address as the local lo0 IP address.

   Specify the l2vpn family to indicate this is a VPLS. Specify the signaling option to configure BGP as the signaling protocol. This enables BGP to carry Layer 2 VPLS NLRI messages.

   Specify the lo0 interface IP address of ASBR2 as the neighbor. Configure an autonomous system identifier.

   ```
   user@PE2# set protocols bgp group core-ibgp type internal
   user@PE2# set protocols bgp group core-ibgp local-address 192.168.11.1
   user@PE2# set protocols bgp group core-ibgp family l2vpn signaling
   user@PE2# set protocols bgp group core-ibgp neighbor 192.168.10.1
   user@PE2# set routing-options autonomous-system 0.65020
   ```

3. On each router, commit the configuration:

   ```
   user@host> commit
   ```

4. On PE2 and ASBR2, display BGP neighbor information:

   ```
   user@ASBR2> show bgp neighbor

   Peer: 192.168.11.1+49443 AS 65020 Local: 192.168.10.1+179 AS 65020
     Type: Internal    State: Established    Flags: ImportEval Sync
     Last State: OpenConfirm   Last Event: RecvKeepAlive
     Last Error: None
     Options: Preference LocalAddress AddressFamily Rib-group Refresh
     Address families configured: l2vpn-signaling inet-labeled-unicast
     Local Address: 192.168.10.1 Holdtime: 90 Preference: 170
     Number of flaps: 0
     Peer ID: 192.168.11.1     Local ID: 192.168.10.1     Active Holdtime:
   90
     Keepalive Interval: 30         Peer index: 0

   ...
   ```

   Verify that the peer connection state is established.

Configuring LDP

Step-by-Step Procedure To configure LDP:

1. On the PE and ASBR routers, configure LDP with the Gigabit Ethernet interfaces between the PE and ASBR routers, and between the two ASBRs. To support LDP-signaled VPLS, additionally configure LDP with the lo0.0 interface on PE1 and ASBR1:

   ```
   user@PE1# set protocols ldp interface ge-1/3/1.0
   user@PE1# set protocols ldp interface lo0.0
   ```

```
user@ASBR1# set protocols ldp interface ge-0/3/1.0
user@ASBR1# set protocols ldp interface ge-0/3/0.0
user@ASBR1# set protocols ldp interface lo0.0

user@ASBR2# set protocols ldp interface ge-3/1/0.0
user@ASBR2# set protocols ldp interface ge-3/1/1.0

user@PE2# set protocols ldp interface ge-0/1/0.0
```

2. On each router, commit the configuration:

```
user@host> commit
```

3. Display LDP configuration information and verify that the correct interfaces are configured. LDP operation can be verified after MPLS is configured.

```
user@ASBR1> show configuration protocols ldp

interface ge-0/3/0.0;
interface ge-0/3/1.0;
interface lo0.0;
```

The example shown above is from ASBR1.

Configuring MPLS

Step-by-Step Procedure To configure MPLS:

1. On the PE and ASBR routers, configure MPLS. Enable MPLS on the logical interfaces. Add the Gigabit Ethernet interfaces to the MPLS protocol. This adds entries to the MPLS forwarding table.

```
user@pe1# set protocols mpls interface ge-1/3/1.0
user@pe1# set interfaces ge-1/3/1 unit 0 family mpls

user@ASBR1# set protocols mpls interface ge-0/3/1.0
user@ASBR1# set protocols mpls interface ge-0/3/0.0
user@ASBR1# set interfaces ge-0/3/1 unit 0 family mpls
user@ASBR1# set interfaces ge-0/3/0 unit 0 family mpls

user@ASBR2# set protocols mpls interface ge-3/1/0.0
user@ASBR2# set protocols mpls interface ge-3/1/1.0
user@ASBR2# set interfaces ge-3/1/0 unit 0 family mpls
user@ASBR2# set interfaces ge-3/1/1 unit 0 family mpls

user@pe2# set protocols mpls interface ge-0/1/0.0
user@pe2# set interfaces ge-0/1/0 unit 0 family mpls
```

2. On each router, commit the configuration:

```
user@host> commit
```

3. On the PE and ASBR routers, display LDP neighbor information and verify that the directly connected LDP neighbors are listed:

```
user@ASBR1> show ldp neighbor

Address            Interface          Label space ID         Hold time
192.168.2.1        lo0.0              192.168.2.1:0            44
10.0.78.2          ge-0/3/0.0         192.168.10.1:0           13
10.0.23.9          ge-0/3/1.0         192.168.2.1:0            11
```

The example shown above is from ASBR1.

Configuring the External BGP Peer Group Between the Loopback Interfaces

Step-by-Step Procedure To configure the external BGP (EBGP) peer group between the loopback interfaces:

1. On ASBR1 and PE1, configure an autonomous system identifier:

```
user@PE1# set routing-options autonomous-system 0.65010

user@ASBR1# set routing-options autonomous-system 0.65010
```

2. On ASBR1, configure an external BGP peer group for the loopback interfaces. Specify the external BGP group type. Include the multihop statement. Specify the local address as the local lo0 IP address. Configure the l2vpn family for BGP signaling. Configure the peer AS as the core AS number. Specify the lo0 IP address of ASBR2 as the neighbor.

```
user@ASBR1# set protocols bgp group vpls-core type external
user@ASBR1# set protocols bgp group vpls-core multihop
user@ASBR1# set protocols bgp group vpls-core local-address 192.168.3.1
user@ASBR1# set protocols bgp group vpls-core family l2vpn signaling
user@ASBR1# set protocols bgp group vpls-core peer-as 65020
user@ASBR1# set protocols bgp group vpls-core neighbor 192.168.10.1
```

3. On ASBR2, configure an external BGP peer group for the loopback interfaces. Specify the external BGP group type. Include the multihop statement. The multihop statement is needed because the EBGP neighbors are in different ASs. Specify the local address as the local lo0 IP address. Configure the l2vpn family for BGP signaling. Configure the peer AS as the metro AS number. Specify the lo0 IP address of ASBR1 as the neighbor.

```
user@ASBR2# set protocols bgp group vpls-metro type external
user@ASBR2# set protocols bgp group vpls-metro multihop
user@ASBR2# set protocols bgp group vpls-metro local-address 192.168.10.1
user@ASBR2# set protocols bgp group vpls-metro family l2vpn signaling
user@ASBR2# set protocols bgp group vpls-metro peer-as 65010
```

```
user@ASBR2# set protocols bgp group vpls-metro neighbor 192.168.3.1
```

4. On each router, commit the configuration:

```
user@host> commit
```

Configuring the External BGP Peer Group Between the Inter-AS Link Interfaces

Step-by-Step Procedure

The purpose of configuring external BGP peer groups between the inter-AS link interfaces is to create a full mesh of BGP LSPs among the ASBRs.To configure the external BGP peer group between the Inter-AS Link interfaces:

1. On ASBR1, configure a policy to export OSPF and direct routes, including the lo0 address of the PE routers, into BGP for the establishment of label-switched paths (LSPs):

```
user@ASBR1# set policy-options policy-statement loopback term term1 from
  protocol ospf
user@ASBR1# set policy-options policy-statement loopback term term1 from
  protocol direct
user@ASBR1# set policy-options policy-statement loopback term term1 from
  route-filter 192.168.0.0/16 longer
user@ASBR1# set policy-options policy-statement loopback term term1 then
  accept
```

2. On ASBR1, configure an external BGP peer group for the inter-AS link. Specify the external BGP group type. Specify the local inter-AS link IP address as the local address. Configure the inet family and include the labeled-unicast and resolve-vpn statements. The labeled-unicast statement advertises labeled routes out of the IPv4 inet.0 route table and places labeled routes into the inet.0 route table. The resolve-vpn option stores labeled routes in the MPLS inet.3 route table.

   Include the export statement and specify the policy you created. Configure the peer AS as the core AS number. Specify the inter-AS link IP address of ASBR2 as the neighbor.

```
user@ASBR1# set protocols bgp group metro-core type external
user@ASBR1# set protocols bgp group metro-core local-address 10.0.78.1
user@ASBR1# set protocols bgp group metro-core family inet labeled-unicast
  resolve-vpn
user@ASBR1# set protocols bgp group metro-core export loopback
user@ASBR1# set protocols bgp group metro-core peer-as 65020
user@ASBR1# set protocols bgp group metro-core neighbor 10.0.78.2
```

3. On ASBR2, configure a policy to export OSPF and direct routes, including the lo0 address, into BGP for the establishment of label-switched paths (LSPs):

```
user@ASBR2# set policy-options policy-statement loopback term term1 from
  protocol ospf
```

```
user@ASBR2# set policy-options policy-statement loopback term term1 from
  protocol direct
user@ASBR2# set policy-options policy-statement loopback term term1 from
  route-filter 192.168.0.0/16 longer
user@ASBR2# set policy-options policy-statement loopback term term1 then
  accept
```

4. On ASBR2, configure an external BGP peer group for the inter-AS link. Specify the external BGP group type. Specify the local inter-AS link IP address as the local address. Configure the inet family and include the labeled-unicast and resolve-vpn statements. Include the export statement and specify the policy you created. Configure the peer AS as the core AS number. Specify the inter-AS link IP address of ASBR1 as the neighbor.

```
user@ASBR2# set protocols bgp group core-metro type external
user@ASBR2# set protocols bgp group core-metro local-address 10.0.78.2
user@ASBR2# set protocols bgp group core-metro family inet labeled-unicast
  resolve-vpn
user@ASBR2# set protocols bgp group core-metro export loopback
user@ASBR2# set protocols bgp group core-metro peer-as 65010
user@ASBR2# set protocols bgp group core-metro neighbor 10.0.78.1
```

5. On each router, commit the configuration:

```
user@host> commit
```

6. On ASBR1, display the BGP neighbors. Verify that the first peer is the IP address of the Gigabit Ethernet interface of ASBR2. Verify that the second peer is the IP address of the lo0 interface of ASBR2. Also verify that the state of each peer is Established. Notice that on ASBR1 the NLRI advertised by ASBR2 the inter-AS link peer is inet-labeled-unicast and the NLRI advertised by ASBR2 the loopback interface peer is l2vpn-signaling.

```
user@ASBR1> show bgp neighbor

Peer: 10.0.78.2+65473 AS 65020 Local: 10.0.78.1+179 AS 65010
  Type: External    State: Established    Flags: Sync
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: Cease
  Export: [ loopback ]
  Options: Preference LocalAddress AddressFamily PeerAS Rib-group Refresh

  Address families configured: inet-labeled-unicast
  Local Address: 10.0.78.1 Holdtime: 90 Preference: 170
  Number of flaps: 3
  Last flap event: Stop
  Error: 'Cease' Sent: 1 Recv: 2
  Peer ID: 192.168.10.1     Local ID: 192.168.3.1      Active Holdtime:
90
  Keepalive Interval: 30         Peer index: 0
  BFD: disabled, down
  Local Interface: ge-0/3/0.0
  NLRI for restart configured on peer: inet-labeled-unicast
  NLRI advertised by peer: inet-labeled-unicast
  NLRI for this session: inet-labeled-unicast
```

```
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-labeled-unicast
  NLRI that restart is negotiated for: inet-labeled-unicast
  NLRI of received end-of-rib markers: inet-labeled-unicast
  NLRI of all end-of-rib markers sent: inet-labeled-unicast
  Peer supports 4 byte AS extension (peer-as 65020)
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              2
    Received prefixes:            3
    Accepted prefixes:            3
    Suppressed due to damping:    0
    Advertised prefixes:          3
  Last traffic (seconds): Received 8    Sent 3    Checked 60
  Input messages:  Total 8713   Updates 3       Refreshes 0     Octets
165688
  Output messages: Total 8745   Updates 2       Refreshes 0     Octets
166315
  Output Queue[0]: 0

Peer: 192.168.10.1+51234 AS 65020 Local: 192.168.3.1+179 AS 65010
  Type: External    State: Established    Flags: Sync
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: Cease
  Options: Multihop Preference LocalAddress AddressFamily PeerAS Rib-group
 Refresh
  Address families configured: l2vpn-signaling
  Local Address: 192.168.3.1 Holdtime: 90 Preference: 170
  Number of flaps: 3
  Last flap event: Stop
  Error: 'Cease' Sent: 1 Recv: 2
  Peer ID: 192.168.10.1     Local ID: 192.168.3.1      Active Holdtime:
90
  Keepalive Interval: 30         Peer index: 0
  BFD: disabled, down
  NLRI for restart configured on peer: l2vpn-signaling
  NLRI advertised by peer: l2vpn-signaling
  NLRI for this session: l2vpn-signaling
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: l2vpn-signaling
  NLRI that restart is negotiated for: l2vpn-signaling
  NLRI of received end-of-rib markers: l2vpn-signaling
  NLRI of all end-of-rib markers sent: l2vpn-signaling
  Peer supports 4 byte AS extension (peer-as 65020)
  Table bgp.l2vpn.0 Bit: 20000
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
    Send state: in sync
    Active prefixes:              1
    Received prefixes:            1
    Accepted prefixes:            1
    Suppressed due to damping:    0
```

```
    Advertised prefixes:          1
  Table inter-as.l2vpn.0
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
    Send state: not advertising
    Active prefixes:              1
    Received prefixes:            1
    Accepted prefixes:            1
    Suppressed due to damping:    0
  Last traffic (seconds): Received 19   Sent 18   Checked 42
  Input messages:  Total 8712   Updates 3       Refreshes 0     Octets
165715
  Output messages: Total 8744   Updates 2       Refreshes 0     Octets
166342
  Output Queue[1]: 0
  Output Queue[2]: 0
```

7. On ASBR2, display the BGP summary. Notice that the first peer is the IP address of the Gigabit Ethernet interface of ASBR1, the second peer is the IP address of the lo0 interface of ASBR1, and the third peer is the lo0 interface of PE2. Verify that the state of each peer is Established.

```
user@ASBR2> show bgp summary

Groups: 3 Peers: 3 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State
Pending
inet.0                 3          2          0          0          0
       0
bgp.l2vpn.0            2          2          0          0          0
       0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last
Up/Dwn State|#Active/Received/Accepted/Damped...
10.0.78.1              65010       8781       8748       0       2 2d
17:54:56 Establ
  inet.0: 2/3/3/0
192.168.3.1            65010       8780       8747       0       2 2d
17:54:54 Establ
  bgp.l2vpn.0: 1/1/1/0
  inter-as.l2vpn.0: 1/1/1/0
192.168.11.1           65020       8809       8763       0       1 2d
17:59:22 Establ
  bgp.l2vpn.0: 1/1/1/0
  inter-as.l2vpn.0: 1/1/1/0
```

8. On PE2, display the BGP group. Verify that the peer is the IP address of the lo0 interface of ASBR2. Verify that the number of established peer sessions is 1.

```
user@PE1> show bgp group

Group Type: Internal    AS: 65020                 Local AS: 65020
  Name: core-ibgp       Index: 1                  Flags: Export Eval
  Holdtime: 0
  Total peers: 1        Established: 1
  192.168.10.1+179
  bgp.l2vpn.0: 1/1/1/0
```

```
  inter-as.l2vpn.0: 1/1/1/0

Groups: 1  Peers: 1    External: 0    Internal: 1    Down peers: 0   Flaps:
 7
Table          Tot Paths  Act Paths Suppressed    History Damp State
Pending
bgp.l2vpn.0            1          1          0          0          0
      0
inte.l2vpn.0           1          1          0          0          0
      0
```

Configuring the VPLS Routing Instances

Step-by-Step Procedure To configure the VPLS routing instances:

1. On PE1, configure the VPLS routing instance. To enable a VPLS instance, specify the vpls instance type. Configure VPLS on the CE-facing Gigabit Ethernet interface.

   Configure the CE-facing interface to use ethernet-vpls encapsulation.

   ```
   user@PE1# set routing-instances metro instance-type vpls
   user@PE1# set routing-instances metro interface ge-1/3/0.0
   ```

2. On PE1, configure the VPLS protocol within the routing instance. To uniquely identify the virtual circuit, configure the VPLS identifier. The VPLS identifier uniquely identifies each VPLS in the router. Configure the same VPLS ID on all the routers for a given VPLS.

   Specify the IP address of the lo0 interface on ASBR2 as the neighbor.

   Configure the CE-facing interface to use ethernet-vpls encapsulation and the vpls protocol family.

   ```
   user@PE1# set routing-instances metro protocols vpls vpls-id 101
   user@PE1# set routing-instances metro protocols vpls neighbor 192.168.3.1
   user@PE1# set interfaces ge-1/3/0 encapsulation ethernet-vpls
   user@PE1# set interfaces ge-1/3/0 unit 0 family vpls
   ```

3. On ASBR1, configure the VPLS routing instance. To enable a VPLS instance, specify the vpls instance type. Configure a route distinguisher and a VRF target. The vrf-target statement causes default VRF import and export policies to be generated that accept and tag routes with the specified target community.

NOTE: A route distinguisher allows the router to distinguish between two identical IP prefixes used as VPN routes. Configure a different route distinguisher on each ASBR router.

NOTE: You must configure the same VRF target on both ASBR routers.

```
user@ASBR1# set routing-instances inter-as instance-type vpls
user@ASBR1# set routing-instances inter-as route-distinguisher 65010:1
user@ASBR1# set routing-instances inter-as vrf-target target:2:1
```

4. On ASBR1, configure the VPLS protocol within the routing instance.

   Configure the VPLS identifier. Specify the IP address of the lo0 interface on PE1 as the neighbor.

```
user@ASBR1# set routing-instances inter-as protocols vpls vpls-id 101
user@ASBR1# set routing-instances inter-as protocols vpls neighbor
  192.168.2.1
```

NOTE: The VPLS identifier uniquely identifies each LDP-signaled VPLS in the router. Configure the same VPLS ID on PE1 and ASBR1.

5. On ASBR1, configure the VPLS site within the routing instance. Configure the site identifier as required by the protocol to establish the EBGP pseudowire. As a best practice for more complex topologies involving multihoming, configure a site preference.

```
user@ASBR1# set routing-instances inter-as protocols vpls site ASBR-metro
  site-identifier 1
user@ASBR1# set routing-instances inter-as protocols vpls site ASBR-metro
  site-preference 10000
```

6. On ASBR1, configure the VPLS mesh group peer-as statement within the routing instance to specify which ASs belong to this AS mesh group. Configure the peer AS for the mesh group as all.

   This statement enables the router to establish a single pseudowire between the ASBRs. VPLS NLRI messages are exchanged across the EBGP sessions on the inter-AS links between the ASBRs. All autonomous systems are in one mesh group.

```
user@ASBR1# set routing-instances inter-as protocols vpls mesh-group metro
  peer-as all
```

7. On ASBR2, configure the VPLS routing instance. To enable a VPLS instance, specify the vpls instance type. Configure a route distinguisher and a VRF target. The vrf-target statement causes default VRF import and export policies to be generated that accept and tag routes with the specified target community.

NOTE: A route distinguisher allows the router to distinguish between two identical IP prefixes used as VPN routes. Configure a different route distinguisher on each ASBR router.

NOTE: You must configure the same VRF target community on both ASBR routers.

```
user@ASBR2# set routing-instances inter-as instance-type vpls
user@ASBR2# set routing-instances inter-as route-distinguisher 65020:1
user@ASBR2# set routing-instances inter-as vrf-target target:2:1
```

8. On ASBR2, configure the VPLS site within the routing instance. Configure the site identifier as required by the protocol.

```
user@ASBR2# set routing-instances inter-as protocols vpls site ASBR-core
  site-identifier 2
```

9. On ASBR2, configure the VPLS mesh group within the routing instance to specify which VPLS PEs belong to this AS mesh group. Configure the peer AS for the mesh group as all.

   This statement enables the router to establish a single pseudowire between the ASBRs. VPLS NLRI messages are exchanged across the EBGP sessions on the inter-AS links between the ASBRs. All autonomous systems are in one mesh group.

```
user@ASBR1# set routing-instances inter-as protocols vpls mesh-group core
  peer-as all
```

10. On PE2, configure the VPLS routing instance. To enable a VPLS instance, specify the vpls instance type. Configure VPLS on the CE-facing Gigabit Ethernet interface. Configure a route distinguisher and a VRF target.

```
user@PE2# set routing-instances inter-as instance-type vpls
user@PE2# set routing-instances inter-as interface ge-0/1/1.0
user@PE2# set routing-instances inter-as route-distinguisher 65020:1
user@PE2# set routing-instances inter-as vrf-target target:2:1
```

11. On PE2, configure the VPLS site within the routing instance. Configure the site identifier as required by the protocol.

    Configure the CE-facing interface to use ethernet-vpls encapsulation and the vpls protocol family.

```
user@PE2# set routing-instances inter-as protocols vpls site PE2 site-identifier
  3
user@PE2# set interfaces ge-0/1/1 encapsulation ethernet-vpls
user@PE2# set interfaces ge-0/1/1 unit 0 family vpls
```

12. On each router, commit the configuration:

```
user@host> commit
```

13. On the PE routers, display the CE-facing Gigabit Ethernet interface information and verify that the encapsulation is configured correctly:

```
user@host> show interfaces ge-1/3/0

Address            Interface          Label space ID         Hold time
10.0.23.10         ge-1/3/1.0         192.168.3.1:0            11

Physical interface: ge-1/3/0, Enabled, Physical link is Up
  Interface index: 147, SNMP ifIndex: 145
  Link-level type: Ethernet, MTU: 1514, Speed: 1000mbps, MAC-REWRITE Error:
 None,
  Loopback: Disabled, Source filtering: Disabled, Flow control: Enabled,
  Auto-negotiation: Enabled, Remote fault: Online
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 4 supported, 4 maximum usable queues
  Schedulers     : 256
  Current address: 00:12:1e:ee:34:db, Hardware address: 00:12:1e:ee:34:db

  Last flapped   : 2008-08-27 19:02:52 PDT (5d 22:32 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Ingress rate at Packet Forwarding Engine      : 0 bps (0 pps)
  Ingress drop rate at Packet Forwarding Engine : 0 bps (0 pps)
  Active alarms  : None
  Active defects : None

  Logical interface ge-1/3/0.0 (Index 84) (SNMP ifIndex 146)
    Flags: SNMP-Traps Encapsulation: ENET2
    Input packets : 0
    Output packets: 1
    Protocol inet, MTU: 1500
      Flags: None
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 10.10.11/24, Local: 10.10.11.11, Broadcast:
10.10.11.255
```

Results The relevant sample configuration for the CE1 router follows.

Router CE1

```
interfaces {
  lo0 {
    unit 0 {
      family inet {
        address 192.168.1.1/32 {
          primary;
        }
        address 127.0.0.1/32;
      }
    }
  }
  ge-0/3/0 {
    unit 0 {
```

```
            family inet {
               address 10.10.11.1/24;
            }
         }
      }
   }
```

The relevant sample configuration for the PE1 router follows.

Router PE1

```
interfaces {
   lo0 {
      unit 0 {
         family inet {
            address 192.168.2.1/32 {
               primary;
            }
            address 127.0.0.1/32;
         }
      }
   }
   ge-1/3/0 {
      encapsulation ethernet-vpls;
      unit 0 {
         family vpls;
      }
   }
   ge-1/3/1 {
      unit 0 {
         family inet {
            address 10.0.23.9/30;
         }
         family mpls;
      }
   }
}
routing-options {
   autonomous-system 0.65010;
}
protocols {
   mpls {
      interface ge-1/3/1.0;
   }
   ospf {
      traffic-engineering;
      area 0.0.0.1 {
         interface ge-1/3/1.0;
         interface lo0.0 {
            passive;
         }
      }
   }
   ldp {
      interface ge-1/3/1.0;
      interface lo0.0;
   }
```

```
}
routing-instances {
   metro {
      instance-type vpls;
      interface ge-1/3/0.0;
      protocols {
         vpls {
            vpls-id 101;
            neighbor 192.168.3.1;
         }
      }
   }
}
```

The relevant sample configuration for the ASBR1 router follows.

Router ASBR1

```
interfaces {
   lo0 {
   unit 0 {
      family inet {
         address 192.168.3.1/32 {
            primary;
         }
         address 127.0.0.1/32;
      }
      }
   }
   }
   ge-0/3/0 {
      unit 0 {
         family inet {
            address 10.0.78.1/30;
         }
         family mpls;
      }
   }
   ge-0/3/1 {
      unit 0 {
         family inet {
            address 10.0.23.10/30;
         }
         family mpls;
      }
   }
}
routing-options {
   autonomous-system 0.65010;
}
protocols {
   mpls {
      interface ge-0/3/1.0;
      interface ge-0/3/0.0;
   }
   bgp {
      group vpls-core {
```

```
            type external;
            multihop;
            local-address 192.168.3.1;
            family l2vpn {
               signaling;
            }
            peer-as 65020;
            neighbor 192.168.10.1;
         }
         group metro-core {
            type external;
            local-address 10.0.78.1;
            family inet {
               labeled-unicast {
                  resolve-vpn;
               }
            }
            export loopback;
            peer-as 65020;
            neighbor 10.0.78.2;
         }
      }
      ospf {
         traffic-engineering;
         area 0.0.0.1 {
            interface ge-0/3/1.0;
            interface lo0.0 {
               passive;
            }
         }
      }
      ldp {
         interface ge-0/3/0.0;
         interface ge-0/3/1.0;
         interface lo0.0;
      }
   }
   policy-options {
      policy-statement loopback {
         term term1 {
            from {
               protocol [ ospf direct ];
               inactive: route-filter 10.0.0.0/8 longer;
               route-filter 192.168.0.0/16 longer;
            }
            then accept;
         }
      }
   }
   routing-instances {
      inter-as {
         instance-type vpls;
         route-distinguisher 65010:1;
         vrf-target target:2:1;
         protocols {
            vpls {
```

```
            site ASBR-metro {
               site-identifier 1;
               site-preference 10000;
            }
            vpls-id 101;
            neighbor 192.168.2.1;
            mesh-group metro {
               peer-as {
                  all;
               }
            }
         }
      }
   }
}
```

The relevant sample configuration for the ASBR2 router follows.

Router ASBR2

```
interfaces {
   lo0 {
      unit 0 {
         family inet {
            address 192.168.10.1/32 {
               primary;
            }
            address 127.0.0.1/32;
         }
      }
   }
   ge-3/1/0 {
      unit 0 {
         family inet {
            address 10.0.78.2/30;
         }
         family mpls;
      }
   }
   ge-3/1/1 {
      unit 0 {
         family inet {
            address 10.0.90.13/30;
         }
         family mpls;
      }
   }
}
routing-options {
   autonomous-system 0.65020;
}
protocols {
   mpls {
      interface ge-3/1/0.0;
      interface ge-3/1/1.0;
   }
   bgp {
```

```
        group core-ibgp {
            type internal;
            local-address 192.168.10.1;
            family inet {
                labeled-unicast {
                    resolve-vpn;
                }
            }
            family l2vpn {
                signaling;
            }
            neighbor 192.168.11.1;
        }
        group vpls-metro {
            type external;
            multihop;
            local-address 192.168.10.1;
            family l2vpn {
                signaling;
            }
            peer-as 65010;
            neighbor 192.168.3.1;
        }
        group core-metro {
            type external;
            local-address 10.0.78.2;
            family inet {
                labeled-unicast {
                    resolve-vpn;
                }
            }
            export loopback;
            peer-as 65010;
            neighbor 10.0.78.1;
        }
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface ge-3/1/1.0;
            interface lo0.0 {
                passive;
            }
        }
    }
    ldp {
        interface ge-3/1/0.0;
        interface ge-3/1/1.0;
    }
}
policy-options {
    policy-statement loopback {
        term term1 {
            from {
                protocol [ ospf direct ];
                route-filter 192.168.0.0/16 longer;
```

```
        }
        then accept;
      }
    }
  }
  routing-instances {
    inter-as {
      instance-type vpls;
      route-distinguisher 65020:1;
      vrf-target target:2:1;
      protocols {
        vpls {
          site ASBR-core {
            site-identifier 2;
          }
          mesh-group core {
            peer-as {
              all;
            }
          }
        }
      }
    }
  }
```

The relevant sample configuration for the PE2 router follows.

Router PE2

```
interfaces {
  lo0 {
    unit 0 {
      family inet {
        address 192.168.11.1/32 {
          primary;
        }
        address 127.0.0.1/32;
      }
    }
  }
  ge-0/1/0 {
    unit 0 {
      family inet {
        address 10.0.90.14/30;
      }
      family mpls;
    }
  }
  ge-0/1/1 {
    encapsulation ethernet-vpls;
    unit 0 {
      family vpls;
    }
  }
}
routing-options {
  autonomous-system 0.65020;
```

```
}
protocols {
    mpls {
        interface ge-0/1/0.0;
    }
    bgp {
        group core-ibgp {
            type internal;
            local-address 192.168.11.1;
            family l2vpn {
                signaling;
            }
            neighbor 192.168.10.1;
        }
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface ge-0/1/0.0;
            interface lo0.0 {
                passive;
            }
        }
    }
    ldp {
        interface ge-0/1/0.0;
    }
}
routing-instances {
    inter-as {
        instance-type vpls;
        interface ge-0/1/1.0;
        route-distinguisher 65020:1;
        vrf-target target:2:1;
        protocols {
            vpls {
                site PE2 {
                    site-identifier 3;
                }
            }
        }
    }
}
```

The relevant sample configuration for the CE2 router follows.

Router CE2

```
interfaces {
    lo0 {
        unit 0 {
            family inet {
                address 192.168.12.1/32 {
                    primary;
                }
                address 127.0.0.1/32;
            }
```

```
        }
      }
      ge-0/1/1 {
        unit 0 {
          family inet {
            address 10.10.11.2/24;
          }
        }
      }
    }
```

Verification

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

- Verifying VPLS Connections on page 756
- Verifying End-to-End Traffic Flow on page 758

Verifying VPLS Connections

Purpose To verify the VPLS connections have been established, enter the following command on the ASBR and PE routers.

Action

```
user@PE1> show vpls connections
Layer-2 VPN connections:

Legend for connection status (St)
EI -- encapsulation invalid      NC -- interface encapsulation not CCC/TCC/VPLS
EM -- encapsulation mismatch     WE -- interface and instance encaps not same
VC-Dn -- Virtual circuit down    NP -- interface hardware not present
CM -- control-word mismatch      -> -- only outbound connection is up
CN -- circuit not provisioned    <- -- only inbound connection is up
OR -- out of range               Up -- operational
OL -- no outgoing label          Dn -- down
LD -- local site signaled down   CF -- call admission control failure
RD -- remote site signaled down  SC -- local and remote site ID collision
LN -- local site not designated  LM -- local site ID not minimum designated
RN -- remote site not designated RM -- remote site ID not minimum designated
XX -- unknown connection status  IL -- no incoming label
MM -- MTU mismatch               MI -- Mesh-Group ID not availble
BK -- Backup connection          ST -- Standby connection

Legend for interface status
Up -- operational
Dn -- down

Instance: metro
  VPLS-id: 101
    Neighbor                Type  St     Time last up          # Up trans
    192.168.3.1(vpls-id 101)  rmt   Up     Sep  9 14:05:18 2008           1
      Remote PE: 192.168.3.1, Negotiated control-word: No
      Incoming label: 800001, Outgoing label: 800000
      Local interface: vt-1/2/0.1048576, Status: Up, Encapsulation: ETHERNET
        Description: Intf - vpls metro neighbor 192.168.3.1 vpls-id 101

user@ASBR1> show vpls connections
```

```
...
Instance: inter-as
  BGP-VPLS State
  Mesh-group connections: metro
    Neighbor         Local-site   Remote-site   St      Time last up
    192.168.10.1     1            2             Up      Sep  8 20:16:28 2008
      Incoming label: 800257, Outgoing label: 800000
      Local interface: vt-1/2/0.1049088, Status: Up, Encapsulation: VPLS
  LDP-VPLS State
  VPLS-id: 101
  Mesh-group connections: __ves__
    Neighbor                 Type  St     Time last up          # Up trans
    192.168.2.1(vpls-id 101)  rmt   Up     Sep  9 14:05:22 2008           1
      Remote PE: 192.168.2.1, Negotiated control-word: No
      Incoming label: 800000, Outgoing label: 800001
      Local interface: vt-0/1/0.1049089, Status: Up, Encapsulation: ETHERNET
        Description: Intf - vpls inter-as neighbor 192.168.2.1 vpls-id 101

user@ASBR2> show vpls connections
...
Instance: inter-as
  BGP-VPLS State
  Mesh-group connections: __ves__
    Neighbor         Local-site   Remote-site   St      Time last up
    192.168.11.1     2            3             Up      Sep 11 15:18:23 2008
      Incoming label: 800002, Outgoing label: 800001
      Local interface: vt-4/0/0.1048839, Status: Up, Encapsulation: VPLS
  Mesh-group connections: core
    Neighbor         Local-site   Remote-site   St      Time last up
    192.168.3.1      2            1             Up      Sep  8 20:16:28 2008
      Incoming label: 800000, Outgoing label: 800257
      Local interface: vt-4/0/0.1048834, Status: Up, Encapsulation: VPLS

user@PE2> show vpls connections
...
Instance: inter-as
  Local site: PE2 (3)
    connection-site           Type  St     Time last up          # Up trans
    2                         rmt   Up     Sep  8 20:16:28 2008           1
      Remote PE: 192.168.10.1, Negotiated control-word: No
      Incoming label: 800001, Outgoing label: 800002
      Local interface: vt-0/3/0.1048832, Status: Up, Encapsulation: VPLS
        Description: Intf - vpls inter-as local site 3 remote site 2
```

Meaning In the display from PE1, notice that the neighbor is the lo0 address of ASBR1 and that the status is up.

In the display from ASBR1, notice that the neighbor is the lo0 address of PE1 and that the status is up.

In the display from ASBR2, notice that the neighbor is the lo0 address of PE2 and that the status is up.

In the display from PE2, notice that the neighbor is the lo0 address of ASBR2 and that the status is up.

Verifying End-to-End Traffic Flow

Purpose To verify that the CEs can send and receive traffic across the VPLS, use the ping command.

Action

```
user@CE1> ping 10.10.11.2
PING 10.10.11.2 (10.10.11.2): 56 data bytes
64 bytes from 10.10.11.2: icmp_seq=0 ttl=64 time=1.369 ms
64 bytes from 10.10.11.2: icmp_seq=1 ttl=64 time=1.360 ms
64 bytes from 10.10.11.2: icmp_seq=2 ttl=64 time=1.333 ms
^C

user@CE2> ping 10.10.11.1
PING 10.10.11.1 (10.10.11.1): 56 data bytes
64 bytes from 10.10.11.1: icmp_seq=0 ttl=64 time=6.209 ms
64 bytes from 10.10.11.1: icmp_seq=1 ttl=64 time=1.347 ms
64 bytes from 10.10.11.1: icmp_seq=2 ttl=64 time=1.324 ms
^C
```

Meaning If CE1 can send and receive traffic from CE2 and CE2 can send and receive traffic from CE1, the VPLS is performing correctly.

For More Information

For additional information about VPLS, see the following:

- *JUNOS VPNs Configuration Guide*
- *JUNOS Network Interfaces Configuration Guide*
- *JUNOS Class of Service Configuration Guide*
- *JUNOS Routing Protocols Configuration Guide*
- *JUNOS Routing Protocols and Policies Command Reference*
- *JUNOS Interfaces Command Reference*
- RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*
- RFC 4762, *Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling*
- RFC 4761, *Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling*

Revision History

13 April 2009—9.5R1 Release. Roy Spencer.

15 January 2009—9.4R1 Release. Roy Spencer.

10 October 2008—9.3R1 Release. Roy Spencer.

10 July 2008—9.2R1 Release. Roy Spencer.

10 April 2008—9.1R1 Release. Added support for VPLS nonstop active routing. Roy Spencer.

1 February 2008—Added support for VPLS interworking between LDP and BGP signaling. Added support for 802.1p classification with Ethernet VPLS over ATM LLC interface encapsulation. Added support for VSTP. Added support for Layer 2 VPLS filters for MX960 router. 9.0R1 Release. Fawn Damitio.

29 June 2007—Added support for LDP signaling and MX-series routers. 8.4R1 Release. Fawn Damitio.

12 January 2007—Added configuration information for aggregated Ethernet and multihoming feature. Added support for MX960 Ethernet Services Router, 8.2R1 Release. Fawn Damitio.

15 September 2006—8.1R1 Release. Richard Hendricks.

29 June 2006—Added limitations for MSTP and logical routers, 8.0R1 Release. Richard Hendricks.

27 March 2006—Added support for VPLS on LSI logical interfaces, 7.6R1 Release. Richard Hendricks.

9 January 2006—Added support for multihoming a CE router to multiple PE routers, 7.5R1 Release. Richard Hendricks.

14 September 2005—7.4R1 Release. Richard Hendricks.

13 June 2005—Added VPLS per-packet load balancing, support for limiting MAC address learning per interface in a VPLS domain, and migration to the new VPLS and Layer 2 VPN signaling statement at the [edit protocols bgp groups *group-name* family l2vpn] hierarchy level, 7.3R1 Release. Richard Hendricks.

5 April 2005—7.2R1 Release. Richard Hendricks.

2 February 2005—7.1R1 Release. Richard Hendricks.

6 October 2004—7.0R1 Release. Richard Hendricks.

6 July 2004—Added support for Ethernet VPLS over ATM LLC interface encapsulation on T-series and M320 routing platforms, the show vpls statistics command, and manual selection of tunnel-enabled PICs used to provide virtual ports for VPLS operation, 6.4R1 Release. Richard Hendricks.

5 April 2004—Updated lt interface families and encapsulation types and added new commands to clear MAC addresses from the VPLS table and modify the VPLS table timeout intervals, 6.3R1 Release. Richard Hendricks.

21 January 2004—Added new PIC support for VPLS. Richard Hendricks.

22 December 2003—Added VPLS CoS, VPLS graceful restart, VPLS interinstance bridging and routing, VPLS support on the T-series routing platforms, and operational mode commands for VPLS source and destination MAC accounting, 6.2R1 Release. Richard Hendricks.

22 September 2003—Added VPLS policers and filters, 6.1R1 Release. Richard Hendricks.

30 June 2003—Added the ether-vpls-over-atm-llc interface encapsulation type and LSP selection for VPLS instances, 6.0R1 Release. Elizabeth Lichtenberg and Richard Hendricks.

2 April 2003—Initial document written, 5.7R1. Richard Hendricks.

Part 5
Index

Index

G

I

L

M

N

O

P

R

S

T

V