JUNOS® Software

VPNs Configuration Guide

Release 9.4

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

Part Number: 530-028715-01, Revision 1

JUNOS® Software VPNs Configuration Guide
Release 9.4

Writing: Albert Statti
Editing: Joanne McClintock
Illustration: Faith Bradford
Cover Design: Edmonds Design

Revision History
15 January 2009—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 Layer 3 VPNs

Part 4 Multicast VPNs

Part 8 Indexes

List of Figures

List of Tables

About This Guide

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

Objectives

This guide provides an overview of and describes how to configure the JUNOS software virtual private network (VPN) functions, virtual private LAN service (VPLS) functions, and Layer 2 circuit functions.

NOTE: This guide documents Release 9.4 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 1 on page xxxiv defines notice icons used in this guide.

Table 1: Notice Icons

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

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

Table 2: Text and Syntax Conventions

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

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

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

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

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

List of Technical Publications

Table 3 on page xxxvi lists the software and hardware guides and release notes for Juniper Networks M-series, MX-series, and T-series routing platforms and describes the contents of each document. Table 4 on page xl lists the books included in the *Network Operations Guide* series. Table 5 on page xli lists the manuals and release notes supporting JUNOS software for J-series and SRX-series platforms. All documents are available at http://www.juniper.net/techpubs/.

Table 6 on page xlii 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 3: Technical Documentation for Supported Routing Platforms

Book	Description
JUNOS Software for Supported Routing Platforms	
Access Privilege	Explains how to configure access privileges in user classes by using permission flags and regular expressions. Lists the permission flags along with their associated command-line interface (CLI) operational mode commands and configuration statements.
Broadband Subscriber Management Solutions	Describes residential subscriber management and how you can deploy solutions that include multisubscriber IP address assignment, service provisioning, authentication, authorization, accounting, and dynamic request services in your network
Class of Service	Provides an overview of the class-of-service (CoS) functions of the JUNOS software and describes how to configure CoS features, including configuring multiple forwarding classes for transmitting packets, defining which packets are placed into each output queue, scheduling the transmission service level for each queue, and managing congestion through the random early detection (RED) algorithm.
CLI User Guide	Describes how to use the JUNOS command-line interface (CLI) to configure, monitor, and manage Juniper Networks routing platforms. This material was formerly covered in the *JUNOS System Basics Configuration Guide*.
Feature Guide	Provides a detailed explanation and configuration examples for several of the most complex features in the JUNOS software.
High Availability	Provides an overview of hardware and software resources that ensure a high level of continuous routing platform operation and describes how to configure high availability (HA) features such as nonstop active routing (NSR) and graceful Routing Engine switchover (GRES).
MPLS Applications	Provides an overview of traffic engineering concepts and describes how to configure traffic engineering protocols.
Multicast Protocols	Provides an overview of multicast concepts and describes how to configure multicast routing protocols.

Table 3: Technical Documentation for Supported Routing Platforms *(continued)*

Book	Description
Multiplay Solutions	Describes how you can deploy IPTV and voice over IP (VoIP) services in your network.
MX-series Layer 2 Configuration Guide	Provides an overview of the Layer 2 functions of the MX-series routers, including configuring bridging domains, MAC address and VLAN learning and forwarding, and spanning-tree protocols. It also details the routing instance types used by Layer 2 applications. All of this material was formerly covered in the *JUNOS Routing Protocols Configuration Guide*.
MX-series Layer 2 Solutions Guide	Describes common configuration scenarios for the Layer 2 features supported on the MX-series routers, including basic bridged VLANs with normalized VLAN tags, aggregated Ethernet links, bridge domains, Multiple Spanning Tree Protocol (MSTP), and integrated routing and bridging (IRB).
Network Interfaces	Provides an overview of the network interface functions of the JUNOS software and describes how to configure the network interfaces on the routing platform.
Network Management	Provides an overview of network management concepts and describes how to configure various network management features, such as SNMP and accounting options.
Policy Framework	Provides an overview of policy concepts and describes how to configure routing policy, firewall filters, and forwarding options.
Protected System Domain	Provides an overview of the JCS 1200 platform and the concept of Protected System Domains (PSDs). The JCS 1200 platform, which contains up to six redundant pairs of Routing Engines running JUNOS software, is connected to a T320 router or to a T640 or T1600 routing node. To configure a PSD, you assign any number of Flexible PIC concentrators (FPCs) in the T-series routing platform to a pair of Routing Engines on the JCS 1200 platform. Each PSD has the same capabilities and functionality as a physical router, with its own control plane, forwarding plane, and administration.
Routing Protocols	Provides an overview of routing concepts and describes how to configure routing instances, and unicast routing protocols.
Secure Configuration Guide for Common Criteria and JUNOS-FIPS	Provides an overview of secure Common Criteria and JUNOS-FIPS protocols for the JUNOS software and describes how to install and configure secure Common Criteria and JUNOS-FIPS on a routing platform.
Services Interfaces	Provides an overview of the services interfaces functions of the JUNOS software and describes how to configure the services interfaces on the router.
Software Installation and Upgrade Guide	Describes the JUNOS software components and packaging and explains how to initially configure, reinstall, and upgrade the JUNOS system software. This material was formerly covered in the *JUNOS System Basics Configuration Guide*.

Table 3: Technical Documentation for Supported Routing Platforms *(continued)*

Book	Description
Subscriber Access	Provides an overview of the subscriber access features of the JUNOS software and describes how to configure subscriber access support on the router, including dynamic profiles, class of service, AAA, and access methods.
System Basics	Describes Juniper Networks routing platforms and explains how to configure basic system parameters, supported protocols and software processes, authentication, and a variety of utilities for managing your router on the network.
VPNs	Provides an overview and describes how to configure Layer 2 and Layer 3 virtual private networks (VPNs), virtual private LAN service (VPLS), and Layer 2 circuits. Provides configuration examples.
JUNOS References	
Hierarchy and RFC Reference	Describes the JUNOS configuration mode commands. Provides a hierarchy reference that displays each level of a configuration hierarchy, and includes all possible configuration statements that can be used at that level. This material was formerly covered in the *JUNOS System Basics Configuration Guide*.
Interfaces Command Reference	Describes the JUNOS software operational mode commands you use to monitor and troubleshoot interfaces.
Routing Protocols and Policies Command Reference	Describes the JUNOS software operational mode commands you use to monitor and troubleshoot routing policies and protocols, including firewall filters.
System Basics and Services Command Reference	Describes the JUNOS software operational mode commands you use to monitor and troubleshoot system basics, including commands for real-time monitoring and route (or path) tracing, system software management, and chassis management. Also describes commands for monitoring and troubleshooting services such as class of service (CoS), IP Security (IPsec), stateful firewalls, flow collection, and flow monitoring.
System Log Messages Reference	Describes how to access and interpret system log messages generated by JUNOS software modules and provides a reference page for each message.
J-Web User Guide	
J-Web Interface User Guide	Describes how to use the J-Web graphical user interface (GUI) to configure, monitor, and manage Juniper Networks routing platforms.
JUNOS API and Scripting Documentation	
JUNOScript API Guide	Describes how to use the JUNOScript application programming interface (API) to monitor and configure Juniper Networks routing platforms.
JUNOS XML API Configuration Reference	Provides reference pages for the configuration tag elements in the JUNOS XML API.

Table 3: Technical Documentation for Supported Routing Platforms *(continued)*

Book	Description
JUNOS XML API Operational Reference	Provides reference pages for the operational tag elements in the JUNOS XML API.
NETCONF API Guide	Describes how to use the NETCONF API to monitor and configure Juniper Networks routing platforms.
JUNOS Configuration and Diagnostic Automation Guide	Describes how to use the commit script and self-diagnosis features of the JUNOS software. This guide explains how to enforce custom configuration rules defined in scripts, how to use commit script macros to provide simplified aliases for frequently used configuration statements, and how to configure diagnostic event policies.
Hardware Documentation	
Hardware Guide	Describes how to install, maintain, and troubleshoot routing platforms and components. Each platform has its own hardware guide.
PIC Guide	Describes the routing platform's Physical Interface Cards (PICs). Each platform has its own PIC guide.
DPC Guide	Describes the Dense Port Concentrators (DPCs) for all MX-series routers.
JUNOScope Documentation	
JUNOScope Software User Guide	Describes the JUNOScope software graphical user interface (GUI), how to install and administer the software, and how to use the software to manage routing platform configuration files and monitor routing platform operations.
Advanced Insight Solutions (AIS) Documentation	
Advanced Insight Solutions Guide	Describes the Advanced Insight Manager (AIM) application, which provides a gateway between JUNOS devices and Juniper Support Systems (JSS) for case management and intelligence updates. Explains how to run AI-Scripts on Juniper Networks devices.
Release Notes	
JUNOS Release Notes	Summarize new features and known problems for a particular software release, provide corrections and updates to published JUNOS, JUNOScript, and NETCONF manuals, provide information that might have been omitted from the manuals, and describe upgrade and downgrade procedures.
Hardware Release Notes	Describe the available documentation for the routing platform and summarize known problems with the hardware and accompanying software. Each platform has its own release notes.
JUNOScope Release Notes	Contain corrections and updates to the published JUNOScope manual, provide information that might have been omitted from the manual, and describe upgrade and downgrade procedures.

Table 3: Technical Documentation for Supported Routing Platforms *(continued)*

Book	Description
AIS Release Notes	Summarize AIS new features and guidelines, identify known and resolved problems, provide information that might have been omitted from the manuals, and provide initial setup, upgrade, and downgrade procedures.
AIS AI-Scripts Release Notes	Summarize AI-Scripts new features, identify known and resolved problems, provide information that might have been omitted from the manuals, and provide instructions for automatic and manual installation, including deleting and rolling back.

Table 4: JUNOS Software Network Operations Guides

Book	Description
Baseline	Describes the most basic tasks for running a network using Juniper Networks products. Tasks include upgrading and reinstalling JUNOS software, gathering basic system management information, verifying your network topology, and searching log messages.
Interfaces	Describes tasks for monitoring interfaces. Tasks include using loopback testing and locating alarms.
MPLS	Describes tasks for configuring, monitoring, and troubleshooting an example MPLS network. Tasks include verifying the correct configuration of the MPLS and RSVP protocols, displaying the status and statistics of MPLS running on all routing platforms in the network, and using the layered MPLS troubleshooting model to investigate problems with an MPLS network.
MPLS Log Reference	Describes MPLS status and error messages that appear in the output of the show mpls lsp extensive command. The guide also describes how and when to configure Constrained Shortest Path First (CSPF) and RSVP trace options, and how to examine a CSPF or RSVP failure in a sample network.
MPLS Fast Reroute	Describes operational information helpful in monitoring and troubleshooting an MPLS network configured with fast reroute (FRR) and load balancing.
Hardware	Describes tasks for monitoring M-series and T-series routing platforms.

To configure and operate a J-series Services Router or an SRX-series Services Gateway running JUNOS software, you must also use the configuration statements and operational mode commands documented in JUNOS configuration guides and command references. To configure and operate a WX Integrated Services Module, you must also use WX documentation.

Table 5: JUNOS Software for J-series Services Routers and SRX-series Services Gateways Documentation

Book	Description
J-series and SRX-series Platforms	
JUNOS Software Interfaces and Routing Configuration Guide	Explains how to configure SRX-series and J-series interfaces for basic IP routing with standard routing protocols, ISDN service, firewall filters (access control lists), and class-of-service (CoS) traffic classification.
JUNOS Software Security Configuration Guide	Explains how to configure and manage SRX-series and J-series security services such as stateful firewall policies, IPsec VPNs, firewall screens, Network Address Translation (NAT), Public Key Cryptography, chassis clusters, Application Layer Gateways (ALGs), and Intrusion Detection and Prevention (IDP).
JUNOS Software Administration Guide	Shows how to monitor SRX-series and J-series devices and routing operations, firewall and security services, system alarms and events, and network performance. This guide also shows how to administer user authentication and access, upgrade software, and diagnose common problems.
JUNOS Software CLI Reference	Provides the complete configuration hierarchy available on SRX-series and J-series devices. This guide also describes the configuration statements and operational mode commands unique to these devices.
JUNOS Release Notes	Summarize new features and known problems for a particular release of JUNOS software, including JUNOS software for J-series and SRX-series devices.The release notes also contain corrections and updates to the manuals and software upgrade and downgrade instructions for JUNOS software.
J-series Only	
JUNOS Software Design and Implementation Guide	Provides guidelines and examples for designing and implementing IPsec VPNs, firewalls, and routing on J-series Services Routers running JUNOS software.
J-series Services Routers Quick Start	Explains how to quickly set up a J-series Services Router. This document contains router declarations of conformity.
J-series Services Routers Hardware Guide	Provides an overview, basic instructions, and specifications for J-series Services Routers. This guide explains how to prepare a site, unpack and install the router, replace router hardware, and establish basic router connectivity. This guide contains hardware descriptions and specifications.
JUNOS Software Migration Guide	Provides instructions for migrating an SSG device running ScreenOS software to JUNOS software or upgrading a J-series device to a later version of the JUNOS software.
WXC Integrated Services Module Installation and Configuration Guide	Explains how to install and initially configure a WXC Integrated Services Module in a J-series Services Router for application acceleration.

Table 6: 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.
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.

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
VPN Overview

Chapter 1

VPN Overview

A virtual private network (VPN) consists of two topological areas: the provider's network and the customer's network. The customer's network is commonly located at multiple physical sites and is also private (non-Internet). A customer site would typically consist of a group of routers or other networking equipment located at a single physical location. The provider's network, which runs across the public Internet infrastructure, consists of routers that provide VPN services to a customer's network as well as routers that provide other services. The provider's network connects the various customer sites in what appears to the customer and the provider to be a private network.

To ensure that VPNs remain private and isolated from other VPNs and from the public Internet, the provider's network maintains policies that keep routing information from different VPNs separate. A provider can service multiple VPNs as long as its policies keep routes from different VPNs separate. Similarly, a customer site can belong to multiple VPNs as long as it keeps routes from the different VPNs separate.

This chapter discusses the following topics that provide background information about VPNs:

VPN Standards

The following IETF RFC and Internet drafts describe VPN features:

- RFC 1918, *Address Allocation for Private Internets*
- RFC 4364, *BGP/MPLS IP Virtual Private Networks (VPNs)*
- RFC 4684, *Constrained Route Distribution for Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs)*

You can access Internet RFCs and drafts on the IETF Web site at http://www.ietf.org.

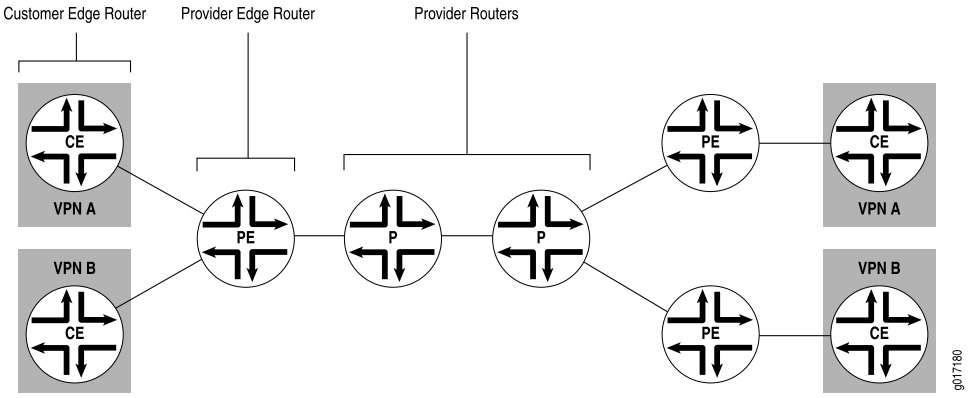
VPN Terminology

VPNs include the following types of network devices (see Figure 1 on page 4):

- Provider edge (PE) routers—Routers in the provider's network that connect to customer edge devices located at customer sites. PE routers support VPN and label functionality. (The label functionality can be provided either by the Resource Reservation Protocol [RSVP] or Label Distribution Protocol [LDP].) Within a single VPN, pairs of PE routers are connected through a tunnel, which can be either a Multiprotocol Label Switching (MPLS) label-switched path (LSP) or an LDP tunnel.
- Provider (P) routers—Routers within the core of the provider's network that are not connected to any routers at a customer site but are part of the tunnel between pairs of PE routers. P routers support MPLS LSP or LDP functionality, but do not need to support VPN functionality.
- Customer edge (CE) devices—Routers or switches located at the customer site that connect to the provider's network. CE devices are typically IP routers, but could also be an Asynchronous Transfer Mode (ATM), Frame Relay, or Ethernet switch.

VPN functionality is provided by the PE routers; the provider and CE routers have no special configuration requirements for VPNs.

Figure 1: Routers in a VPN



Types of VPNs

The JUNOS software provides several types of VPNs; you can choose the best solution for your network environment. Each of the following VPNs has different capabilities and requires different types of configuration:

- Layer 2 VPNs on page 5
- Layer 3 VPNs on page 5
- VPLS on page 6
- Virtual-Router Routing Instances on page 6

Layer 2 VPNs

Implementing a Layer 2 VPN on a router is similar to implementing a VPN using a Layer 2 technology such as ATM or Frame Relay. However, for a Layer 2 VPN on a router, traffic is forwarded to the router in Layer 2 format. It is carried by MPLS over the service provider's network and then converted back to Layer 2 format at the receiving site. You can configure different Layer 2 formats at the sending and receiving sites. The security and privacy of an MPLS Layer 2 VPN are equal to those of an ATM or Frame Relay VPN.

On a Layer 2 VPN, routing occurs on the customer's routers, typically on the CE router. The CE router connected to a service provider on a Layer 2 VPN must select the appropriate circuit on which to send traffic. The PE router receiving the traffic sends it across the service provider's network to the PE router connected to the receiving site. The PE routers do not need to store or process the customer's routes; they only need to be configured to send data to the appropriate tunnel.

For a Layer 2 VPN, customers need to configure their own routers to carry all Layer 3 traffic. The service provider needs to know only how much traffic the Layer 2 VPN needs to carry. The service provider's routers carry traffic between the customer's sites using Layer 2 VPN interfaces. The VPN topology is determined by policies configured on the PE routers.

Layer 3 VPNs

In a Layer 3 VPN, the routing occurs on the service provider's routers. Therefore, Layer 3 VPNs require more configuration on the part of the service provider, because the service provider's PE routers must store and process the customer's routes.

In JUNOS software, Layer 3 VPNs are 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.

VPLS

Virtual private LAN service (VPLS) allows you to connect geographically dispersed customer sites as if they were connected to the same LAN. In many ways, it works like a Layer 2 VPN. VPLS and Layer 2 VPNs use the same network topology and function similarly. A packet originating within a customer's network is sent first to a CE device. It is then sent to a PE router within the service provider's network. The packet traverses the service provider's network over an MPLS LSP. It arrives at the egress PE router, which then forwards the traffic to the CE device at the destination customer site.

The key difference in VPLS is that packets can traverse the service provider's network in a point-to-multipoint fashion, meaning that a packet originating from a CE device can be broadcast to PE routers in the VPLS. In contrast, a Layer 2 VPN forwards packets in a point-to-point fashion only. The destination of a packet received from a CE device by a PE router must be known for the Layer 2 VPN to function properly.

VPLS is designed to carry Ethernet traffic across an MPLS-enabled service provider network. In certain ways, VPLS mimics the behavior of an Ethernet network. When a PE router configured with a VPLS routing instance receives a packet from a CE device, it first checks the appropriate routing table for the destination of the VPLS packet. If the router has the destination, it forwards it to the appropriate PE router. If it does not have the destination, it broadcasts the packet to all the other PE routers that are members of the same VPLS routing instance. The PE routers forward the packet to their CE devices. The CE device that is the intended recipient of the packet forwards it to its final destination. The other CE devices discard it.

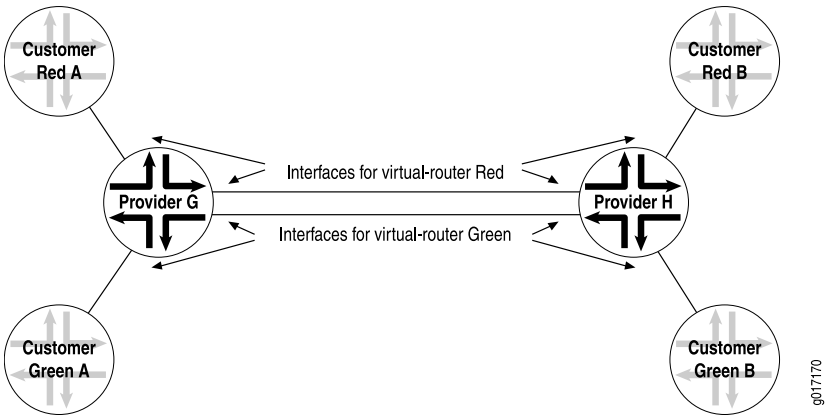
Virtual-Router Routing Instances

A virtual-router routing instance, like a VPN routing and forwarding (VRF) routing instance, maintains separate routing and forwarding tables for each instance. However, many configuration steps required for VRF routing instances are not required for virtual-router routing instances. Specifically, you do not need to configure a route distinguisher, a routing table policy (the vrf-export, vrf-import, and route-distinguisher statements), or MPLS between the P routers.

However, you need to configure separate logical interfaces between each of the service provider routers participating in a virtual-router routing instance. You also need to configure separate logical interfaces between the service provider routers and the customer routers participating in each routing instance. Each virtual-router instance requires its own unique set of logical interfaces to all participating routers.

Figure 2 on page 7 shows how this works. The service provider routers G and H are configured for virtual-router routing instances Red and Green. Each service provider router is directly connected to two local customer routers, one in each routing instance. The service provider routers are also connected to each other over the service provider network. These routers need four logical interfaces: a logical interface to each of the locally connected customer routers and a logical interface to carry traffic between the two service provider routers for each virtual-router instance.

Figure 2: Logical Interface per Router in a Virtual-Router Routing Instance



Layer 3 VPNs do not have this configuration requirement. If you configure several Layer 3 VPN routing instances on a PE router, all the instances can use the same logical interface to reach another PE router. This is possible because Layer 3 VPNs use MPLS (VPN) labels that differentiate traffic going to and from various routing instances. Without MPLS and VPN labels, as in a virtual-router routing instance, you need separate logical interfaces to separate traffic from different instances.

One method of providing this logical interface between the service provider routers is by configuring tunnels between them. You can configure IP Security (IPSec), generic routing encapsulation (GRE), or IP-IP tunnels between the service provider routers, terminating the tunnels at the virtual-router instance.

VPNs and Class of Service

You can configure JUNOS class-of-service (CoS) features to provide multiple classes of service for VPNs. The CoS features are supported on Layer2 VPNs, Layer 3 VPNs, and VPLS. On the router, you can configure multiple forwarding classes for transmitting packets, define which packets are placed into each output queue, schedule the transmission service level for each queue, and manage congestion using a random early detection (RED) algorithm.

VPNs use the standard CoS configuration. For information on how to configure CoS, see the *JUNOS Class of Service Configuration Guide*.

VPNs and Logical Systems

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

Logical systems perform a subset of the actions of a physical router and have their own unique routing tables, interfaces, policies, and routing instances. A set of logical systems within a single router can handle the functions previously performed by several small routers.

You can configure Layer 2 VPNs, Layer 3 VPNs, VPLS, and Layer 2 circuits within a logical system. For more information on logical systems, see the *JUNOS Routing Protocols Configuration Guide*.

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.

VPN Graceful Restart

VPN graceful restart allows a router whose VPN control plane is undergoing a restart to continue to forward traffic while recovering its state from neighboring routers. Without graceful restart, a control plane restart disrupts any VPN services provided by the router.

For VPN graceful restart to function properly, the following items need to be configured on the PE router:

- BGP graceful restart must be active on the PE-to-PE sessions carrying any service-signaling data in the session's network layer reachability information (NLRI).
- Open Shortest Path First (OSPF), Intermediate System-to-Intermediate System (IS-IS), LDP, and RSVP graceful restart must be active, because routes added by these protocols are used to resolve VPN NLRIs.
- For other protocols (static, Routing Information Protocol [RIP], and so on), graceful restart functionality must also be active when these protocols are run between the PE and CE routers. Layer 2 VPNs do not rely on this because protocols are not configured between the PE and CE routers.

In VPN graceful restart, a restarting router completes the following procedures:

- Waits for all the BGP NLRI information from other PE routers before it starts advertising routes to its CE routers.
- Waits for all protocols in all routing instances to converge (or finish graceful restart) before sending CE router information to the other PE routers.
- Waits for all routing instance information (whether it is local configuration or advertisements from a remote peer router) to be processed before sending it to the other PE routers.
- Preserves all forwarding state information in the MPLS routing tables until new labels and transit routes are allocated and then advertises them to other PE routers (and CE routers in carrier-of-carriers VPNs).

Graceful restart is supported on Layer 2 VPNs, Layer 3 VPNs, and virtual-router routing instances.

Redundant Pseudowires for Layer 2 Circuits and VPLS

A redundant pseudowire can act as a backup connection between PE routers and CE devices, maintaining Layer 2 circuit and VPLS services after certain types of failures. This feature can help improve the reliability of certain types of networks (metro for example) where a single point of failure could interrupt service for multiple customers. Redundant pseudowires cannot reduce traffic loss to zero. However, they provide a way to gracefully recover from pseudowire failures in such a way that service can be restarted within a known time limit.

When you configure redundant pseudowires to remote PE routers, you configure one to act as the primary pseudowire over which customer traffic is being transmitted and you configure another pseudowire to act as a backup in the event the primary fails. You configure the two pseudowires statically. A separate label is allocated for the primary and backup neighbors.

For information on how to configure redundant pseudowires, see “Configuring Redundant Pseudowires for Layer 2 Circuits and VPLS” on page 34.

The following sections provide an overview of redundant pseudowires for Layer 2 circuits and VPLS:

Types of Redundant Pseudowire Configurations

You can configure redundant pseudowires for Layer 2 circuits and VPLS in either of the following manners:

- You can configure a single active pseudowire. The PE router configured as the primary neighbor is given preference and this connection is the one used for customer traffic. For the LDP signalling, labels are exchanged for both incoming and outgoing traffic with the primary neighbor. The LDP label advertisement is accepted from the backup neighbor, but no label advertisement is forwarded to it, leaving the pseudowire in an incomplete state. The pseudowire to the backup neighbor is completed only when the primary neighbor fails. The decision to switch between the two pseudowires is made by the device configured with the redundant pseudowires. The primary remote PE router is unaware of the redundant configuration, ensuring that traffic is always switched using just the active pseudowire.
- Alternatively, you can configure two active pseudowires, one to each of the PE routers. Using this approach, control plane signalling is completed and active pseudowires are established with both the primary and backup neighbors. However, the data plane forwarding is done only over a one of the pseudowires (designated as the active pseudowire by the local device). The other pseudowire is on standby. The active pseudowire is preferably established with the primary neighbor and can switch to the backup pseudowire if the primary fails.

The decision to switch between the active and standby pseudowires is controlled by the local device. The remote PE routers are unaware of the redundant connection, and so both remote PE routers send traffic to the local device. The local device only accepts traffic from the active pseudowire and drops the traffic from the standby. In addition, the local device only sends traffic to the active pseudowire. If the active pseudowire fails, traffic is immediately switched to the standby pseudowire.

The two configurations available for pseudowire redundancy have the following limitations:

- For the single active pseudowire configuration, it takes more time (compared to the two active pseudowire configuration) to switchover to the backup pseudowire when a failure is detected. This approach requires additional control plane signalling to complete the pseudowire with the backup neighbor and traffic can be lost during the switchover from primary to backup.
- If you configure two active pseudowires, bandwidth is lost on the link carrying the backup pseudowire between the remote PE router and the local device. Traffic is always duplicated over both the active and standby pseudowires. The single active pseudowire configuration does not waste bandwidth in this fashion.
- You cannot enable GRES (graceful Routing Engine switchover) for redundant pseudowires.
- You cannot enable NSR (nonstop active routing) for redundant pseudowires.

Pseudowire Failure Detection

The following events are used to detect a failure (control and data plane) of the pseudowire configured between a local device and a remote PE router and initiates the switch to a redundant pseudowire:

- Manual switchover (user initiated)
- Remote PE router withdraws the label advertisement
- LSP to the remote PE router goes down
- LDP session with the remote PE router goes down
- Local configuration changes
- Periodic pseudowire OAM procedure fails (Layer 2 circuit-based MPLS ping to the PE router fails)

  When you configure a redundant pseudowire between a CE device and a PE router, a periodic (once a minute) ping packet is forwarded through the active pseudowire to verify data plane connectivity. If the ping fails, traffic is automatically switched to the redundant pseudowire.

When a failure is detected, traffic is switched to the redundant pseudowire which is then also designated as the active pseudowire. The switch is nonreversible, meaning that once traffic has been switched to the redundant pseudowire, it remains active unless it also fails unless the switch to the redundant pseudowire is never done unless there is a failure in the currently active pseudowire. For example, a primary

pseudowire has failed and traffic has been successfully switched to the redundant pseudowire. After a period of time, the cause of the failure of the primary pseudowire has been resolved and it is now possible to reestablish the original connection. However, traffic is not switched back to the original pseudowire unless a failure is detected on the now active pseudowire.

Chapter 2

Configuring VPNs

Layer 2 virtual private networks (VPNs), Layer 3 VPNs, virtual-router routing instances, and virtual private LAN service (VPLS) use a common infrastructure within JUNOS and common configuration procedures. This chapter describes the common configuration steps. Complete these configuration steps, regardless of which type of VPN you are configuring, before proceeding to the more specific configuration steps described in other chapters.

For information on the configuration procedures specific to Layer 2 VPNs, Layer 3 VPNs, and VPLS, see the following configuration chapters:

This chapter describes the general procedures required to configure Layer 2 VPNs, Layer 3 VPNs, virtual-router routing instances, and VPLS, discussing the following topics:

Enabling a Signaling Protocol on the PE Routers

For VPNs to function, you must enable a signaling protocol on the provider edge (PE) routers.

NOTE: As with any configuration involving Multiprotocol Label Switching (MPLS), you cannot configure any of the core-facing interfaces on the PE routers over dense Fast Ethernet Physical Interface Cards (PICs).

To enable a signaling protocol, perform the steps in one of the following sections:

- Using LDP for VPN Signaling on page 14
- Using RSVP for VPN Signaling on page 15

Using LDP for VPN Signaling

To use Label Distribution Protocol (LDP) for VPN signaling, perform the following steps on the PE and provider (P) routers:

1. Configure LDP on the interfaces in the core of the service provider's network by including the ldp statement at the [edit protocols] hierarchy level. You need to configure LDP only on the interfaces between PE routers or between PE and P routers. You can think of these as the "core-facing" interfaces. You do not need to configure LDP on the interface between the PE and customer edge (CE) routers.

   ```
   [edit]
   protocols {
      ldp {
         interface type-fpc/pic/port;
      }
   }
   ```

2. Configure the MPLS address family on the interfaces on which you enabled LDP (the interfaces you configured in Step 1) by including the family mpls statement at the [edit interfaces *type-fpc/pic/port* unit *logical-unit-number*] hierarchy level:

   ```
   [edit]
   interfaces {
      type-fpc/pic/port {
         unit logical-unit-number {
            family mpls;
         }
      }
   }
   ```

3. Configure Open Shortest Path First (OSPF) or Intermediate System-to-Intermediate System (IS-IS) on each PE and P router. You configure these protocols at the master instance of the routing protocol, not within the routing instance used for the VPN.

   To configure OSPF, include the ospf statement at the [edit protocols] hierarchy level. At a minimum, you must configure a backbone area on at least one of the router's interfaces.

   ```
   [edit]
   protocols {
      ospf {
         area 0.0.0.0 {
   ```

```
            interface type-fpc/pic/port;
        }
    }
}
```

To configure IS-IS, include the **isis** statement at the **[edit protocols]** hierarchy level and configure the loopback interface and International Organization for Standardization (ISO) family at the **[edit interfaces]** hierarchy level. At a minimum, you must enable IS-IS on the router, configure a network entity title (NET) on one of the router's interfaces (preferably the loopback interface, **lo0**), and configure the ISO family on all interfaces on which you want IS-IS to run. When you enable IS-IS, Level 1 and Level 2 are enabled by default. The following is the minimum IS-IS configuration. In the **address** statement, *address* is the NET.

```
[edit]
interfaces {
    lo0 {
        unit logical-unit-number {
            family iso {
                address address;
            }
        }
    }
    type-fpc/pic/port {
        unit logical-unit-number {
            family iso;
        }
    }
}
protocols {
    isis {
        interface all;
    }
}
```

For more information about configuring OSPF and IS-IS, see the *JUNOS Routing Protocols Configuration Guide*.

Using RSVP for VPN Signaling

To use the Resource Reservation Protocol (RSVP) for VPN signaling, perform the following steps:

1. On each PE router, configure traffic engineering. To do this, you must configure an interior gateway protocol (IGP) that supports traffic engineering (either IS-IS or OSPF) and enable traffic engineering support for that protocol.

   To enable OSPF traffic engineering support, include the **traffic-engineering** statement at the **[edit protocols ospf]** hierarchy level:

   ```
   [edit protocols ospf]
   traffic-engineering {
       shortcuts;
   }
   ```

For IS-IS, traffic engineering support is enabled by default.

2. On each PE and P router, enable RSVP on the interfaces that participate in the label-switched path (LSP). On the PE router, these interfaces are the ingress and egress points to the LSP. On the P router, these interfaces connect the LSP between the PE routers. Do not enable RSVP on the interface between the PE and the CE routers, because this interface is not part of the LSP.

   To configure RSVP on the PE and P routers, include the **interface** statement at the **[edit protocols rsvp]** hierarchy level. Include one **interface** statement for each interface on which you are enabling RSVP.

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

3. On each PE router, configure an MPLS LSP to the PE router that is the LSP's egress point. To do this, include the **label-switched-path** and **interface** statements at the **[edit protocols mpls]** hierarchy level:

   ```
   [edit protocols]
   mpls {
      label-switched-path path-name {
         to ip-address;
      }
      interface interface-name;
   }
   ```

   In the **to** statement, specify the address of the LSP's egress point, which is an address on the remote PE router.

   In the **interface** statement, specify the name of the interface (both the physical and logical portions). Include one **interface** statement for the interface associated with the LSP.

   When you configure the logical portion of the same interface at the **[edit interfaces]** hierarchy level, you must also configure the **family mpls** and **family inet** statements:

   ```
   [edit interfaces]
   interface-name {
      unit logical-unit-number {
         family inet;
         family mpls;
      }
   }
   ```

4. On all P routers that participate in the LSP, enable MPLS by including the **interface** statement at the **[edit mpls]** hierarchy level. Include one **interface** statement for each connection to the LSP.

   ```
   [edit]
   mpls {
      interface interface-name;
   ```

```
    interface interface-name;
  }
```

5. Enable MPLS on the interface between the PE and CE routers by including the interface statement at the [edit mpls] hierarchy level. Doing this allows the PE router to assign an MPLS label to traffic entering the LSP or to remove the label from traffic exiting the LSP.

   ```
   [edit]
   mpls {
     interface interface-name;
   }
   ```

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

Configuring an IGP on the PE and P Routers

For Layer 2 VPNs, Layer 3 VPNs, virtual-router routing instances, and VPLS to function properly, the service provider's PE and P routers must be able to exchange routing information. To allow them to do this, you must configure either an IGP or static routes on these routers. You configure the IGP on the master instance of the routing protocol process at the [edit protocols] hierarchy level, not within the routing instance used for the VPN—that is, not at the [edit routing-instances] hierarchy level.

When you configure the PE router, do not configure any summarization of the PE router's loopback addresses at the area boundary. Each PE router's loopback address should appear as a separate route.

For information about configuring IGPs and static routes, see the *JUNOS Routing Protocols Configuration Guide*.

Configuring an IBGP Session Between PE Routers

You must configure an internal BGP (IBGP) session between the PE routers to allow the PE routers to exchange information about routes originating and terminating in the VPN. The PE routers rely on this information to determine which labels to use for traffic destined for remote sites.

Configure an IBGP session for the VPN at the [edit protocols bgp group *group-name*] hierarchy level as follows:

```
[edit protocols]
bgp {
  group group-name {
    type internal;
    local-address ip-address;
    family (inet-vpn | inet6-vpn) {
      unicast;
    }
    family l2vpn {
      signaling;
```

```
        }
        neighbor ip-address;
    }
}
```

The IP address in the local-address statement is the address of the loopback interface (lo0) on the local PE router. The IBGP session for the VPN runs through the loopback address. (You must also configure the lo0 interface at the [edit interfaces] hierarchy level.)

The IP address in the neighbor statement is the loopback address of the neighboring PE router. If you are using RSVP signaling, this IP address is the same address you specify in the to statement at the [edit mpls label-switched-path *lsp-path-name*] hierarchy level when you configure the MPLS LSP.

The family statement allows you to configure the IBGP session for either Layer 2 VPNs and VPLS or for Layer 3 VPNs. To configure an IBGP session for Layer 2 VPNs and VPLS, include the signaling statement at the [edit protocols bgp group *group-name* family l2vpn] hierarchy level:

```
[edit protocols bgp group group-name family l2vpn]
signaling;
```

To configure an IPv4 IBGP session for Layer 3 VPNs, configure the unicast statement at the [edit protocols bgp group *group-name* family inet-vpn] hierarchy level:

```
[edit protocols bgp group group-name family inet-vpn]
unicast;
```

To configure an IPv6 IBGP session for Layer 3 VPNs, configure the unicast statement at the [edit protocols bgp group *group-name* family inet6-vpn] hierarchy level:

```
[edit protocols bgp group group-name family inet6-vpn]
unicast;
```

NOTE: You can configure both family inet and family inet-vpn or both family inet6 and family inet6-vpn within the same peer group. This allows you to enable support for both IPv4 and IPv4 VPN routes or both IPv6 and IPv6 VPN routes within the same peer group.

Configuring a VPN Routing Instance on the PE Routers

You need to configure a routing instance for each VPN on each of the PE routers participating in the VPN. The configuration procedures outlined in this section are applicable to Layer 2 VPNs, Layer 3 VPNs, and VPLS. The configuration procedures specific to each type of VPN are described in the corresponding sections in the other configuration chapters.

To configure routing instances for VPNs, include the following statements:

```
description text;
```

```
instance-type type;
interface interface-name;
route-distinguisher (as-number:number | ip-address:number );
vrf-import [ policy-names ];
vrf-export [ policy-names ];
vrf-target {
   export community-name;
   import community-name;
}
```

You can include these statements at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*]

To configure VPN routing instances, you perform the steps in the following sections:

Configuring the Description

To provide a text description for the routing instance, include the description statement. If the text includes one or more spaces, enclose them in quotation marks (" "). Any descriptive text you include is displayed in the output of the show route instance detail command and has no effect on the operation of the routing instance.

To configure a text description, include the description statement:

```
description text;
```

You can include the description statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*]

Configuring the Instance Type

The instance type you configure varies depending on whether you are configuring Layer 2 VPNs, Layer 3 VPNs, VPLS, or virtual routers. Specify the instance type by configuring the instance-type statement:

- To enable Layer 2 VPN routing on a PE router, include the instance-type statement and specify the value l2vpn:

  ```
  instance-type l2vpn;
  ```

- To enable VPLS routing on a PE router, include the instance-type statement and specify the value vpls:

  ```
  instance-type vpls;
  ```

- Layer 3 VPNs require that each PE router have a VPN routing and forwarding (VRF) table for distributing routes within the VPN. To create the VRF table on the PE router, include the instance-type statement and specify the value vrf:

  ```
  instance-type vrf;
  ```

- To enable the virtual-router routing instance, include the instance-type statement and specify the value virtual-router:

  ```
  instance-type virtual-router;
  ```

You can include the instance-type statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*]

Configuring Interfaces for VPN Routing

On each PE router, you must configure an interface over which the VPN traffic travels between the PE and CE routers.

The sections that follow describe how to configure interfaces for VPNs:

General Configuration for VPN Routing

The configuration described in this section applies to all types of VPNs. For Layer 3 VPNs and carrier-of-carriers VPNs, complete the configuration described in this section before proceeding to the interface configuration sections specific to those topics.

To configure interfaces for VPN routing, include the interface statement:

```
interface interface-name;
```

You can include the interface statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*]

Specify both the physical and logical portions of the interface name, in the following format:

```
physical.logical
```

For example, in at-1/2/1.2, at-1/2/1 is the physical portion of the interface name and 2 is the logical portion. If you do not specify the logical portion of the interface name, 0 is set by default.

A logical interface can be associated with only one routing instance. If you enable a routing protocol on all instances by specifying interfaces all when configuring the master instance of the protocol at the [edit protocols] hierarchy level, and if you configure a specific interface for VPN routing at the [edit routing-instances *routing-instance-name*] hierarchy level or at the [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*] hierarchy level, the latter interface statement takes precedence and the interface is used exclusively for the VPN.

If you explicitly configure the same interface name at the [edit protocols] hierarchy level and at either the [edit routing-instances *routing-instance-name*] or [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*] hierarchy levels, an attempt to commit the configuration fails.

Configuring Interfaces for Layer 3 VPNs

When you configure the Layer 3 VPN interfaces at the [edit interfaces] hierarchy level, you must also configure family inet when configuring the logical interface:

```
[edit interfaces]
interface-name {
    unit logical-unit-number {
        family inet;
    }
}
```

Configuring Interfaces for Carrier-of-Carriers VPNs

When you configure carrier-of-carriers VPNs, you need to configure the family mpls statement in addition to the family inet statement for the interfaces between the PE and CE routers. For carrier-of-carriers VPNs, configure the logical interface as follows:

```
[edit interfaces]
interface-name {
    unit logical-unit-number {
        family inet;
```

```
        family mpls;
    }
}
```

If you configure family mpls on the logical interface and then configure this interface for a non-carrier-of-carriers routing instance, the family mpls statement is automatically removed from the configuration for the logical interface, since it is not needed.

Configuring Unicast RPF on VPN Interfaces

For VPN interfaces that carry IP version 4 or version 6 (IPv4 or IPv6) traffic, you can reduce the impact of denial-of-service (DoS) attacks by configuring unicast reverse path forwarding (RPF). Unicast RPF helps determine the source of attacks and rejects packets from unexpected source addresses on interfaces where unicast RPF is enabled.

You can configure unicast RPF on a VPN interface by enabling unicast RPF on the interface and including the interface statement at the [edit routing-instances routing-instance-name] hierarchy level.

You cannot configure unicast RPF on the core-facing interfaces. You can only configure unicast RPF on the CE router-to-PE router interfaces on the PE router. However, for virtual-router routing instances, unicast RPF is supported on all interfaces you specify in the routing instance.

For information on how to configure unicast RPF on VPN interfaces, see the *JUNOS Network Interfaces Configuration Guide*.

Configuring the Route Distinguisher

Each routing instance that you configure on a PE router must have a unique route distinguisher associated with it. VPN and VPLS routing instances need a route distinguisher to help BGP to distinguish between potentially identical network layer reachability information (NLRI) messages received from different VPNs. If you configure different VPN or VPLS routing instances with the same route distinguisher, the commit fails.

To configure a route distinguisher on a PE router, include the route-distinguisher statement:

```
route-distinguisher (as-number:number | ip-address:number);
```

You can include the route-distinguisher statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*]

The route distinguisher is a 6-byte value that you can specify in one of the following formats:

- *as-number*:*number*, where *as-number* is an autonomous system (AS) number (a 2-byte value) and *number* is any 4-byte value. The AS number can be in the range 1 through 65,535. We recommend that you use an Internet Assigned Numbers

Authority (IANA)-assigned, nonprivate AS number, preferably the Internet service provider's (ISP's) own or the customer's own AS number.

- *ip-address*:*number*, where *ip-address* is an IP address (a 4-byte value) and *number* is any 2-byte value. The IP address can be any globally unique unicast address. We recommend that you use the address that you configure in the router-id statement, which is a nonprivate address in your assigned prefix range.

Configuring Automatic Route Distinguishers

If you configure the route-distinguisher-id statement at the [edit routing-options] hierarchy level, a route distinguisher is automatically assigned to the routing instance. If you also configure the route-distinguisher statement in addition to the route-distinguisher-id statement, the value configured for route-distinguisher supersedes the value generated from route-distinguisher-id.

To assign a route distinguisher automatically, include the route-distinguisher-id statement:

```
route-distinguisher-id ip-address;
```

You can include the route-distinguisher-id statement at the following hierarchy levels:

- [edit routing-options]
- [edit logical-systems *logical-system-name* routing-options]

A type 1 route distinguisher is automatically assigned to the routing instance using the format *ip-address*:*number*. The IP address is specified by the route-distinguisher-id statement and the number is unique for the routing instance.

Configuring Policies for the PE Router's VRF Table

On each PE router, you must define policies that define how routes are imported into and exported from the router's VRF table. In these policies, you must define the route target, and you can optionally define the route origin.

To configure policy for the VRF tables, you perform the steps in the following sections:

Configuring the Route Target

As part of the policy configuration for the VPN routing table, you must define a route target, which defines which VPN the route is a part of. When you configure different types of VPN services (Layer 2 VPNs, Layer 3 VPNs, or VPLS) on the same PE router,

be sure to assign unique route target values to avoid the possibility of adding route and signaling information to the wrong VPN routing table.

To configure the route target, include the target option in the community statement:

community *name* members target:*community-id*;

You can include the community statement at the following hierarchy levels:

- [edit policy-options]
- [edit logical-systems *logical-system-name* policy-options]

name is the name of the community.

community-id is the identifier of the community. Specify it in one of the following formats:

- *as-number*:*number*, where *as-number* is an AS number (a 2-byte value) and *number* is a 4-byte community value. The AS number can be in the range 1 through 65,535. We recommend that you use an IANA-assigned, nonprivate AS number, preferably the ISP's own or the customer's own AS number. The community value can be a number in the range 0 through 4,294,967,295 ($2^{32} - 1$).
- *ip-address*:*number*, where *ip-address* is an IPv4 address (a 4-byte value) and *number* is a 2-byte community value. The IP address can be any globally unique unicast address. We recommend that you use the address that you configure in the router-id statement, which is a nonprivate address in your assigned prefix range. The community value can be a number in the range 1 through 65,535.

Configuring the Route Origin

In the import and export policies for the PE router's VRF table, you can optionally assign the route origin (also known as the site of origin) for a PE router's VRF routes using a VRF export policy applied to multiprotocol external BGP (MP-EBGP) VPN IPv4 route updates sent to other PE routers.

Matching on the assigned route origin attribute in a receiving PE's VRF import policy helps ensure that VPN-IPv4 routes learned through MP-EBGP updates from one PE are not reimported to the same VPN site from a different PE connected to the same site.

To configure a route origin, complete the following steps:

1. Include the origin option in the community statement:

   community *name* members origin:*community-id*;

   You can include the community statement at the following hierarchy levels:

   - [edit policy-options]
   - [edit logical-systems *logical-system-name* policy-options]

   name is the name of the community.

community-id is the identifier of the community. Specify it in one of the following formats:

- *as-number*:*number*, where *as-number* is an AS number (a 2-byte value) and *number* is a 4-byte community value. The AS number can be in the range 1 through 65,535. We recommend that you use an IANA-assigned, nonprivate AS number, preferably the ISP's own or the customer's own AS number. The community value can be a number in the range 0 through $2^{32} - 1$.
- *ip-address*:*number*, where *ip-address* is an IPv4 address (a 4-byte value) and *number* is a 2-byte community value. The IP address can be any globally unique unicast address. We recommend that you use the address that you configure in the router-id statement, which is a nonprivate address in your assigned prefix range. The community value can be a number in the range 1 through 65,535.

2. Include the community in the import policy for the PE router's VRF table by configuring the community statement with the *community-id* identifier defined in Step 1 at the [edit policy-options policy-statement *import-policy-name* term *import-term-name* from] hierarchy level. See "Configuring an Import Policy for the PE Router's VRF Table" on page 25.
3. Include the community in the export policy for the PE router's VRF table by configuring the community statement with the *community-id* identifier defined in Step 1 at the [edit policy-options policy-statement *export-policy-name* term *export-term-name* then] hierarchy level. See "Configuring an Export Policy for the PE Router's VRF Table" on page 26.

See "Route Origin for VPNs" on page 51 for a configuration example.

Configuring an Import Policy for the PE Router's VRF Table

Each VPN can have a policy that defines how routes are imported into the PE router's VRF table. An import policy is applied to routes received from other PE routers in the VPN. A policy must evaluate all routes received over the IBGP session with the peer PE router. If the routes match the conditions, the route is installed in the PE router's *routing-instance-name*.inet.0 VRF table. An import policy must contain a second term that rejects all other routes.

Unless an import policy contains only a then reject statement, it must include a reference to a community. Otherwise, when you try to commit the configuration, the commit fails. You can configure multiple import policies.

An import policy determines what to import to a specified VRF table based on the VPN routes learned from the remote PE routers through IBGP. The IBGP session is configured at the [edit protocols bgp] hierarchy level. If you also configure an import policy at the [edit protocols bgp] hierarchy level, the import policies at the [edit policy-options] hierarchy level and the [edit protocols bgp] hierarchy level are combined through a logical AND operation. This allows you to filter traffic as a group.

To configure an import policy for the PE router's VRF table, follow these steps:

1. To define an import policy, include the policy-statement statement. For all PE routers, an import policy must always include the policy-statement statement, at a minimum:

```
policy-statement import-policy-name {
    term import-term-name {
        from {
            protocol bgp;
            community community-id;
        }
        then accept;
    }
    term term-name {
        then reject;
    }
}
```

   You can include the policy-statement statement at the following hierarchy levels:

   - [edit policy-options]
   - [edit logical-systems *logical-system-name* policy-options]

   The *import-policy-name* policy evaluates all routes received over the IBGP session with the other PE router. If the routes match the conditions in the from statement, the route is installed in the PE router's *routing-instance-name*.inet.0 VRF table. The second term in the policy rejects all other routes.

   For more information about creating policies, see the *JUNOS Policy Framework Configuration Guide*.

2. To configure an import policy, include the vrf-import statement:

```
vrf-import import-policy-name;
```

   You can include the vrf-import statement at the following hierarchy levels:

   - [edit routing-instances *routing-instance-name*]
   - [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*]

Configuring an Export Policy for the PE Router's VRF Table

Each VPN can have a policy that defines how routes are exported from the PE router's VRF table. An export policy is applied to routes sent to other PE routers in the VPN. An export policy must evaluate all routes received over the routing protocol session with the CE router. (This session can use the BGP, OSPF, or Routing Information Protocol [RIP] routing protocols, or static routes.) If the routes match the conditions, the specified community target (which is the route target) is added to them and they are exported to the remote PE routers. An export policy must contain a second term that rejects all other routes.

Export policies defined within the VPN routing instance are the only export policies that apply to the VRF table. Any export policy that you define on the IBGP session between the PE routers has no effect on the VRF table. You can configure multiple export policies.

To configure an export policy for the PE router's VRF table, follow these steps:

1. For all PE routers, an export policy must distribute VPN routes to and from the connected CE routers in accordance with the type of routing protocol that you configure between the CE and PE routers within the routing instance.

   To define an export policy, include the policy-statement statement. An export policy must always include the policy-statement statement, at a minimum:

```
policy-statement export-policy-name {
    term export-term-name {
        from protocol (bgp | ospf | rip | static);
        then {
            community add community-id;
            accept;
        }
    }
    term term-name {
        then reject;
    }
}
```

NOTE: Configuring the community add statement is a requirement for Layer 2 VPN VRF export policies.

NOTE: When configuring Draft-rosen multicast VPNs operating in source-specific mode and using the vrf-export statement to specify the export policy, the policy must have a term that accepts routes from the vrf-name.mdt.0 routing table. This term ensures proper PE autodiscovery using the inet-mdt address family.

When configuring Draft-rosen multicast VPNs operating in source-specific mode and using the vrf-target statement, the VRF export policy is automatically generated and automatically accepts routes from the vrf-name.mdt.0 routing table.

You can include the policy-statement statement at the following hierarchy levels:

- [edit policy-options]
- [edit logical-systems *logical-system-name* policy-options]

The *export-policy-name* policy evaluates all routes received over the routing protocol session with the CE router. (This session can use the BGP, OSPF, or RIP routing protocols, or static routes.) If the routes match the conditions in the from statement, the community target specified in the then community add statement

is added to them and they are exported to the remote PE routers. The second term in the policy rejects all other routes.

For more information about creating policies, see the *JUNOS Policy Framework Configuration Guide*.

2. To apply the policy, include the vrf-export statement:

```
vrf-export export-policy-name;
```

You can include the vrf-export statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*]

Applying Both the VRF Export and the BGP Export Policies

When you apply a VRF export policy as described in "Configuring an Export Policy for the PE Router's VRF Table" on page 26, routes from VPN routing instances are advertised to other PE routers based on this policy, where as the BGP export policy is ignored.

If you configure the vpn-apply-export statement, both the VRF export and BGP group or neighbor export policies are applied (VRF first, then BGP) before routes are advertised in the VPN routing tables to other PE routers.

If you configure a PE router as a route reflector or as an AS border router, the behavior enabled by the vpn-apply-export statement is enabled on these routers automatically. For information on how to configure a route reflector or an AS border router, see the *JUNOS Routing Protocols Configuration Guide*.

When you configure the vpn-apply-export statement, be aware of the following:

- Routes imported into the l3vpn.bgp.0 routing table retain the attributes of the original routes (for example, an OSPF route remains an OSPF route even when it is stored in the l3vpn.bgp.0 routing table). You should be aware of this when you configure an export policy for connections between an IBGP PE router and a PE router, a route reflector and a PE router, or AS boundary router (ASBR) peer routers.
- By default, all routes in the l3vpn.bgp.0 routing table are exported to the IBGP peers. If the last statement of the export policy is deny all and if the export policy does not specifically match on routes in the l3vpn.bgp.0 routing table, no routes are exported.

To apply both the VRF export and BGP export policies to VPN routes, include the vpn-apply-export statement:

```
vpn-apply-export;
```

For a list of hierarchy levels at which you can configure this statement, see the statement summary section for this statement.

Configuring a VRF Target

Configuring a VRF target community using the vrf-target statement causes default VRF import and export policies to be generated that accept and tag routes with the specified target community. You can still create more complex policies by explicitly configuring VRF import and export policies. These policies override the default policies generated when you configure the vrf-target statement.

If you do not configure the import and export options of the vrf-target statement, the specified community string is applied in both directions. The import and export keywords give you more flexibility, allowing you to specify a different community for each direction.

The syntax for the VRF target community is not a name. You must specify it in the format target:*x*:*y*. A community name cannot be specified because this would also require you to configure the community members for that community using the policy-options statement. If you define the policy-options statements, then you can just configure VRF import and export policies as usual. The purpose of the vrf-target statement is to simplify the configuration by allowing you to configure most statements at the [edit routing-instances] hierarchy level.

To configure a VRF target, include the vrf-target statement:

```
vrf-target community;
```

You can include the vrf-target statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*]

An example of how you might configure the vrf-target statement follows:

```
[edit routing-instances sample]
vrf-target target:69:102;
```

To configure the vrf-target statement with the export and import options, include the following statements:

```
vrf-target {
    export community-name;
    import community-name;
}
```

You can include the vrf-target statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*]

Configuring BGP Route Target Filtering

BGP route target filtering allows you to distribute VPN routes to only the routers that need them. In VPN networks without BGP route target filtering configured, BGP distributes all VPN routes to all VPN peer routers.

For more information on BGP route target filtering, see RFC 4684, *Constrained Route Distribution for Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs)*.

The following sections provide an overview of BGP route target filtering and how to configure it for VPNs:

BGP Route Target Filtering Overview

PE routers, unless they are configured as route reflectors or are running an EBGP session, discard any VPN routes that do not include a route target extended community as specified in the local VRF import policies. This is the default behavior of the JUNOS software.

However, unless it is explicitly configured not to store VPN routes, any router configured either as a route reflector or border router for a VPN address family must store all of the VPN routes that exist in the service provider's network. Also, though PE routers can automatically discard routes that do not include a route target extended community, route updates continue to be generated and received.

By reducing the number of routers receiving VPN routes and route updates, BGP route target filtering helps to limit the amount of overhead associated with running a VPN. BGP route target filtering is most effective at reducing VPN-related administrative traffic in networks where there are many route reflectors or AS border routers that do not participate in the VPNs directly (not acting as PE routers for the CE devices).

BGP route target filtering uses standard UPDATE messages to distributes route target extended communities between routers. The use of UPDATE messages allows BGP to use its standard loop detection mechanisms, path selection, policy support, and database exchange implementation.

Configuring BGP Route Target Filtering for VPNs

BGP route target filtering is enabled through the exchange of the route-target address family, stored in the bgp.rtarget.0 routing table. Based on the route-target address family, the route target NLRI (address family indicator [AFI] = 1, subsequent AFI [SAFI] = 132) is negotiated with its peers.

On a system that has locally configured VRF instances, BGP automatically generates local routes corresponding to targets referenced in the vrf-import policies.

To configure BGP route target filtering, include the family route-target statement:

```
family route-target {
    advertise-default;
    external-paths number;
    prefix-limit number;
}
```

For a list of hierarchy levels at which you can configure the family route-target statement, see the statement summary section for this statement.

The advertise-default, external-paths, and prefix-limit statements affect the BGP route target filtering configuration as follows:

- The advertise-default statement causes the router to advertise the default route target route (0:0:0/0) and suppress all routes that are more specific. This can be used by a route reflector on BGP groups consisting of neighbors that act as PE routers only. PE routers often need to advertise all routes to the route reflector.

  Suppressing all route target advertisements other than the default route reduces the amount of information exchanged between the route reflector and the PE routers. The JUNOS software further helps to reduce route target advertisement overhead by not maintaining dependency information unless a nondefault route is received.
- The external-paths statement (which has a default value of 1) causes the router to advertise the VPN routes that reference a given route target. The number you specify determines the number of external peer routers (currently advertising that route target) that receive the VPN routes.
- The prefix-limit statement limits the number of prefixes that can be received from a peer router.

The route-target, advertise-default, and external-path statements affect the RIB-OUT state and must be consistent between peer routers that share the same BGP group. The prefix-limit statement affects the receive side only and can have different settings between different peer routers in a BGP group.

For examples illustrating how to configure BGP route target filtering for VPNs, see "VPN Examples" on page 41.

Configuring a Virtual-Router Routing Instance

A virtual-router routing instance, like a VRF routing instance, maintains separate routing and forwarding tables for each instance. However, many of the configuration steps required for VRF routing instances are not required for virtual-router routing instances. Specifically, you do not need to configure a route distinguisher, a routing table policy (the vrf-export, vrf-import, and route-distinguisher statements), or MPLS between the service provider routers.

Configure a virtual-router routing instance by including the following statements:

```
description text;
instance-type virtual-router;
interface interface-name;
protocols { ... }
```

You can include these statements at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*]

The following sections explain how to configure a virtual-router routing instance:

- Configuring a Routing Protocol Between the Service Provider Routers on page 32
- Configuring Logical Interfaces Between Participating Routers on page 33

Configuring a Routing Protocol Between the Service Provider Routers

The service provider routers need to be able to exchange routing information. You can configure the following protocols for the virtual-router routing instance protocols statement configuration at the [edit routing-instances *routing-instance-name*] hierarchy level:

- BGP
- IS-IS
- LDP
- OSPF
- Protocol Independent Multicast (PIM)
- RIP

You can also configure static routes.

IBGP route reflection is not supported for virtual-router routing instances.

If you configure LDP under a virtual-router instance, LDP routes are placed by default in the routing instance's inet.0 and inet.3 routing tables (for example, sample.inet.0 and sample.inet.3). To restrict LDP routes to only the routing instance's inet.3 table, include the no-forwarding statement:

```
no-forwarding;
```

You can include the no-forwarding statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols ldp]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols ldp]

When you restrict the LDP routes to only the inet.3 routing table, the corresponding IGP route in the inet.0 routing table can be redistributed and advertised into other routing protocols.

For information on how to configure routing protocols, see the *JUNOS Routing Protocols Configuration Guide*.

Configuring Logical Interfaces Between Participating Routers

You must configure an interface to each customer router participating in the routing instance and to each P router participating in the routing instance. Each virtual-router routing instance requires its own separate logical interfaces to all P routers participating in the instance. To configure interfaces for virtual-router instances, include the interface statement:

```
interface interface-name;
```

You can include the interface statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*]

Specify both the physical and logical portions of the interface name, in the following format:

```
physical.logical
```

For example, in at-1/2/1.2, at-1/2/1 is the physical portion of the interface name and 2 is the logical portion. If you do not specify the logical portion of the interface name, 0 is set by default.

You must also configure the interfaces at the [edit interfaces] hierarchy level.

One method of providing this logical interface between the provider routers is by configuring tunnels between them. You can configure IP Security (IPSec), generic routing encapsulation (GRE), or IP-IP tunnels between the provider routers, terminating the tunnels at the virtual-router instance.

For information on how to configure tunnels and interfaces, see the *JUNOS Services Interfaces Configuration Guide*.

Configuring Graceful Restart

Graceful restart allows a router whose VPN control plane is undergoing a restart to continue to forward traffic while recovering its state from neighboring routers. Without graceful restart, a control plane restart disrupts any VPN services provided by the router. Graceful restart is supported on Layer 2 VPNs, Layer 3 VPNs, virtual-router routing instances, and VPLS.

To enable VPN graceful restart, include the graceful-restart statement:

```
graceful-restart {
    disable;
    restart-duration time-limit;
}
```

You can configure the restart-duration option at either the global or routing instance level. The routing instance value overrides the global value if both are configured.

To configure the graceful-restart statement globally, include it at the following hierarchy levels:

- [edit routing-options]
- [edit logical-systems *logical-system-name* routing-options]

To configure the graceful-restart statement in the routing instance configuration, include it at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* routing-options]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options]

The restart-duration option sets the period of time that the router waits for a graceful restart to be completed. You can configure a time between 1 through 600 seconds. The default value is 300 seconds. At the end of the configured time period, the router performs a standard restart without recovering its state from the neighboring routers. This disrupts VPN services, but is probably necessary if the router is not functioning normally.

Configuring Redundant Pseudowires for Layer 2 Circuits and VPLS

A redundant pseudowire can act as a backup connection between PE routers and CE devices, maintaining Layer 2 circuit and VPLS services after certain types of failures. This feature can help improve the reliability of certain types of networks (metro for example) where a single point of failure could interrupt service for multiple customers. Redundant pseudowires cannot reduce traffic loss to zero. However, they provide a way to gracefully recover from pseudowire failures in such a way that service can be restarted within a known time limit.

For an overview of how redundant pseudowires work, see “Redundant Pseudowires for Layer 2 Circuits and VPLS” on page 9.

To configure pseudowire redundancy for Layer 2 circuits and VPLS, complete the procedures in the following sections:

- Configuring Pseudowire Redundancy on the PE Router on page 34
- Configuring the Switchover Delay for the Pseudowires on page 35

Configuring Pseudowire Redundancy on the PE Router

You configure pseudowire redundancy on the PE router acting as the egress for the primary and standby pseudowires using the backup-neighbor statement.

To configure pseudowire redundancy on the PE router, include the backup-neighbor statement:

```
backup-neighbor {
    community name;
    psn-tunnel-endpoint address;
    standby;
    virtual-circuit-id number;
```

```
}
```

For a list of hierarchy levels at which you can include this statement, see the statement summary for this statement.

The backup-neighbor statement includes the following configuration options:

- community—Specifies the community for the backup neighbor.
- psn-tunnel-endpoint—Specifies the endpoint address for the packet switched network (PSN) tunnel on the remote PE router. The PSN tunnel endpoint address is the destination address for the LSP on the remote PE router.
- standby—Configures the pseudowire to the specified backup neighbor as the standby. When you configure this statement, traffic flows over both the active and standby pseudowires to the CE device. The CE device drops the traffic from the standby pseudowire, unless the active pseudowire fails. If the active pseudowire fails, the CE device automatically switches to the standby pseudowire.
- virtual-circuit-id—Uniquely identifies the primary and standby Layer 2 circuits. This option is configurable for Layer 2 circuits only.

Configuring the Switchover Delay for the Pseudowires

To configure the time the router waits before switching traffic from the failed primary pseudowire to a backup pseudowire, include the switchover-delay statement:

```
switchover-delay milliseconds;
```

For a list of hierarchy levels at which you can include this statement, see the statement summary for this statement.

Configuring Aggregate Labels for VPNs

Aggregate labels for VPNs allow a Juniper Networks routing platform to aggregate a set of incoming labels (labels received from a peer router) into a single forwarding label that is selected from the set of incoming labels. The single forwarding label corresponds to a single next hop for that set of labels. Label aggregation reduces the number of VPN labels that the router must examine.

For a set of labels to share an aggregate forwarding label, they must belong to the same forwarding equivalence class (FEC). The labeled packets must have the same destination egress interface.

Including the community *community-name* statement with the aggregate-label statement lets you specify prefixes with a common origin community. Set by policy on the peer PE, these prefixes represent an FEC on the peer PE router.

CAUTION: If the target community is set by mistake instead of the origin community, forwarding problems at the egress PE can result. All prefixes from the peer PE will appear to be in the same FEC, resulting in a single inner label for all CE routers behind a given PE in the same VPN.

To work with route reflectors in Layer 3 VPN networks, the Juniper Networks M10i router aggregates a set of incoming labels only when the routes:

- Are received from the same peer router
- Have the same site of origin community
- Have the same next hop

The next hop requirement is important because route reflectors forward routes originated from different BGP peers to another BGP peer without changing the next hop of those routes.

To configure aggregate labels for VPNs, include the aggregate-label statement:

```
aggregate-label {
    community community-name;
}
```

For a list of hierarchy levels at which you can include this statement, see the statement summary for this statement.

For information on how to configure a community, see the *JUNOS Policy Framework Configuration Guide*.

Rewriting Markers and VPNs

A marker reads the current forwarding class and loss priority information associated with a packet and finds the chosen code point from a table. It then writes the code point information into the packet header. Entries in a marker configuration represent the mapping of the current forwarding class into a new forwarding class, to be written into the header.

You define markers in the rewrite rules section of the class-of-service (CoS) configuration hierarchy and reference them in the logical interface configuration. You can configure different rewrite rules to handle VPN traffic and non-VPN traffic. The rewrite rule can be applied to MPLS and IPv4 packet headers simultaneously, making it possible to initialize MPLS experimental (EXP) and IP precedence bits at LSP ingress.

For a detailed example of how to configure rewrite rules for MPLS and IPv4 packets and for more information on how to configure statements at the [edit class-of-service] hierarchy level, see the *JUNOS Class of Service Configuration Guide*

Transmitting Nonstandard BPDUs

Circuit cross-connect (CCC) protocol, Layer 2 circuit, and Layer 2 VPN configurations can transmit nonstandard bridge protocol data units (BPDUs) generated by other vendors' equipment. This is the default behavior on all supported PICs and requires no additional configuration.

The following PICs are supported on T-series and M320 routers and can transmit nonstandard BPDUs:

- 1-port Gigabit Ethernet PIC
- 2-port Gigabit Ethernet PIC
- 4-port Gigabit Ethernet PIC
- 10-port Gigabit Ethernet PIC

Pinging VPNs and Layer 2 Circuits

For testing purposes, you can ping Layer 2 VPNs, Layer 3 VPNs, and Layer 2 circuits by using the ping mpls command. The ping mpls command helps to verify that a VPN or circuit has been enabled. This command tests the integrity of the VPN or Layer 2 circuit connection between the PE routers. It does not test the connection between a PE router and a CE router.

You issue the ping mpls command from the ingress PE router of the VPN or Layer 2 circuit to the egress PE router of the same VPN or Layer 2 circuit. When you execute the ping command, echo requests are sent as MPLS packets.

The payload is a User Datagram Protocol (UDP) packet forwarded to the address 127.0.0.1. The contents of this packet are defined in RFC 4379, *Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures*. The label and interface information for building and sending this information as an MPLS packet is the same as for standard VPN traffic, but the time-to-live (TTL) of the innermost label is set to 1.

When the echo request arrives at the egress PE router, the contents of the packet are checked, and then a reply that contains the correct return is sent by means of UDP. The PE router sending the echo request waits to receive an echo reply after a timeout of 2 seconds (you cannot configure this value).

You must configure MPLS at the [edit protocols mpls] hierarchy level on the egress PE router (the router receiving the MPLS echo packets) to be able to ping the VPN or Layer 2 circuit. You must also configure the address 127.0.0.1/32 on the egress PE router's lo0 interface. If this is not configured, the egress PE router does not have this forwarding entry and therefore simply drops the incoming MPLS pings.

The ping mpls command has the following limitations:

- You cannot ping an IPv6 destination prefix.
- You cannot ping a VPN or Layer 2 circuit from a router that is attempting a graceful restart.
- You cannot ping a VPN or Layer 2 circuit from a logical system.

You can also determine whether an LSP linking two PE routers in a VPN is up by pinging the end point address of the LSP. The command you use to ping an MPLS LSP end point is ping mpls lsp-end-point *address*. This command tells you what type of LSP (RSVP or LDP) terminates at the address specified and whether that LSP is up or down.

For a detailed description of this command, see the *JUNOS Routing Protocols and Policies Command Reference*.

Pinging a Layer 2 VPN

To ping a Layer 2 VPN, use one of the following commands:

- ping mpls l2vpn interface *interface-name*

  You ping an interface configured for the Layer 2 VPN on the egress PE router.

- ping mpls l2vpn instance *l2vpn-instance-name* local-site-id *local-site-id-number* remote-site-id *remote-site-id-number*

  You ping a combination of the Layer 2 VPN routing instance name, the local site identifier, and the remote site identifier to test the integrity of the Layer 2 VPN connection (specified by the identifiers) between the ingress and egress PE routers.

Pinging a Layer 3 VPN

To ping a Layer 3 VPN, use the following command:

ping mpls l3vpn *l3vpn-name* prefix *prefix* <count *count*>

You ping a combination of an IPv4 destination prefix and a Layer 3 VPN name on the egress PE router to test the integrity of the VPN connection between the ingress and egress PE routers. The destination prefix corresponds to a prefix in the Layer 3 VPN. However, the ping tests only whether the prefix is present in a PE router's VRF table. It does not test the connection between a PE router and a CE router.

Pinging a Layer 2 Circuit

To ping a Layer 2 circuit, use one of the following commands:

- ping mpls l2circuit interface *interface-name*

  You ping an interface configured for the Layer 2 circuit on the egress PE router.

- ping mpls l2circuit virtual-circuit neighbor <*prefix*> <*virtual-circuit-id*>

  You ping a combination of the IPv4 prefix and the virtual circuit identifier on the egress PE router to test the integrity of the Layer 2 circuit between the ingress and egress PE routers.

Setting the Forwarding Class of the Ping Packets

When you execute the ping mpls command, the ping packets forwarded to the destination include MPLS labels. It is possible to set the value of the forwarding class for these ping packets by using the exp option with the ping mpls command. For example, to set the forwarding class to 5 when pinging a Layer 3 VPN, issue the following command:

```
ping mpls l3vpn westcoast source 1.1.1.1 prefix 2.2.2.2 exp 5 count 20 detail
```

This command would makes the router attempt to ping the Layer 3 VPN westcoast using ping packets with an EXP forwarding class of 5. The default forwarding class used for the ping mpls command packets is 7.

Configuring a Path MTU Check for VPNs

By default, the maximum transmission unit (MTU) check for VPN routing instances is disabled on M-series routers (except the M320 router) and enabled for the M320, T-series, and J-series routers. On M-series routers, you can configure path MTU checks on the outgoing interfaces for unicast traffic routed on VRF routing instances and on virtual-router routing instances.

When you enable an MTU check, the routing platform sends an Internet Control Message Protocol (ICMP) message when a packet traversing the routing instance exceeds the MTU size and has the do-not-fragment bit set. The ICMP message uses the VRF local address as its source address.

For an MTU check to work in a routing instance, you must both include the vrf-mtu-check statement at the [edit chassis] hierarchy level and assign at least one interface containing an IP address to the routing instance.

For more information on the Path MTU check, see the *JUNOS System Basics Configuration Guide*.

To configure path MTU checks, do the tasks described in the following sections:

- Enabling Path MTU Checks for a VPN Routing Instance on page 39
- Assigning an IP Address to the VPN Routing Instance on page 39

Enabling Path MTU Checks for a VPN Routing Instance

To enable path checks on the outgoing interface for unicast traffic routed on a VRF or virtual-router routing instance, include the vrf-mtu-check statement at the [edit chassis] hierarchy level:

```
[edit chassis]
vrf-mtu-check;
```

Assigning an IP Address to the VPN Routing Instance

To ensure that the path MTU check functions properly, at least one IP address must be associated with each VRF or virtual-router routing instance. If an IP address is not associated with the routing instance, ICMP reply messages cannot be sent.

Typically, the VRF or virtual-router routing instance IP address is drawn from among the IP addresses associated with interfaces configured for that routing instance. If none of the interfaces associated with a VRF or virtual-router routing instance is configured with an IP address, you need to explicitly configure a logical loopback interface with an IP address. This interface must then be associated with the routing

instance. See “Configuring a Logical Unit on the Loopback Interface” on page 168 for details.

Enabling Unicast Reverse-Path Forwarding Check for VPNs

IP spoofing may occur during a denial-of-service (DoS) attack. IP spoofing allows an intruder to pass IP packets to a destination as genuine traffic, when in fact the packets are not actually meant for the destination. This type of spoofing is harmful because it consumes the destination’s resources.

Unicast reverse-path forwarding (RPF) check is a tool to reduce forwarding of IP packets that may be spoofing an address. A unicast RPF check performs a route table lookup on an IP packet’s source address, and checks the incoming interface. The router determines whether the packet is arriving from a path that the sender would use to reach the destination. If the packet is from a valid path, the router forwards the packet to the destination address. If it is not from a valid path, the router discards the packet. Unicast RPF is supported for the IPv4 and IPv6 protocol families, as well as for the virtual private network (VPN) address family. You can also enable unicast RPF within a VPN routing instance.

To enable unicast RPF check, include the unicast-reverse-path statement:

```
unicast-reverse-path (active-paths | feasible-paths);
```

For a list of hierarchy levels at which you can configure this statement, see the statement summary section for this statement.

To consider only active paths during the unicast RPF check, include the active-paths option. To consider all feasible paths during the unicast RPF check, include the feasible-paths option.

The unicast-reverse-path statement is documented in greater detail in the *JUNOS Routing Protocols Configuration Guide* and the *JUNOS Network Interfaces Configuration Guide*.

Chapter 3

VPN Examples

The following examples illustrate how to configure BGP route target filtering for virtual private networks (VPNs):

BGP Route Target Filtering for VPNs Overview

BGP route target filtering is enabled by configuring the family route-target statement at the appropriate BGP hierarchy level. This statement enables the exchange of a new route-target address family, which is stored in the bgp.rtarget.0 routing table.

The following configuration illustrates how you could configure BGP route target filtering for a BGP group titled to_vpn04:

```
[edit]
protocols {
    bgp {
        group to_vpn04 {
            type internal;
            local-address 10.255.14.182;
            peer-as 200;
            neighbor 10.255.14.174 {
                family inet-vpn {
                    unicast;
                }
                family route-target;
            }
        }
    }
}
```

The following configuration illustrates how you could configure a couple of local VPN routing and forwarding (VRF) routing instances to take advantage of the functionality provided by BGP route target filtering. Based on this configuration, BGP would automatically generate local routes corresponding to the route targets referenced in the VRF import policies (note the targets defined by the vrf-target statements).

```
[edit]
routing-instances {
```

```
    vpn1 {
      instance-type vrf;
      interface t1-0/1/2.0;
      vrf-target target:200:101;
      protocols {
        ospf {
          export bgp-routes;
          area 0.0.0.0 {
            interface t1-0/1/2.0;
          }
        }
      }
    }
    vpn2 {
      instance-type vrf;
      interface t1-0/1/2.1;
      vrf-target target:200:102;
      protocols {
        ospf {
          export bgp-routes;
          area 0.0.0.0 {
            interface t1-0/1/2.1;
          }
        }
      }
    }
  }
```

Issue the show route table bgp.rtarget.0 show command to verify the BGP route target filtering configuration:

```
user@host> show route table bgp.rtarget.0
 bgp.rtarget.0: 4 destinations, 6 routes (4 active, 0 holddown, 0 hidden)
 + = Active Route, - = Last Active, * = Both
200:200:101/96
                    *[RTarget/5] 00:10:00
                       Local
200:200:102/96
                    *[RTarget/5] 00:10:00
                       Local
200:200:103/96
                    *[BGP/170] 00:09:48, localpref 100, from 10.255.14.174
                       AS path: I
                     > t3-0/0/0.0
200:200:104/96
                    *[BGP/170] 00:09:48, localpref 100, from 10.255.14.174
                       AS path: I
                     > t3-0/0/0.0
```

The show command display format for route target prefixes is:

AS number:route target extended community/length

The first number represents the autonomous system (AS) of the router that sent this advertisement. The remainder of the display follows the JUNOS show command convention for extended communities.

The output from the show route table bgp-rtarget.0 command displays the locally generated and remotely generated routes.

The first two entries correspond to the route targets configured for the two local VRF routing instances (vpn1 and vpn2):

- 200:200:101/96—Community 200:101 in the vpn1 routing instance
- 200:200:102/96—Community 200:102 in the vpn2 routing instance

The last two entries are prefixes received from a BGP peer:

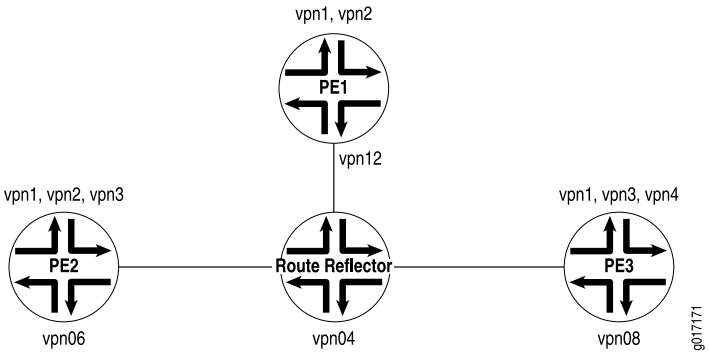
- 200:200:103/96—Tells the local router that routes tagged with this community (200:103) should be advertised to peer 10.255.14.174 through t3-0/0/0.0
- 200:200:104/96—Tells the local router that routes tagged with this community (200:104) should be advertised to peer 10.255.14.174 through t3-0/0/0.0

BGP Route Target Filtering for VPNs

BGP route target filtering reduces the number of routers that receive VPN routes and route updates, helping to limit the amount of overhead associated with running a VPN. BGP route target filtering is most effective at reducing VPN-related administrative traffic in networks where there are many route reflectors or AS border routers that do not participate in the VPNs directly (do not act as PE routers for the CE devices).

Figure 3 on page 43 illustrates the topology for a network configured with BGP route target filtering for a group of VPNs.

Figure 3: BGP Route Target Filtering Enabled for a Group of VPNs



The following sections describe how to configure BGP route target filtering for a group of VPNs:

- Configure BGP Route Target Filtering on Router PE1 on page 44
- Configure BGP Route Target Filtering on Router PE2 on page 45
- Configure BGP Route Target Filtering on the Route Reflector on page 48
- Configure BGP Route Target Filtering on Router PE3 on page 49

Configure BGP Route Target Filtering on Router PE1

This section describes how to enable BGP route target filtering on Router PE1 for this example.

Configure the routing options on router PE1 as follows:

```
[edit]
routing-options {
    route-distinguisher-id 10.255.14.182;
    autonomous-system 200;
}
```

Configure the BGP protocol on Router PE1 as follows:

```
[edit]
protocols {
    bgp {
        group to_VPN_D {
            type internal;
            local-address 10.255.14.182;
            peer-as 200;
            neighbor 10.255.14.174 {
                family inet-vpn {
                    unicast;
                }
                family route-target;
            }
        }
    }
}
```

Configure the vpn1 routing instance as follows:

```
[edit]
routing-instances {
    vpn1 {
        instance-type vrf;
        interface t1-0/1/2.0;
        vrf-target target:200:101;
        protocols {
            ospf {
                export bgp-routes;
                area 0.0.0.0 {
                    interface t1-0/1/2.0;
                }
            }
        }
    }
}
```

Configure the vpn2 routing instance on Router PE1 as follows:

```
[edit]
routing-instances {
    vpn2 {
```

```
        instance-type vrf;
        interface t1-0/1/2.1;
        vrf-target target:200:102;
        protocols {
            ospf {
                export bgp-routes;
                area 0.0.0.0 {
                    interface t1-0/1/2.1;
                }
            }
        }
    }
}
```

Once you have implemented this configuration, you should see the following when you issue a show route table bgp.rtarget.0 command:

```
user@host> show route table bgp.rtarget.0
bgp.rtarget.0: 4 destinations, 6 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

200:200:101/96
                   *[RTarget/5] 00:27:42
                      Local
                    [BGP/170] 00:27:30, localpref 100, from
10.255.14.174
                      AS path: I
                    > via t3-0/0/0.0
200:200:102/96
                   *[RTarget/5] 00:27:42
                      Local
                    [BGP/170] 00:27:30, localpref 100, from
10.255.14.174
                      AS path: I
                    > via t3-0/0/0.0
200:200:103/96
                   *[BGP/170] 00:27:30, localpref 100, from
10.255.14.174
                      AS path: I
                    > via t3-0/0/0.0
200:200:104/96
                   *[BGP/170] 00:27:30, localpref 100, from
10.255.14.174
                      AS path: I
                    > via t3-0/0/0.0
```

Configure BGP Route Target Filtering on Router PE2

This section describes how to enable BGP route target filtering on Router PE2 for this example.

Configure the routing options on Router PE2 as follows:

```
[edit]
routing-options {
    route-distinguisher-id 10.255.14.176;
    autonomous-system 200;
```

```
}
```

Configure the BGP protocol on Router PE2 as follows:

```
[edit]
protocols {
    bgp {
        group to_vpn04 {
            type internal;
            local-address 10.255.14.176;
            peer-as 200;
            neighbor 10.255.14.174 {
                family inet-vpn {
                    unicast;
                }
                family route-target;
            }
        }
    }
}
```

Configure the vpn1 routing instance on Router PE2 as follows:

```
[edit]
routing-instances {
    vpn1 {
        instance-type vrf;
        interface t3-0/0/0.0;
        vrf-target target:200:101;
        protocols {
            bgp {
                group vpn1 {
                    type external;
                    peer-as 101;
                    as-override;
                    neighbor 10.49.11.2;
                }
            }
        }
    }
}
```

Configure the vpn2 routing instance on Router PE2 as follows:

```
[edit]
routing-instances {
    vpn2 {
        instance-type vrf;
        interface t3-0/0/0.1;
        vrf-target target:200:102;
        protocols {
            bgp {
                group vpn2 {
                    type external;
                    peer-as 102;
                    as-override;
```

```
            neighbor 10.49.21.2;
          }
        }
      }
    }
  }
```

Configure the vpn3 routing instance on Router PE2 as follows:

```
[edit]
routing-instances {
  vpn3 {
    instance-type vrf;
    interface t3-0/0/0.2;
    vrf-import vpn3-import;
    vrf-export vpn3-export;
    protocols {
      bgp {
        group vpn3 {
          type external;
          peer-as 103;
          as-override;
          neighbor 10.49.31.2;
        }
      }
    }
  }
}
```

Once you have configured router PE2 in this manner, you should see the following when you issue the show route table bgp.rtarget.0 command:

```
user@host> show route table bgp.rtarget.0
 bgp.rtarget.0: 4 destinations, 7 routes (4 active, 0 holddown, 0 hidden)
 + = Active Route, - = Last Active, * = Both

 200:200:101/96
                    *[RTarget/5] 00:28:15
                       Local
                     [BGP/170] 00:28:03, localpref 100, from
 10.255.14.174
                       AS path: I
                     > via t1-0/1/0.0
 200:200:102/96
                    *[RTarget/5] 00:28:15
                       Local
                     [BGP/170] 00:28:03, localpref 100, from
 10.255.14.174
                       AS path: I
                     > via t1-0/1/0.0
 200:200:103/96
                    *[RTarget/5] 00:28:15
                       Local
                     [BGP/170] 00:28:03, localpref 100, from
 10.255.14.174
                       AS path: I
                     > via t1-0/1/0.0
 200:200:104/96
```

```
                    *[BGP/170] 00:28:03, localpref 100, from
10.255.14.174
                      AS path: I
                    > via t1-0/1/0.0
```

Configure BGP Route Target Filtering on the Route Reflector

This section illustrates how to enable BGP route target filtering on the route reflector for this example.

Configure the routing options on the route reflector as follows:

```
[edit]
routing-options {
   route-distinguisher-id 10.255.14.174;
   autonomous-system 200;
}
```

Configure the BGP protocol on the route reflector as follows:

```
[edit]
protocols {
   bgp {
      group rr-group {
         type internal;
         local-address 10.255.14.174;
         cluster 10.255.14.174;
         peer-as 200;
         neighbor 10.255.14.182 {
            description to_PE1_vpn12;
            family inet-vpn {
               unicast;
            }
            family route-target;
         }
         neighbor 10.255.14.176 {
            description to_PE2_vpn06;
            family inet-vpn {
               unicast;
            }
            family route-target;
         }
         neighbor 10.255.14.178 {
            description to_PE3_vpn08;
            family inet-vpn {
               unicast;
            }
            family route-target;
         }
      }
   }
}
```

Once you have configured the route reflector in this manner, you should see the following when you issue the show route table bgp.rtarget.0 command:

```
user@host> show route table bgp.rtarget.0
 bgp.rtarget.0: 4 destinations, 8 routes (4 active, 0 holddown, 0 hidden)
 + = Active Route, - = Last Active, * = Both

 200:200:101/96
                    *[BGP/170] 00:29:03, localpref 100, from
 10.255.14.176
                      AS path: I
                     > via t1-0/2/0.0
                     [BGP/170] 00:29:03, localpref 100, from
 10.255.14.178
                      AS path: I
                     > via t3-0/1/1.0
                     [BGP/170] 00:29:03, localpref 100, from
 10.255.14.182
                      AS path: I
                     > via t3-0/1/3.0
 200:200:102/96
                    *[BGP/170] 00:29:03, localpref 100, from
 10.255.14.176
                      AS path: I
                     > via t1-0/2/0.0
                     [BGP/170] 00:29:03, localpref 100, from
 10.255.14.182
                      AS path: I
                     > via t3-0/1/3.0
 200:200:103/96
                    *[BGP/170] 00:29:03, localpref 100, from
 10.255.14.176
                      AS path: I
                     > via t1-0/2/0.0
                     [BGP/170] 00:29:03, localpref 100, from
 10.255.14.178
                      AS path: I
                     > via t3-0/1/1.0
 200:200:104/96
                    *[BGP/170] 00:29:03, localpref 100, from
 10.255.14.178
                      AS path: I
                     > via t3-0/1/1.0
```

Configure BGP Route Target Filtering on Router PE3

The following section describes how to enable BGP route target filtering on Router PE3 for this example.

Configure the routing options on Router PE3 as follows:

```
[edit]
routing-options {
    route-distinguisher-id 10.255.14.178;
    autonomous-system 200;
}
```

Configure the BGP protocol on Router PE3 as follows:

```
[edit]
protocols {
```

```
    bgp {
        group to_vpn04 {
            type internal;
            local-address 10.255.14.178;
            peer-as 200;
            neighbor 10.255.14.174 {
                family inet-vpn {
                    unicast;
                }
                family route-target;
            }
        }
    }
}
```

Configure the **vpn1** routing instance on Router PE3 as follows:

```
[edit]
routing-instances {
    vpn1 {
        instance-type vrf;
        interface t3-0/0/0.0;
        vrf-target target:200:101;
        protocols {
            rip {
                group vpn1 {
                    export bgp-routes;
                    neighbor t3-0/0/0.0;
                }
            }
        }
    }
}
```

Configure the **vpn3** routing instance on Router PE3 as follows:

```
[edit]
routing-instances {
    vpn3 {
        instance-type vrf;
        interface t3-0/0/0.1;
        vrf-target target:200:103;
        protocols {
            rip {
                group vpn3 {
                    export bgp-routes;
                    neighbor t3-0/0/0.1;
                }
            }
        }
    }
}
```

Configure the **vpn4** routing instance on Router PE3 as follows:

```
[edit]
```

```
routing-instances {
   vpn4 {
      instance-type vrf;
      interface t3-0/0/0.2;
      vrf-target target:200:104;
      protocols {
         rip {
            group vpn4 {
               export bgp-routes;
               neighbor t3-0/0/0.2;
            }
         }
      }
   }
}
```

Once you have configured Router PE3 in this manner, you should see the following when you issue the show route table bgp.rtarget.0 command:

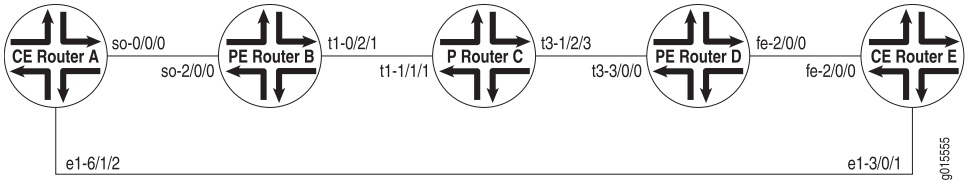
```
user@host> show route table bgp.rtarget.0
bgp.rtarget.0: 4 destinations, 7 routes (4 active, 0 holddown, 0 hidden)
 + = Active Route, - = Last Active, * = Both

 200:200:101/96
                    *[RTarget/5] 00:29:42
                       Local
                     [BGP/170] 00:29:30, localpref 100, from
 10.255.14.174
                       AS path: I
                     > via t3-0/0/1.0
 200:200:102/96
                    *[BGP/170] 00:29:29, localpref 100, from
 10.255.14.174
                       AS path: I
                     > via t3-0/0/1.0
 200:200:103/96
                    *[RTarget/5] 00:29:42
                       Local
                     [BGP/170] 00:29:30, localpref 100, from
 10.255.14.174
                       AS path: I
                     > via t3-0/0/1.0
 200:200:104/96
                    *[RTarget/5] 00:29:42
                       Local
                     [BGP/170] 00:29:30, localpref 100, from
 10.255.14.174
                       AS path: I
                     > via t3-0/0/1.0
```

Route Origin for VPNs

You can use route origin to prevent routes learned from one customer edge (CE) router marked with origin community from being advertised back to it from another CE router in the same AS.

In the example, the route origin is used to prevent routes learned from CE Router A that are marked with origin community from being advertised back to CE Router E by AS 200. The example topology is shown in Figure 4 on page 52.

Figure 4: Network Topology of Site of Origin Example



In this topology, CE Router A and CE Router E are in the same AS (AS200). They use external BGP (EBGP) to exchange routes with their respective provider edge (PE) routers, PE Router B and PE Router D. The two CE routers have a back connection.

The following sections describe how to configure the route origin for a group of VPNs:

- Configuring the Site of Origin Community on CE Router A on page 52
- Configuring the Community on CE Router A on page 53
- Applying the Policy Statement on CE Router A on page 53
- Configuring the Policy on PE Router D on page 54
- Configuring the Community on PE Router D on page 54
- Applying the Policy on PE Router D on page 54

Configuring the Site of Origin Community on CE Router A

The following section describes how to configure CE Router A to advertise routes with a site of origin community to PE Router B for this example.

NOTE: In this example, direct routes are configured to be advertised, but any route can be configured.

Configure a policy to advertise routes with my-soo community on CE Router A as follows:

```
[edit]
policy-options {
   policy-statement export-to-my-isp {
      term a {
         from {
            protocol direct;
         }
         then {
            community add my-soo;
            accept;
         }
      }
   }
```

```
}
```

Configuring the Community on CE Router A

Configure the my-soo community on CE Router A as follows:

```
[edit]
policy-options {
   community my-soo {
      members origin:100:1;
   }
}
```

Applying the Policy Statement on CE Router A

Apply the export-to-my-isp policy statement as an export policy to the EBGP peering on the CE Router A as follows:

```
[edit]
protocols {
   bgp {
      group my_isp {
         export export-to-my-isp;
      }
   }
}
```

When you issue the show route receive-protocol bgp detail command, you should see the following routes originated from PE Router B with my-soo community:

```
user@host> show route receive-protocol bgp 10.12.99.2 detail
inet.0: 16 destinations, 16 routes (15 active, 0 holddown, 1 hidden)
inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
vpn_blue.inet.0: 8 destinations, 10 routes (8 active, 0 holddown, 0 hidden)
* 10.12.33.0/30 (2 entries, 1 announced)
     Nexthop: 10.12.99.2
     AS path: 100 I
     Communities: origin:100:1
  10.12.99.0/30 (2 entries, 1 announced)
     Nexthop: 10.12.99.2
     AS path: 100 I
     Communities: origin:100:1
* 10.255.71.177/32 (1 entry, 1 announced)
     Nexthop: 10.12.99.2
     AS path: 100 I
     Communities: origin:100:1
* 192.168.64.0/21 (1 entry, 1 announced)
     Nexthop: 10.12.99.2
     AS path: 100 I
     Communities: origin:100:1
iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
mpls.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
bgp.l3vpn.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
__juniper_private1__.inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0
hidden)
```

Configuring the Policy on PE Router D

Configure a policy on PE Router D that prevents routes with my-soo community tagged by CE Router A from being advertised to CE Router E as follows:

```
[edit]
policy-options {
   policy-statement soo-ce1-policy {
      term a {
         from {
            community my-soo;
            then {
               reject;
            }
         }
      }
   }
}
```

Configuring the Community on PE Router D

Configure the community on PE Router D as follows:

```
[edit]
policy-options {
   community my-soo {
      members origin:100:1;
   }
}
```

Applying the Policy on PE Router D

To prevent routes learned from CE Router A from being advertised to CE Router E (the two routers can communicate these routes directly), apply the soo-ce1-policy policy statement as an export policy to the PE Router D and CE Router E EBGP session vpn_blue.

View the EBGP session on PE Router D using the show routing-instances command.

```
user@host# show routing-instances
vpn_blue {
    instance-type vrf;
    interface fe-2/0/0.0;
    vrf-target target:100:200;
    protocols {
        bgp {
            group ce2 {
                advertise-peer-as;
                peer-as 100;
                neighbor 10.12.99.6;
            }
        }
```

```
    }
}
```

Apply the soo-ce1-policy policy statement as an export policy to the PE Router D and CE Router E EBGP session vpn_blue as follows:

```
[edit routing-instances]
vpn_blue {
    protocols {
        bgp {
            group ce2{
                export soo-ce1-policy;
            }
        }
    }
}
```

Chapter 4

Summary of VPN Configuration Statements

This chapter summarizes the statements used in the configuration of virtual private networks (VPNs) and virtual private LAN service (VPLS). The statements are organized alphabetically.

Statements configured at the [edit routing-instances] and the [edit protocols] hierarchy levels are explained in complete detail in the *JUNOS Routing Protocols Configuration Guide*.

Statements configured at the [edit policy-options] hierarchy level are explained in complete detail in the *JUNOS Policy Framework Configuration Guide*.

Statements configured at the [edit interfaces] hierarchy level are explained in complete detail in the *JUNOS Network Interfaces Configuration Guide*.

aggregate-label

Syntax

```
aggregate-label {
   community community-name;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols bgp family inet labeled-unicast],
[edit logical-systems *logical-system-name* protocols bgp family inet6 labeled-unicast],
[edit logical-systems *logical-system-name* protocols bgp family inet-vpn unicast],
[edit logical-systems *logical-system-name* protocols bgp family inet-vpn6 unicast],
[edit protocols bgp family inet labeled-unicast],
[edit protocols bgp family inet6 labeled-unicast],
[edit protocols bgp family inet-vpn unicast],
[edit protocols bgp family inet6-vpn unicast]

Release Information

Statement introduced before JUNOS Release 7.4.

Description

Specify matching criteria (in the form of a community) such that all routes which match are assigned the same VPN label, selected from one of the several routes in the set defined by this criteria. This reduces the number of VPN labels that the router must consider, and aggregates the received labels.

Options

community *community-name*—Specify the name of the community to which to apply the aggregate label.

Usage Guidelines

See "Configuring Aggregate Labels for VPNs" on page 35.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

backup-neighbor

Syntax

```
backup-neighbor address {
    community name;
    psn-tunnel-endpoint address;
    standby;
    virtual-circuit-id number;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls neighbor *address*],
[edit protocols l2circuit neighbor *address* interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols vpls neighbor *address*],

Release Information

Statement introduced in JUNOS Release 9.2.

Description

Configures pseudowire redundancy for Layer 2 circuits and VPLS. A redundant pseudowire can act as a backup connection between a PE router and a CE device, maintaining Layer 2 circuit and VPLS services after certain types of failures. This feature can help improve the reliability of certain types of networks where a single point of failure could interrupt service for multiple customers.

Options

community—Specifies the community for the backup neighbor.

psn-tunnel-endpoint—Specifies the endpoint address for the packet switched network (PSN) tunnel on the remote PE router. The PSN tunnel endpoint address is the destination address for the LSP on the remote PE router.

standby—Configures the pseudowire to the specified backup neighbor as the standby. When you configure this statement, traffic flows over both the active and standby pseudowires to the CE device. The CE device drops the traffic from the standby pseudowire, unless the active pseudowire fails. If the active pseudowire fails, the CE device automatically switches to the standby pseudowire.

virtual-circuit-id—Uniquely identifies the primary and standby Layer 2 circuits. This option is configurable for Layer 2 circuits only.

The community, psn-tunnel-endpoint, and virtual-circuit-id statements are explained in detail in "Summary of Layer 2 Circuit Configuration Statements" on page 557.

Usage Guidelines

See "Configuring Pseudowire Redundancy on the PE Router" on page 34.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

description

Syntax description *text*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*], [edit routing-instances *routing-instance-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Describe the VPN or VPLS routing instance.

Options *text*—Provide a text description. If the text includes one or more spaces, enclose the text in quotation marks (" "). Any descriptive text you include is displayed in the output of the show route instance detail command and has no effect on operation.

Usage Guidelines See "Configuring the Description" on page 19.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

family route-target

Syntax

```
family route-target {
    advertise-default;
    external-paths number;
    prefix-limit number;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*]

Release Information

Statement introduced before JUNOS Release 7.4.

Description

Enable BGP route target filtering on the Layer 3 VPN.

Options

advertise-default—Cause the router to advertise the default route target route (0:0:0/0) and suppress all routes that are more specific. This can be used by a route reflector on BGP groups consisting of neighbors that act as provider edge (PE) routers only. PE routers often need to advertise all routes to the route reflector. Suppressing all route target advertisements other than the default route reduces the amount of information exchanged between the route reflector and the PE routers. The JUNOS software further helps to reduce route target advertisement overhead by not maintaining dependency information unless a nondefault route is received.

external-paths *number*—Cause the router to advertise the VPN routes that reference a given route target. The number you specify with the external-paths statement determines the number of external peer routers (currently advertising that route target) that receive the VPN routes. The default value is 1.

prefix-limit *number*—The number of prefixes that can be received from a peer router.

Usage Guidelines

See "Configuring BGP Route Target Filtering for VPNs" on page 30.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

graceful-restart

Syntax

```
graceful-restart {
    disable;
    restart-duration time-limit;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options],
[edit logical-systems *logical-system-name* routing-options],
[edit routing-instances *routing-instance-name* routing-options],
[edit routing-options]

Release Information Statement introduced before JUNOS Release 7.4.

Description Allow a router whose VPN control plane is undergoing a restart to continue to forward traffic while recovering its state from neighboring routers.

Options disable—Disable graceful restart.

restart-duration *time-limit*—Grace period for graceful restart, in seconds.
Default: 300 seconds
Range: 1 through 600 seconds

Usage Guidelines See "Configuring Graceful Restart" on page 33.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

instance-type

Syntax instance-type *type*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*], [edit routing-instances *routing-instance-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Define the type of routing instance.

Options *type*—Can be one of the following:

- l2vpn—Enable a Layer 2 VPN on the routing instance. You must configure the interface, route-distinguisher, vrf-import, and vrf-export statements for this type of routing instance.
- virtual-router—Enable a virtual router routing instance. You must configure the interface statement for this type of routing instance. You do not need to configure the route-distinguisher, vrf-import, and vrf-export statements.
- vpls—Enable VPLS on the routing instance. You must configure the interface, route-distinguisher, vrf-import, and vrf-export statements for this type of routing instance.
- vrf—VPN routing and forwarding (VRF) instance. Required to create a Layer 3 VPN. Create a VRF table (*instance-name*.inet.0) that contains the routes originating from and destined for a particular Layer 3 VPN. You must configure the interface, route-distinguisher, vrf-import, and vrf-export statements for this type of routing instance.

Usage Guidelines See “Configuring the Instance Type” on page 20.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

interface

Syntax interface *interface-name*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*],
[edit routing-instances *routing-instance-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Interface over which the VPN traffic travels between the PE router and customer edge (CE) router. You configure the interface on the PE router. If the instance-type statement is configured as vrf (see instance-type), this statement is required.

Options *interface-name*—Name of the interface.

Usage Guidelines See "Configuring Interfaces for VPN Routing" on page 20.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

no-forwarding

Syntax no-forwarding;

Hierarchy Level [edit logical-systems *logical-system-name* protocols ldp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols ldp],
[edit protocols ldp],
[edit routing-instances *routing-instance-name* protocols ldp]

Release Information Statement introduced before JUNOS Release 7.4.

Description Do not add ingress routes to the inet.0 routing table even if traffic-engineering bgp-igp (configured at the [edit protocols mpls] hierarchy level) is enabled.

Default The no-forwarding statement is disabled. Ingress routes are added to the inet.0 routing table instead of the inet.3 routing table when traffic-engineering bgp-igp is enabled.

Usage Guidelines See "Configuring a Routing Protocol Between the Service Provider Routers" on page 32.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

route-distinguisher

Syntax route-distinguisher (*as-number*:*id* | *ip-address*:*id*);

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*], [edit routing-instances *routing-instance-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Identifier attached to a route that distinguishes to which VPN or VPLS routing instance it belongs. Each routing instance must have a unique distinguisher associated with it. Each route distinguisher is a 6-byte value.

Options *as-number*:*id*—Specify your assigned autonomous system number (*as-number* a 2-byte value) and a 4–byte value for the *id*. The AS number can be in the range from 1 through 65,535.

ip-address:*id*—Specify an IP address (*ip-address* a 4–byte value) within your assigned prefix range and a 2–byte value for the *id*. The IP address can be any globally unique unicast address.

Usage Guidelines See "Configuring the Route Distinguisher" on page 22.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

route-distinguisher-id

Syntax route-distinguisher-id *ip-address*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-options], [edit routing-options]

Release Information Statement introduced before JUNOS Release 7.4.

Description Automatically assign a route distinguisher to the routing instance. If you configure the route-distinguisher statement in addition to the route-distinguisher-id statement, the value configured for route-distinguisher supersedes the value generated from route-distinguisher-id.

Options *ip-address*—Address for routing instance.

Usage Guidelines See "Configuring the Route Distinguisher" on page 22.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

switchover-delay

Syntax switchover-delay *milliseconds*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls neighbor *address*],
[edit protocols l2circuit neighbor *address* interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols vpls neighbor *address*],

Release Information Statement introduced in JUNOS Release 9.2.

Description After the primary pseudowire goes down, specifies the delay (in milliseconds) to wait before the backup pseudowire takes over. You configure this statement for each backup neighbor configuration to adjust the switchover time after a failure is detected.

Options *milliseconds*—Specify the time to wait before switching to the backup pseudowire after the primary pseudowire fails.
Default: 10,000 milliseconds
Range: 0 through 180,000 milliseconds

Usage Guidelines See "Configuring the Switchover Delay for the Pseudowires" on page 35.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

unicast-reverse-path

Syntax unicast-reverse-path (active-paths | feasible-paths);

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options forwarding-table],
[edit logical-systems *logical-system-name* routing-options forwarding-table],
[edit routing-instances *routing-instance-name* routing-options forwarding-table],
[edit routing-options forwarding-table]

Release Information Statement introduced before JUNOS 7.4. Statement added at the [edit routing-instances] hierarchy level in JUNOS 8.3.

Description Enable unicast reverse-path-forwarding check.

Options active-paths—Consider only active paths during the unicast RPF check.

feasible-paths—Consider all feasible paths during the unicast RPF check.

Usage Guidelines See "Enabling Unicast Reverse-Path Forwarding Check for VPNs" on page 40

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

vpn-apply-export

Syntax vpn-apply-export;

Hierarchy Level [edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *neighbor*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *neighbor*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Apply both the VRF export and BGP group or neighbor export policies (VRF first, then BGP) before routes from the vrf or l2vpn routing tables are advertised to other PE routers.

Usage Guidelines See "Applying Both the VRF Export and the BGP Export Policies" on page 28.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

vrf-export

Syntax vrf-export [*policy-names*];

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*],
[edit routing-instances *routing-instance-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Specify how routes are exported from the local PE router's VRF table (*routing-instance-name*.inet.0) to the remote PE router. If the instance-type statement is configured as vrf (see instance-type), this statement is required.

You can configure multiple export policies on the PE router.

Options *policy-names*—Names for the export policies.

Usage Guidelines See "Configuring an Export Policy for the PE Router's VRF Table" on page 26.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

vrf-import

Syntax vrf-import [*policy-names*];

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*], [edit routing-instances *routing-instance-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Specify how routes are imported into the local PE router's VRF table (*routing-instance-name*.inet.0) from the remote PE router. If the instance-type statement is configured as vrf (see instance-type), this statement is required.

You can configure multiple import policies on the PE router.

Options *policy-names*—Names for the import policies.

Usage Guidelines See "Configuring an Import Policy for the PE Router's VRF Table" on page 25.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

vrf-target

Syntax
```
vrf-target {
    community;
    import community-name;
    export community-name;
}
```

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*], [edit routing-instances *routing-instance-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Specify a VRF target community. If you configure the *community* option only, default VRF import and export policies are generated that accept and tag routes with the specified target community. The purpose of the vrf-target statement is to simplify the configuration by allowing you to configure most statements at the [edit routing-instances] hierarchy level.

You can still create more complex policies by explicitly configuring VRF import and export policies using the import and export options.

Options *community*—Community name.

import *community-name*—Communities accepted from neighbors.

export *community-name*—Communities sent to neighbors.

Usage Guidelines See "Configuring a VRF Target" on page 29.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

vrf-mtu-check

Syntax
```
vrf-mtu-check;
```

Hierarchy Level [edit chassis]

Release Information Statement introduced before JUNOS Release 7.4.

Description Enable path checks on the outgoing interface for unicast traffic routed on a VRF or virtual-router routing instance.

Usage Guidelines See "Configuring a Path MTU Check for VPNs" on page 39.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Part 2
Layer 2 VPNs

Chapter 5

Layer 2 VPN Overview

This chapter provides an overview of Layer 2 Multiprotocol Label Switching (MPLS) virtual private networks (VPNs) as they are implemented in the JUNOS software.

For information about the different types of VPNs, see "VPN Overview" on page 3.

This chapter discusses the following topics that provide background information about Layer 2 VPNs:

Layer 2 VPN Overview

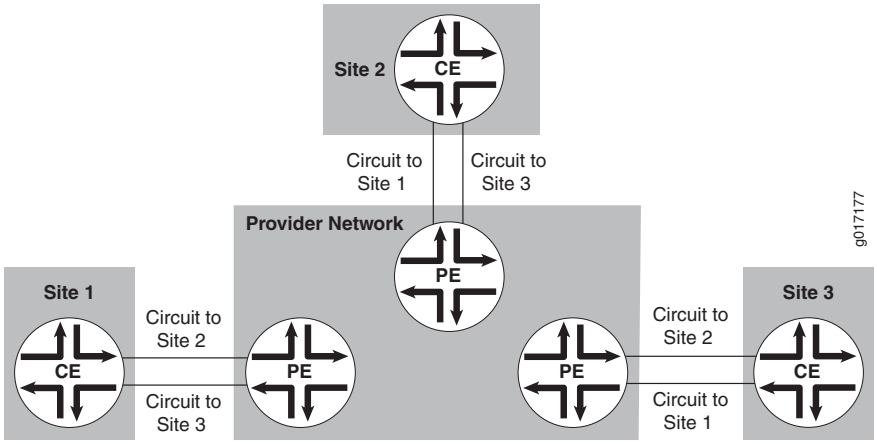
Implementing a Layer 2 VPN on a router is similar to implementing a VPN using a Layer 2 technology such as Asynchronous Transfer Mode (ATM) or Frame Relay. However, for a Layer 2 VPN on a router, traffic is forwarded to the router in a Layer 2 format. It is carried by MPLS over the service provider's network, and then converted back to Layer 2 format at the receiving site. You can configure different Layer 2 formats at the sending and receiving sites. The security and privacy of an MPLS Layer 2 VPN are equal to those of an ATM or Frame Relay VPN.

On a Layer 2 VPN, routing occurs on the customer's routers, typically on the customer edge (CE) router. The CE router connected to a service provider on a Layer 2 VPN must select the appropriate circuit on which to send traffic. The provider edge (PE) router receiving the traffic sends it across the service provider's network to the PE router connected to the receiving site. The PE routers do not need to store or process the customer's routes; they only need to be configured to send data to the appropriate tunnel.

For a Layer 2 VPN, customers need to configure their own routers to carry all Layer 3 traffic. The service provider needs to know only how much traffic the Layer 2 VPN will need to carry. The service provider's routers carry traffic between the customer's sites using Layer 2 VPN interfaces. The VPN topology is determined by policies configured on the PE routers.

Customers need to know only which VPN interfaces connect to which of their own sites. Figure 5 on page 74 illustrates a Layer 2 VPN in which each site has a VPN interface linked to each of the other customer sites.

Figure 5: Layer 2 VPN Connecting CE Routers



Implementing a Layer 2 MPLS VPN includes the following benefits:

- Service providers do not have to invest in separate Layer 2 equipment to provide Layer 2 VPN service. A Layer 2 MPLS VPN allows you to provide Layer 2 VPN service over an existing IP and MPLS backbone.
- You can configure the PE router to run any Layer 3 protocol in addition to the Layer 2 protocols.
- Customers who prefer to maintain control over most of the administration of their own networks might want Layer 2 VPN connections with their service provider instead of a Layer 3 VPN.

Layer 2 VPN Standards

The JUNOS software substantially supports the following Layer 2 VPN Internet draft: draft-kompella-ppvpn-l2vpn-03.txt, *Layer 2 VPN Over Tunnels*.

You can access Internet RFCs and drafts on the IETF Web site at http://www.ietf.org.

Chapter 6

Configuring Layer 2 VPNs

To configure Layer 2 virtual private network (VPN) functionality, you must enable Layer 2 VPN support on the provider edge (PE) router. You must also configure PE routers to distribute routing information to the other PE routers in the VPN and configure the circuits between the PE routers and the customer edge (CE) routers.

Each Layer 2 VPN is configured under a routing instance of type l2vpn. An l2vpn routing instance can transparently carry Layer 3 traffic across the service provider's network. As with other routing instances, all logical interfaces belonging to a Layer 2 VPN routing instance are listed under that instance.

The configuration of the CE routers is not relevant to the service provider. The CE routers need to provide only appropriate Layer 2 circuits (with appropriate circuit identifiers, such as data-link connection identifier [DLCI], virtual path identifier/virtual channel identifier [VPI/VCI], or virtual LAN [VLAN] ID) to send traffic to the PE router.

To configure Layer 2 VPNs, include the following statements:

```
description text;
instance-type l2vpn;
interface interface-name;
route-distinguisher (as-number:id| ip-address:id);
vrf-export [ policy-names ];
vrf-import [ policy-names ];
vrf-target {
   community;
   import community-name;
   export community-name;
}
protocols {
   l2vpn {
      (control-word | no-control-word);
      encapsulation-type type;
      traceoptions {
         file filename <files number> <size size> <world-readable | no-world-readable>;
         flag flag <flag-modifier> <disable>;
      }
      site site-name {
         site-identifier identifier;
         site-preference preference-value {
            backup;
            primary;
         }
```

```
            interface interface-name {
                description text;
                remote-site-id remote-site-id;
            }
        }
    }
}
```

You can include these statements at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*]

For Layer 2 VPNs, only some of the statements in the [edit routing-instances] hierarchy are valid. For the full hierarchy, see the *JUNOS Routing Protocols Configuration Guide*.

In addition to these statements, you must configure Multiprotocol Label Switching (MPLS) label-switched paths (LSPs) between the PE routers, internal BGP (IBGP) sessions between the PE routers, and an interior gateway protocol (IGP) on the PE and provider (P) routers. You must also configure the statements that are required for all types of VPN configuration. See “Configuring VPNs” on page 13 for more information.

By default, Layer 2 VPNs are disabled.

Many of the configuration procedures for Layer 2 VPNs are identical to the procedures for Layer 3 VPNs and virtual private LAN service (VPLS). These procedures are described in detail in “Configuring VPNs” on page 13.

The following sections describe how to configure Layer 2 VPNs:

Configuring the Connections to the Local Site

For each local site, the PE router advertises a set of VPN labels to the other PE routers servicing the Layer 2 VPN. The VPN labels constitute a single block of contiguous labels; however, to allow for reprovisioning, more than one such block can be advertised. Each label block consists of a label base, a range (the size of the block), and a remote site ID that identifies the sequence of remote sites that connect to the local site using this label block (the remote site ID is the first site identifier in the sequence). The encapsulation type is also advertised along with the label block.

The following sections explain how to configure the connections to the local site on the PE router:

Configuring a Layer 2 VPN Routing Instance

To configure a Layer 2 VPN on your network, you need to configure a Layer 2 VPN routing instance on the PE router by including the l2vpn statement:

```
l2vpn {
    (control-word | no-control-word);
    encapsulation-type type;
    traceoptions {
        file filename <files number> <size size> <world-readable | no-world-readable>;
        flag flag <flag-modifier> <disable>;
    }
    site site-name {
        site-identifier identifier;
        site-preference preference-value {
            backup;
            primary;
        }
        interface interface-name {
            description text;
            remote-site-id remote-site-id;
        }
    }
}
```

You can include this statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols]

Instructions for how to configure the remaining statements are included in the sections that follow.

Configuring the Site

All the Layer 2 circuits provisioned for a local site are listed as the set of logical interfaces (using the interface statement) within the site statement.

On each PE router, you must configure each site that has a circuit to the PE router. To do this, include the site statement:

```
site site-name {
    site-identifier identifier;
    site-preference preference-value {
        backup;
        primary;
    }
    interface interface-name {
        description text;
        remote-site-id remote-site-ID;
    }
}
```

You include the site statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols l2vpn]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn]

You must configure the following for each site:

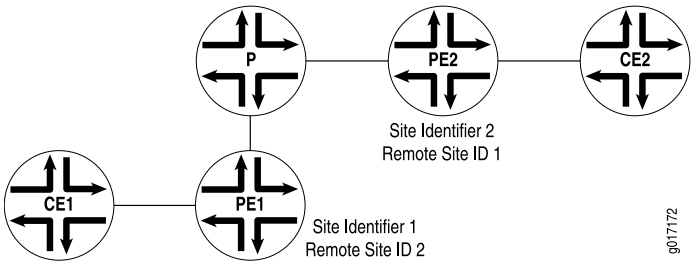
- *site-name*—Name of the site.
- site-identifier *identifier*—Unsigned 16-bit number greater than zero that uniquely identifies the site. The site identifier should correspond to a remote site ID configured on another site within the same VPN.
- interface *interface-name*—The name of the interface and, optionally, a remote site ID for remote site connections. See "Configuring the Remote Site ID" on page 78.

Configuring the Remote Site ID

The remote site ID allows you to configure a sparse Layer 2 VPN topology. A sparse topology means that each site does not have to connect to all the other sites in the VPN; thus it is unnecessary to allocate circuits for all the remote sites. Remote site IDs are particularly important if you configure a topology more complicated than full-mesh, such as a hub-and-spoke topology.

The remote site ID (configured with the remote-site-id statement) corresponds to the site ID (configured with the site-identifier statement) configured at a separate site. Figure 6 on page 78 illustrates the relationship between the site identifier and the remote site ID.

Figure 6: Relationship Between the Site Identifier and the Remote Site ID



As illustrated by the figure, the configuration for Router PE1 connected to Router CE1 is as follows:

```
site-identifier 1;
interface so-0/0/0 {
   remote-site-id 2;
}
```

The configuration for Router PE2 connected to Router CE2 is as follows:

```
site-identifier 2;
interface so-0/0/1 {
   remote-site-id 1;
}
```

The remote site ID (2) on Router PE1 corresponds to the site identifier (2) on Router PE2. On Router PE2, the remote site ID (1) corresponds to the site identifier (1) on Router PE1.

To configure the remote site ID, include the remote-site-id statement:

```
remote-site-id remote-site-id;
```

You can include the remote-site-id statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols l2vpn site *site-name* interface *interface-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn site *site-name* interface *interface-name*]

If you do not explicitly include the remote-site-id statement for the interface configured at the [edit routing-instances *routing-instance-name* protocols l2vpn site *site-name*] hierarchy level, a remote site ID is assigned to that interface.

The remote site ID for an interface is automatically set to 1 higher than the remote site ID for the previous interface. The order of the interfaces is based on their site-identifier statements. For example, if the first interface in the list does not have a remote site ID, its ID is set to 1. The second interface in the list has its remote site ID set to 2, and the third has its remote site ID set to 3. The remote site IDs of any interfaces that follow are incremented in the same manner if you do not explicitly configure them.

Configuring the Encapsulation Type

The encapsulation type you configure at each Layer 2 VPN site varies depending on which Layer 2 protocol you choose to configure. If you configure ethernet-vlan as the encapsulation type, you need to use the same protocol at each Layer 2 VPN site.

You do *not* need to use the same protocol at each Layer 2 VPN site if you configure any of the following encapsulation types:

- atm-aal5—Asynchronous Transfer Mode (ATM) Adaptation Layer (AAL5)
- atm-cell—ATM cell relay

- atm-cell-port-mode—ATM cell relay port promiscuous mode
- atm-cell-vc-mode—ATM virtual circuit (VC) cell relay nonpromiscuous mode
- atm-cell-vp-mode—ATM virtual path (VP) cell relay promiscuous mode
- cisco-hdlc—Cisco Systems-compatible High-Level Data Link Control (HDLC)
- ethernet—Ethernet
- ethernet-vlan—Ethernet virtual LAN (VLAN)
- frame-relay—Frame Relay
- frame-relay-port-mode—Frame Relay port mode
- interworking—Layer 2.5 interworking VPN
- ppp—Point-to-Point Protocol (PPP)

If you configure different protocols at your Layer 2 VPN sites, you need to configure a translational cross-connect (TCC) encapsulation type. For more information, see "Configuring TCC Encapsulation on Interfaces" on page 83.

To configure the Layer 2 protocol accepted by the PE router, specify the encapsulation type by including the encapsulation-type statement:

```
encapsulation-type type;
```

You can include the encapsulation-type statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols l2vpn]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn]

Configuring a Site Preference and Layer 2 VPN Multihoming

You can specify the preference value advertised for a particular Layer 2 VPN site. The site preference value is encoded in the BGP local preference attribute. When a PE router receives multiple advertisements with the same CE device identifier, the advertisement with the highest local preference value is preferred.

You can also use the site-preference statement to enable multihoming for Layer 2 VPNs. Multihoming allows you to connect a CE device to multiple PE routers. In the event that a connection to the primary PE router fails, traffic can be automatically switched to the backup PE router.

To configure a site preference for a Layer 2 VPN, include the site-preference statement:

```
site-preference preference-value {
    backup;
    primary;
}
```

You can configure this statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols l2vpn site *site-name*]

- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn site *site-name*]

You can also specify either the backup option or the primary option for the site-preference statement. The backup option specifies the preference value as 1, the lowest possible value, ensuring that the Layer 2 VPN site is the least likely to be selected. The primary option specifies the preference value as 65,535, the highest possible value, ensuring that the Layer 2 VPN site is the most likely to be selected.

For Layer 2 VPN multihoming configurations, specifying the primary option for a Layer 2 VPN site designates the connection from the PE router to the CE device as the preferred connection if the CE device is also connected to another PE router. Specifying the backup option for a Layer 2 VPN site designates the connection from the PE router to the CE device as the secondary connection if the CE device is also connected to another PE router.

Tracing Layer 2 VPN Traffic and Operations

To trace Layer 2 VPN protocol traffic, you can specify options in the Layer 2 VPN traceoptions statement:

```
traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
}
```

You can configure the traceoptions statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols l2vpn]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn]

The following trace flags display the operations associated with Layer 2 VPNs:

- all—All Layer 2 VPN tracing options.
- connections—Layer 2 connections (events and state changes).
- error—Error conditions.
- general—General events.
- nlri—Layer 2 advertisements received or sent by means of the BGP.
- normal—Normal events.
- policy—Policy processing.
- route—Routing information.
- state—State transitions.
- task—Routing protocol task processing.

- timer—Routing protocol timer processing.
- topology—Layer 2 VPN topology changes caused by reconfiguration or advertisements received from other PE routers using BGP.

Disabling Normal TTL Decrementing for VPNs

To diagnose networking problems related to VPNs, it can be useful to disable normal time-to-live (TTL) decrementing. In JUNOS, you can do this with the no-propagate-ttl and no-decrement-ttl statements. However, when you are tracing VPN traffic, only the no-propagate-ttl statement is effective.

For the no-propagate-ttl statement to have an effect on VPN behavior, you need to clear the PE-router-to-PE-router BGP session, or disable and then enable the VPN routing instance.

For more information about the no-propagate-ttl and no-decrement-ttl statements, see the *JUNOS MPLS Applications Configuration Guide*.

Configuring CCC Encapsulation on Interfaces

You need to specify a circuit cross-connect (CCC) encapsulation type for each PE-router-to-CE-router interface running a Layer 2 VPN. This encapsulation type should match the encapsulation type configured under the routing instance. For information about how to configure the encapsulation type under the routing instance, see "Configuring the Encapsulation Type" on page 79.

NOTE: A Layer 2 VPN or Layer 2 circuit is not supported if the PE-router-to-P-router interface has VLAN-tagging enabled and uses a nonenhanced Flexible PIC Concentrator (FPC).

For Layer 2 VPNs, you need to configure the CCC encapsulation on the logical interface. You also need to configure an encapsulation on the physical interface. The physical interface encapsulation does not have to be a CCC encapsulation. However, it should match the logical interface encapsulation. For example, if you configure an ATM CCC encapsulation type on the logical interface, you should configure a compatible ATM encapsulation on the physical interface.

To configure the CCC encapsulation type, include the encapsulation-type statement:

```
encapsulation-type ccc-encapsulation-type;
```

To configure the CCC encapsulation type on the physical interface, include the encapsulation-type statement at the following hierarchy levels:

- [edit interfaces *interface-name*]
- [edit logical-systems *logical-system-name* interfaces *interface-name*]

To configure the CCC encapsulation type on the logical interface, include the encapsulation-type statement at the following hierarchy levels:

- [edit interfaces *interface-name* unit *logical-unit-number*]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]

You configure the encapsulation type at the [edit interfaces] hierarchy level differently from the [edit routing-instances] hierarchy level. For example, you specify the encapsulation as frame-relay at the [edit routing-instances] hierarchy level and as frame-relay-ccc at the [edit interfaces] hierarchy level.

You can run both standard Frame Relay and CCC Frame Relay on the same device. If you specify Frame Relay encapsulation (frame-relay-ccc) for the interface, you should also configure the encapsulation at the [edit interfaces *interface name* unit *unit-number*] hierarchy level as frame-relay-ccc. Otherwise, the logical interface unit defaults to standard Frame Relay.

For more information on how to configure interfaces and interface encapsulations, see the *JUNOS Network Interfaces Configuration Guide*.

Configuring TCC Encapsulation on Interfaces

Also known as Layer 2.5 VPNs, the translation cross-connect (TCC) encapsulation types allow you to configure different encapsulation types at the ingress and egress of a Layer 2 VPN or the ingress and egress of a Layer 2 circuit. For example, a CE router at the ingress of a Layer 2 VPN path can send traffic in a Frame Relay encapsulation. A CE router at the egress of that path can receive the traffic in an ATM encapsulation.

For information on how to configure encapsulations for Layer 2 circuits, see "Configuring the Interface Encapsulation Type for Layer 2 Circuits" on page 540.

The configuration for TCC encapsulation types is similar to the configuration for CCC encapsulation types. For Layer 2 VPNs, you specify a TCC encapsulation type for each PE-router-to-CE-router interface. The encapsulation type configured for the interface should match the encapsulation type configured under the routing instance. For information about how to configure the encapsulation type under the routing instance, see "Configuring the Encapsulation Type" on page 79.

You need to configure the TCC encapsulation on both the physical and logical interfaces. To configure the TCC encapsulation type, include the encapsulation-type statement:

```
encapsulation-type tcc-encapsulation-type;
```

To configure the TCC encapsulation type on the physical interface, include the encapsulation-type statement at the following hierarchy levels:

- [edit interfaces *interface-name*]
- [edit logical-systems *logical-system-name* interfaces *interface-name*]

To configure the TCC encapsulation type on the logical interface, include the encapsulation-type statement at the following hierarchy levels:

- [edit interfaces *interface-name* unit *logical-unit-number*]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]

You configure the encapsulation type at the [edit interfaces] hierarchy level differently than at the [edit routing-instances] hierarchy level. For example, you specify the encapsulation as frame-relay at the [edit routing-instances] hierarchy level and as frame-relay-tcc at the [edit interfaces] hierarchy level.

For Layer 2.5 VPNs employing an Ethernet interface as the TCC router, you can configure an Ethernet TCC or an extended VLAN TCC.

To configure an Ethernet TCC or an extended VLAN TCC, include the proxy and remote statements:

```
proxy inet-address;
remote (inet-address | mac-address);
```

You can include these statements at the following hierarchy levels:

- [edit interfaces *interface-name* unit *logical-unit-number* family tcc]
- [edit logical-interfaces *logical-interface-name* interfaces *interface-name* unit *logical-unit-number* family tcc]

The proxy inet-address address statement defines the IP address for which the TCC router is acting as proxy.

The remote (inet-address | mac-address) statement defines the location of the remote router.

Ethernet TCC is supported on interfaces that carry IP version 4 (IPv4) traffic only. Ethernet TCC encapsulation is supported on 1-port Gigabit Ethernet, 2-port Gigabit Ethernet, 4-port Gigabit Ethernet, and 4-port Fast Ethernet Physical Interface Cards (PICs) only.

For more information on how to configure interfaces and interface encapsulations, see the *JUNOS Network Interfaces Configuration Guide*.

Configuring Layer 2 VPN Policing on Interfaces

You can use policing to control the amount of traffic flowing over the interfaces servicing a Layer 2 VPN. If policing is disabled on an interface, all the available bandwidth on a Layer 2 VPN tunnel can be used by a single CCC or TCC interface.

For more information about the policer statement, see the *JUNOS Policy Framework Configuration Guide*.

To enable Layer 2 VPN policing on an interface, include the policer statement:

```
policer {
    input policer-template-name;
```

```
    output policer-template-name;
}
```

If you configure CCC encapsulation, you can include the policer statement at the following hierarchy levels:

- [edit interfaces *interface-name* unit *logical-unit-number* family ccc]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family ccc]

If you configure TCC encapsulation, you can include the policer statement at the following hierarchy levels:

- [edit interfaces *interface-name* unit *logical-unit-number* family tcc]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family tcc]

For information about how to configure the encapsulation type, see "Configuring the Encapsulation Type" on page 79.

Disabling the Control Word for Layer 2 VPNs

A 4-byte control word provides support for the emulated VC encapsulation for Layer 2 VPNs. This control word is added between the Layer 2 protocol data unit (PDU) being transported and the VC label that is used for demultiplexing. Various networking formats (ATM, Frame Relay, Ethernet, and so on) use the control word in a variety of ways.

On networks with equipment that does not support the control word, you can disable it by including the no-control-word statement:

```
no-control-word;
```

For a list of hierarchy levels at which you can configure this statement, see the statement summary section for this statement.

For more information on configuring the control word, see "Configuring the Control Word for Layer 2 Circuits" on page 535 and the *JUNOS Feature Guide*.

NOTE: Use the no-control word statement to disable the control word when the topology uses generic routing encapsulation (GRE) as the connection mechanism between PEs, and one of the PEs is an M-series router.

Chapter 7

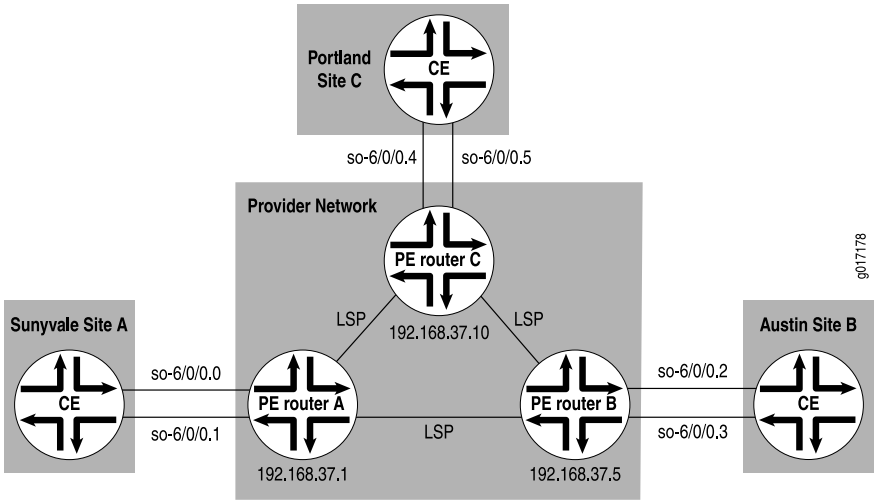
Layer 2 VPN Configuration Example

This chapter provides an example of a Layer 2 virtual private network (VPN) spanning three sites. The following sections explain how to configure Layer 2 VPN functionality on the provider edge (PE) routers connected to each site:

Simple Full-Mesh Layer 2 VPN Overview

In the sections that follow, you configure a simple full-mesh Layer 2 VPN spanning three sites: Sunnyvale, Austin, and Portland. Each site connects to a PE router. The customer edge (CE) routers at each site use Frame Relay to carry Layer 2 traffic to the PE routers. Since this example uses a full-mesh topology between all three sites, each site requires two logical interfaces (one for each of the other CE routers), although only one physical link is needed to connect each PE router to each CE router. Figure 7 on page 88 illustrates the topology of this Layer 2 VPN.

Figure 7: Example of a Simple Full-Mesh Layer 2 VPN Topology



Enabling an IGP on the PE Routers

To allow the PE routers to exchange routing information among themselves, you must configure an interior gateway protocol (IGP) or static routes on these routers. You configure the IGP on the master instance of the routing protocol process (rpd) (that is, at the [edit protocols] hierarchy level), not within the Layer 2 VPN routing instance (that is, not at the [edit routing-instances] hierarchy level). Turn on traffic engineering on the IGP.

You configure the IGP in the standard way. This example does not include this portion of the configuration.

Configuring MPLS LSP Tunnels Between the PE Routers

In this configuration example, Resource Reservation Protocol (RSVP) is used for Multiprotocol Label Switching (MPLS) signaling. Therefore, in addition to configuring RSVP, you must create an MPLS label-switched path (LSP) to tunnel the VPN traffic.

On Router A, enable RSVP and configure one end of the MPLS LSP tunnel to Router B. When configuring the MPLS LSP, include all interfaces using the interface all statement.

```
[edit]
protocols {
    rsvp {
        interface all;
    }
    mpls {
        label-switched-path RouterA-to-RouterB {
            to 192.168.37.5;
            primary Path-to-RouterB;
        }
        label-switched-path RouterA-to-RouterC {
            to 192.168.37.10;
```

```
            primary Path-to-RouterC;
        }
        interface all;
    }
}
```

On Router B, enable RSVP and configure the other end of the MPLS LSP tunnel. Again, configure the interfaces by using the **interface all** statement.

```
[edit]
protocols {
    rsvp {
        interface all;
    }
    mpls {
        label-switched-path RouterB-to-RouterA {
            to 192.168.37.1;
            primary Path-to-RouterA;
        }
        label-switched-path RouterB-to-RouterC {
            to 192.168.37.10;
            primary Path-to-RouterC;
        }
        interface all;
    }
}
```

On Router C, enable RSVP and configure the other end of the MPLS LSP tunnel. Again, configure all interfaces using the **interface all** statement.

```
[edit]
protocols {
    rsvp {
        interface all;
    }
    mpls {
        label-switched-path RouterC-to-RouterA {
            to 192.168.37.1;
            primary Path-to-RouterA;
        }
        label-switched-path RouterC-to-RouterB {
            to 192.168.37.5;
            primary Path-to-RouterB;
        }
        interface all;
    }
}
```

Configuring IBGP on the PE Routers

On the PE routers, configure an internal BGP (IBGP) session with the following parameters:

- Layer 2 VPN—To indicate that the IBGP session is for a Layer 2 VPN, include the **family l2vpn** statement.

- Local address—The IP address in the **local-address** statement is the same as the address configured in the **to** statement at the [**edit protocols mpls label-switched-path** *lsp-path-name*] hierarchy level on the remote PE router. The IBGP session for Layer 2 VPNs runs through this address.
- Neighbor address—Include the **neighbor** statement, specifying the IP address of the neighboring PE router.

On Router A, configure IBGP:

```
[edit]
protocols {
    bgp {
        import match-all;
        export match-all;
        group pe-pe {
            type internal;
            neighbor 192.168.37.5 {
                local-address 192.168.37.1;
                family l2vpn {
                    signaling;
                }
            }
            neighbor 192.168.37.10 {
                local-address 192.168.37.1;
                family l2vpn {
                    signaling;
                }
            }
        }
    }
}
```

On Router B, configure IBGP:

```
[edit]
protocols {
    bgp {
        local-address 192.168.37.5;
        import match-all;
        export match-all;
        group pe-pe {
            type internal;
            neighbor 192.168.37.1 {
                local-address 192.168.37.5;
                family l2vpn {
                    signaling;
                }
            }
            neighbor 192.168.37.10 {
                local-address 192.168.37.5;
                family l2vpn {
                    signaling;
                }
            }
```

```
        }
    }
}
```

On Router C, configure IBGP:

```
[edit]
protocols {
    bgp {
        local-address 192.168.37.10;
        import match-all;
        export match-all;
        group pe-pe {
            type internal;
            neighbor 192.168.37.1 {
                local-address 192.168.37.10;
                family l2vpn {
                    signaling;
                }
            }
            neighbor 192.168.37.5 {
                local-address 192.168.37.10;
                family l2vpn {
                    signaling;
                }
            }
        }
    }
}
```

Configuring Routing Instances for Layer 2 VPNs on the PE Routers

The three PE routers service the Layer 2 VPN, so you need to configure a routing instance on each router. For the VPN, you must define the following in each routing instance:

- Route distinguisher, which must be unique for each routing instance on the PE router. It is used to distinguish the addresses in one VPN from those in another VPN.
- Instance type of l2vpn, which configures the router to run a Layer 2 VPN.
- Interfaces connected to the CE routers.
- VPN routing and forwarding (VRF) import and export policies, which must be the same on each PE router that services the same VPN and are used to control the network topology. Unless the import policy contains only a then reject statement, it must include a reference to a community. Otherwise, when you attempt to commit the configuration, the commit operation fails.

On Router A, configure the following routing instance for the Layer 2 VPN:

```
[edit]
routing-instances {
    VPN-Sunnyvale-Portland-Austin {
        instance-type l2vpn;
```

```
        interface so-6/0/0.0;
        interface so-6/0/0.1;
        route-distinguisher 100:1;
        vrf-import vpn-SPA-import;
        vrf-export vpn-SPA-export;
        protocols {
            l2vpn {
                encapsulation-type frame-relay;
                site Sunnyvale {
                    site-identifier 1;
                    interface so-6/0/0.0 {
                        remote-site-id 2;
                    }
                    interface so-6/0/0.1 {
                        remote-site-id 3;
                    }
                }
            }
        }
    }
}
```

On Router B, configure the following routing instance for the Layer 2 VPN:

```
[edit]
routing-instances {
    VPN-Sunnyvale-Portland-Austin {
        instance-type l2vpn;
        interface so-6/0/0.2;
        interface so-6/0/0.3;
        route-distinguisher 100:1;
        vrf-import vpn-SPA-import;
        vrf-export vpn-SPA-export;
        protocols {
            l2vpn {
                encapsulation-type frame-relay;
                site Austin {
                    site-identifier 2;
                    interface so-6/0/0.2 {
                        remote-site-id 1;
                    }
                    interface so-6/0/0.3 {
                        remote-site-id 3;
                    }
                }
            }
        }
    }
}
```

On Router C, configure the following routing instance for the Layer 2 VPN:

```
[edit]
routing-instances {
    VPN-Sunnyvale-Portland-Austin {
        instance-type l2vpn;
```

```
        interface so-6/0/0.4;
        interface so-6/0/0.5;
        route-distinguisher 100:1;
        vrf-import vpn-SPA-import;
        vrf-export vpn-SPA-export;
        protocols {
            l2vpn {
                encapsulation-type frame-relay;
                site Portland {
                    site-identifier 3;
                    interface so-6/0/0.4 {
                        remote-site-id 1;
                    }
                    interface so-6/0/0.5 {
                        remote-site-id 2;
                    }
                }
            }
        }
    }
}
```

Configuring CCC Encapsulation on the Interfaces

You need to specify a circuit cross-connect (CCC) encapsulation type for each PE-router-to-CE-router interface running in the Layer 2 VPN. This encapsulation type should match the encapsulation type configured under the routing instance.

Configure the following CCC encapsulation types for the interfaces on Router A:

```
[edit]
interfaces {
    interface so-6/0/0 {
        encapsulation frame-relay-ccc;
        unit 0 {
            encapsulation frame-relay-ccc;
        }
    }
    interface so-6/0/0 {
        encapsulation frame-relay-ccc;
        unit 1 {
            encapsulation frame-relay-ccc;
        }
    }
}
```

Configure the following CCC encapsulation types for the interfaces on Router B:

```
[edit]
interfaces {
    interface so-6/0/0 {
        encapsulation frame-relay-ccc;
        unit 2 {
            encapsulation frame-relay-ccc;
        }
```

```
    }
    interface so-6/0/0 {
        encapsulation frame-relay-ccc;
        unit 3 {
            encapsulation frame-relay-ccc;
        }
    }
}
```

Configure the following CCC encapsulation types for the interfaces on Router C:

```
[edit]
interfaces {
    interface so-6/0/0 {
        encapsulation frame-relay-ccc;
        unit 4 {
            encapsulation frame-relay-ccc;
        }
    }
    interface so-6/0/0 {
        encapsulation frame-relay-ccc;
        unit 5 {
            encapsulation frame-relay-ccc;
        }
    }
}
```

Configuring VPN Policy on the PE Routers

You must configure VPN import and export policies on each of the PE routers so that they install the appropriate routes in their VRF tables, which the routers use to forward packets within the VPN.

NOTE: Use the community add statement at the [edit policy-options policy statement term] hierarchy level to facilitate Layer 2 VPN VRF export policies.

On Router A, configure the following VPN import and export policies:

```
[edit]
policy-options {
    policy-statement match-all {
        term acceptable {
            then accept;
        }
    }
    policy-statement vpn-SPA-export {
        term a {
            then {
                community add SPA-com;
                accept;
            }
        }
        term b {
```

```
            then reject;
        }
    }
    policy-statement vpn-SPA-import {
        term a {
            from {
                protocol bgp;
                community SPA-com;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    community SPA-com members target:69:100;
}
```

On Router B, configure the following VPN import and export policies:

```
[edit]
policy-options {
    policy-statement match-all {
        term acceptable {
            then accept;
        }
    }
    policy-statement vpn-SPA-import {
        term a {
            from {
                protocol bgp;
                community SPA-com;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    policy-statement vpn-SPA-export {
        term a {
            then {
                community add SPA-com;
                accept;
            }
        }
        term b {
            then reject;
        }
    }
    community SPA-com members target:69:100;
}
```

On Router C, configure the following VPN import and export policies:

```
[edit]
```

```
policy-options {
   policy-statement match-all {
      term acceptable {
         then accept;
      }
   }
   policy-statement vpn-SPA-import {
      term a {
         from {
            protocol bgp;
            community SPA-com;
         }
         then accept;
      }
      term b {
         then reject;
      }
   }
   policy-statement vpn-SPA-export {
      term a {
         then {
            community add SPA-com;
            accept;
         }
      }
      term b {
         then reject;
      }
   }
   community SPA-com members target:69:100;
}
```

To apply the VPN policies on the routers, include the vrf-export and vrf-import statements when you configure the routing instance. The VRF import and export policies handle the route distribution across the IBGP session running between the PE routers.

To apply the VPN policies on Router A, include the following statements:

```
[edit]
routing-instances {
   VPN-Sunnyvale-Portland-Austin {
      vrf-import vpn-SPA-import;
      vrf-export vpn-SPA-export;
   }
}
```

To apply the VPN policies on Router B, include the following statements:

```
[edit]
routing-instances {
   VPN-Sunnyvale-Portland-Austin {
      vrf-import vpn-SPA-import;
      vrf-export vpn-SPA-export;
   }
}
```

To apply the VPN policies on Router C, include the following statements:

```
[edit]
routing-instances {
    VPN-Sunnyvale-Portland-Austin {
        vrf-import vpn-SPA-import;
        vrf-export vpn-SPA-export;
    }
}
```

Layer 2 VPN Configuration Summarized by Router

For a summary of the configuration on each router in the examples in this chapter, see the following sections:

- Summary for Router A (PE Router for Sunnyvale) on page 97
- Summary for Router B (PE Router for Austin) on page 99
- Summary for Router C (PE Router for Portland) on page 101

Summary for Router A (PE Router for Sunnyvale)

Routing Instance for Layer 2 VPN

```
[edit]
routing-instances {
    VPN-Sunnyvale-Portland-Austin {
        instance-type l2vpn;
        interface so-6/0/0.0;
        interface so-6/0/0.1;
        route-distinguisher 100:1;
        vrf-import vpn-SPA-import;
        vrf-export vpn-SPA-export;
        protocols {
            l2vpn {
                encapsulation-type frame-relay;
                site Sunnyvale {
                    site-identifier 1;
                    interface so-6/0/0.0 {
                        remote-site-id 2;
                    }
                    interface so-6/0/0.1 {
                        remote-site-id 3;
                    }
                }
            }
        }
    }
}
```

Configure CCC Encapsulation Types for Interfaces

```
interfaces {
    interface so-6/0/0 {
        encapsulation frame-relay-ccc;
        unit 0 {
            encapsulation frame-relay-ccc;
        }
    }
```

```
    interface so-6/0/0 {
      encapsulation frame-relay-ccc;
      unit 1 {
        encapsulation frame-relay-ccc;
      }
    }
}
```

Master Protocol Instance

```
protocols {
}
```

Enable RSVP

```
rsvp {
    interface all;
}
```

Configure MPLS LSPs

```
mpls {
    label-switched-path RouterA-to-RouterB {
        to 192.168.37.5;
        primary Path-to-RouterB {
            cspf;
        }
    }
    label-switched-path RouterA-to-RouterC {
        to 192.168.37.10;
        primary Path-to-RouterC {
            cspf;
        }
    }
    interface all;
}
```

Configure IBGP

```
bgp {
    import match-all;
    export match-all;
    group pe-pe {
        type internal;
        neighbor 192.168.37.5 {
            local-address 192.168.37.1;
            family l2vpn {
                signaling;
            }
        }
        neighbor 192.168.37.10 {
            local-address 192.168.37.1;
            family l2vpn {
                signaling;
            }
        }
    }
}
```

Configure VPN Policy

```
policy-options {
    policy-statement match-all {
        term acceptable {
```

```
            then accept;
        }
    }
    policy-statement vpn-SPA-export {
        term a {
            then {
                community add SPA-com;
                accept;
            }
        }
        term b {
            then reject;
        }
    }
    policy-statement vpn-SPA-import {
        term a {
            from {
                protocol bgp;
                community SPA-com;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    community SPA-com members target:69:100;
}
```

Summary for Router B (PE Router for Austin)

Routing Instance for VPN

```
[edit]
routing-instances {
    VPN-Sunnyvale-Portland-Austin {
        instance-type l2vpn;
        interface so-6/0/0.2;
        interface so-6/0/0.3;
        route-distinguisher 100:1;
        vrf-import vpn-SPA-import;
        vrf-export vpn-SPA-export;
    }
}
```

Configure Layer 2 VPN

```
protocols {
    l2vpn {
        encapsulation-type frame-relay;
        site Austin {
            site-identifier 2;
            interface so-6/0/0.2 {
                remote-site-id 1;
            }
            interface so-6/0/0.3 {
                remote-site-id 3;
            }
```

```
    }
  }
}
```

Configure CCC Encapsulation Types for Interfaces

```
[edit]
interfaces {
  interface so-6/0/0 {
    encapsulation frame-relay-ccc;
    unit 2 {
      encapsulation frame-relay-ccc;
    }
  }
  interface so-6/0/0 {
    encapsulation frame-relay-ccc;
    unit 3 {
      encapsulation frame-relay-ccc;
    }
  }
}
```

Master Protocol Instance

```
protocols {
}
```

Enable RSVP

```
rsvp {
  interface all;
}
```

Configure MPLS LSPs

```
mpls {
  label-switched-path RouterB-to-RouterA {
    to 192.168.37.1;
    primary Path-to-RouterA {
      cspf;
    }
  }
  label-switched-path RouterB-to-RouterC {
    to 192.168.37.10;
    primary Path-to-RouterC {
      cspf;
    }
  }
  interface all;
}
```

Configure IBGP

```
bgp {
  local-address 192.168.37.5;
  import match-all;
  export match-all;
  group pe-pe {
    type internal;
    neighbor 192.168.37.1 {
      local-address 192.168.37.5;
      family l2vpn {
        signaling;
      }
```

```
        }
        neighbor 192.168.37.10 {
            local-address 192.168.37.5;
            family l2vpn {
                signaling;
            }
        }
    }
}
```

Configure VPN Policy

```
policy-options {
    policy-statement match-all {
        term acceptable {
            then accept;
        }
    }
    policy-statement vpn-SPA-import {
        term a {
            from {
                protocol bgp;
                community SPA-com;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    policy-statement vpn-SPA-export {
        term a {
            then {
                community add SPA-com;
                accept;
            }
        }
        term b {
            then reject;
        }
    }
    community SPA-com members target:69:100;
}
```

Summary for Router C (PE Router for Portland)

Routing Instance for VPN

```
[edit]
routing-instances {
    VPN-Sunnyvale-Portland-Austin {
        instance-type l2vpn;
        interface so-6/0/0.3;
        interface so-6/0/0.4;
        route-distinguisher 100:1;
        vrf-import vpn-SPA-import;
        vrf-export vpn-SPA-export;
    }
```

```
}
```

Configure Layer 2 VPN

```
protocols {
    l2vpn {
        encapsulation-type frame-relay;
        site Portland {
            site-identifier 3;
            interface so-6/0/0.4 {
                remote-site-id 1;
            }
            interface so-6/0/0.5 {
                remote-site-id 2;
            }
        }
    }
}
```

Configure CCC Encapsulation Types for Interfaces

```
[edit]
interfaces {
    interface so-6/0/0 {
        encapsulation frame-relay-ccc;
        unit 4 {
            encapsulation frame-relay-ccc;
        }
    }
    interface so-6/0/0 {
        encapsulation frame-relay-ccc;
        unit 5 {
            encapsulation frame-relay-ccc;
        }
    }
}
```

Master Protocol Instance

```
protocols {
}
```

Enable RSVP

```
rsvp {
    interface all;
}
```

Configure MPLS LSPs

```
mpls {
    label-switched-path RouterC-to-RouterA {
        to 192.168.37.1;
        primary Path-to-RouterA {
            cspf;
        }
    }
    label-switched-path RouterC-to-RouterB {
        to 192.168.37.5;
        primary Path-to-RouterB {
            cspf;
        }
    }
    interface all;
```

```
}
```

Configure IBGP

```
bgp {
    local-address 192.168.37.10;
    import match-all;
    export match-all;
    group pe-pe {
        type internal;
        neighbor 192.168.37.1 {
            local-address 192.168.37.10;
            family l2vpn {
                signaling;
            }
        }
        neighbor 192.168.37.5 {
            local-address 192.168.37.10;
            family l2vpn {
                signaling;
            }
        }
    }
}
```

Configure VPN Policy

```
policy-options {
    policy-statement match-all {
        term acceptable {
            then accept;
        }
    }
    policy-statement vpn-SPA-import {
        term a {
            from {
                protocol bgp;
                community SPA-com;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    policy-statement vpn-SPA-export {
        term a {
            then {
                community add SPA-com;
                accept;
            }
        }
        term b {
            then reject;
        }
    }
    community SPA-com members target:69:100;
}
```

Chapter 8

Summary of Layer 2 VPN Configuration Statements

The following sections explain the major routing-instances configuration statements that apply specifically to Layer 2 virtual private networks (VPNs). The statements are organized alphabetically. Routing instances and the statements at the [edit routing-instances *routing-instance-name* protocols] hierarchy level are explained in the *JUNOS Routing Protocols Configuration Guide*.

control-word

Syntax (control-word | no-control-word);

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn],
[edit routing-instances *routing-instance-name* protocols l2vpn]

Release Information Statement introduced before JUNOS Release 7.4.

Description Specify the control word. The control word is 4 bytes long and is inserted between the Layer 2 protocol data unit (PDU) being transported and the virtual connection (VC) label that is used for demultiplexing.

- control-word—Enables the use of the control word.
- no-control-word—Disables the use of the control word.

Default The control word is enabled by default. You can also configure the control word explicitly using the control-word statement.

Usage Guidelines See "Disabling the Control Word for Layer 2 VPNs" on page 85.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

description

Syntax description *text*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn site *site-name* interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols l2vpn site *site-name* interface *interface-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Describe the VPN or virtual private LAN service (VPLS) routing instance.

Options *text*—Provide a text description. If the text includes one or more spaces, enclose it in quotation marks (" "). Any descriptive text you include is displayed in the output of the show route instance detail command and has no effect on operation.

Usage Guidelines See "Configuring the Description" on page 19.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

encapsulation

See the following sections:

- encapsulation (Logical Interface) on page 108
- encapsulation (Physical Interface) on page 110

encapsulation (Logical Interface)

Syntax encapsulation (atm-ccc-cell-relay | atm-ccc-vc-mux | atm-cisco-nlpid | atm-mlppp-llc | atm-nlpid | atm-ppp-llc | atm-ppp-vc-mux | atm-snap | atm-tcc-snap | atm-tcc-vc-mux | atm-vc-mux | ether-over-atm-llc | ether-vpls-over-atm-llc | ethernet | frame-relay-ccc | frame-relay-ppp | frame-relay-tcc | multilink-frame-relay-end-to-end | multilink-ppp | ppp-over-ether | ppp-over-ether-over-atm-llc | vlan-ccc | vlan-tcc | vlan-vpls);

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Logical link-layer encapsulation type.

Options atm-ccc-cell-relay—Use Asynchronous Transfer Mode (ATM) cell relay encapsulation.

atm-ccc-vc-mux—Use ATM VC multiplex encapsulation on circuit cross-connect (CCC) circuits. When you use this encapsulation type, you can configure the family ccc only.

atm-cisco-nlpid—Use Cisco ATM Network Layer Protocol identifier (NLPID) encapsulation. When you use this encapsulation type, you can configure the family inet only.

atm-mlppp-llc—For ATM2 intelligent queuing (IQ) interfaces only, use Multilink Point-to-Point (MLPPP) over ATM adaptation layer 5 (AAL5) logical link control (LLC). For this encapsulation type, your routing platform must be equipped with a Link Services or Voice Services Physical Interface Card (PIC).

atm-nlpid—Use ATM NLPID encapsulation. When you use this encapsulation type, you can configure the family inet only.

atm-ppp-llc—For ATM2 IQ interfaces only, use Point-to-Point Protocol (PPP) over AAL5 logical link control (LLC) encapsulation.

atm-ppp-vc-mux—For ATM2 IQ interfaces only, use PPP over AAL5 multiplex encapsulation.

atm-snap—Use ATM Subnetwork Access Protocol (SNAP) encapsulation.

atm-tcc-snap—Use ATM SNAP encapsulation on translational cross-connect (TCC) circuits.

atm-tcc-vc-mux—Use ATM VC multiplex encapsulation on TCC circuits. When you use this encapsulation type, you can configure the family tcc only.

atm-vc-mux—Use ATM VC multiplex encapsulation. When you use this encapsulation type, you can configure the family inet only.

ether-over-atm-llc—For interfaces that carry IP version 4 (IPv4) traffic, use Ethernet over ATM LLC encapsulation. When you use this encapsulation type, you cannot configure multipoint interfaces.

ether-vpls-over-atm-llc—For ATM2 IQ interfaces only, use the Ethernet VPLS over ATM LLC encapsulation to bridge Ethernet interfaces and ATM interfaces over a VPLS routing instance (as described in RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*). Packets from the ATM interfaces are converted to standard ENET2/802.3 encapsulated Ethernet frames with the frame check sequence (FCS) field removed.

ethernet—Use Ethernet II encapsulation (as described in RFC 894, *A Standard For The Transmission Of IP Datagrams Over Ethernet Networks*).

frame-relay-ccc—Use Frame Relay encapsulation on CCC circuits. When you use this encapsulation type, you can configure the family ccc only.

frame-relay-ppp—Use Frame Relay encapsulation on PPP circuits.

frame-relay-tcc—Use Frame Relay encapsulation on TCC circuits for connecting unlike media. When you use this encapsulation type, you can configure the family tcc only.

multilink-frame-relay-end-to-end—Use Multilink Frame Relay (MLFR) FRF.15 encapsulation. This encapsulation is used only on multilink, link services, and voice services interfaces and their constituent T1 or E1 interfaces.

multilink-ppp—Use MLPPP encapsulation. This encapsulation is used only on multilink, link services, and voice services interfaces and their constituent T1 or E1 interfaces.

ppp-over-ether—For underlying Ethernet interfaces on J-series Services Routers only, use PPP over Ethernet encapsulation. When you use this encapsulation type, you cannot configure the interface address. Instead you configure the interface address on the PPP interface. For more information, see the *J-series Services Router Advanced WAN Access Configuration Guide*.

ppp-over-ether-over-atm-llc—For underlying ATM interfaces on J-series Services Routers only, use PPP over Ethernet over ATM LLC encapsulation. When you use this encapsulation type, you cannot configure the interface address. Instead you configure the interface address on the PPP interface. For more information, see the *J-series Services Router Advanced WAN Access Configuration Guide*.

vlan-ccc—Use Ethernet virtual LAN (VLAN) encapsulation on CCC circuits. When you use this encapsulation type, you can configure the family ccc only.

vlan-tcc—Use Ethernet VLAN encapsulation on TCC circuits. When you use this encapsulation type, you can configure the family tcc only.

vlan-vpls—Use Ethernet VLAN encapsulation on virtual private LAN service (VPLS) circuits.

Usage Guidelines See “Configuring CCC Encapsulation on Interfaces” on page 82 or “Configuring TCC Encapsulation on Interfaces” on page 83.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

encapsulation (Physical Interface)

Syntax encapsulation (atm-ccc-cell-relay | atm-pvc | cisco-hdlc | cisco-hdlc-ccc | cisco-hdlc-tcc | ethernet-ccc | ethernet-over-atm | ethernet-tcc | ethernet-vpls | extended-frame-relay-ccc | extended-frame-relay-tcc | extended-vlan-ccc | extended-vlan-tcc | extended-vlan-vpls | flexible-ethernet-services | flexible-frame-relay | frame-relay | frame-relay-ccc | frame-relay-port-ccc | frame-relay-tcc | multilink-frame-relay-uni-nni | ppp | ppp-ccc | ppp-tcc | vlan-ccc | vlan-vpls);

Hierarchy Level [edit interfaces *interface-name*],
[edit logical-systems *logical-system-name* interfaces *interface-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Physical link-layer encapsulation type.

Default PPP encapsulation.

Options atm-ccc-cell-relay—Use ATM cell-relay encapsulation.

atm-pvc—Use ATM permanent virtual connection (PVC) encapsulation.

cisco-hdlc—Use Cisco-compatible HDLC framing.

cisco-hdlc-ccc—Use Cisco-compatible HDLC framing on CCC circuits.

cisco-hdlc-tcc—Use Cisco-compatible HDLC framing on TCC circuits for connecting unlike media.

ethernet-ccc—Use Ethernet CCC encapsulation on Ethernet interfaces that must accept packets carrying standard Tag Protocol ID (TPID) values. For example, Ethernet CCC encapsulation can be used to transparently transport any VLANs or other Ethernet frames entering a port across a Layer 2 circuit.

ethernet-over-atm—For interfaces that carry IPv4 traffic, use Ethernet over ATM encapsulation. When you use this encapsulation type, you cannot configure multipoint interfaces. As defined in RFC 1483 *Multiprotocol Encapsulation over ATM Adaptation Layer 5*, this encapsulation type allows ATM interfaces to connect to devices that support only bridged-mode protocol data units (BPDUs). The JUNOS software does not completely support bridging, but accepts BPDU packets as a default gateway. If you use the router as an edge device, then the router acts as a default gateway. It accepts Ethernet LLC/SNAP frames with IP or Address Resolution Protocol (ARP) in the payload and drops the rest. For packets destined for the Ethernet LAN, a route lookup is done using the destination IP address. If the route lookup yields a full address match, the packet is encapsulated with an LLC/SNAP and media access control (MAC) header and forwarded to the ATM interface.

ethernet-tcc—For interfaces that carry IPv4 traffic, use Ethernet TCC encapsulation on interfaces that must accept packets carrying standard TPID values. Ethernet TCC is not currently supported on Fast Ethernet 48-port PICs.

ethernet-vpls—Use Ethernet VPLS encapsulation on Ethernet interfaces that have VPLS enabled and that must accept packets carrying standard TPID values.

extended-frame-relay-ccc—Use Frame Relay encapsulation on CCC circuits. This encapsulation type allows you to dedicate data link connection identifiers (DLCIs) 1 through 1022 to CCC.

extended-frame-relay-tcc—Use Frame Relay encapsulation on TCC circuits to connect unlike media. This encapsulation type allows you to dedicate DLCIs 1 through 1022 to TCC.

extended-vlan-ccc—Use extended VLAN encapsulation on CCC circuits with Gigabit Ethernet and 4-port Fast Ethernet interfaces that must accept packets carrying 802.1Q values.

extended-vlan-tcc—For interfaces that carry IPv4 traffic, use extended VLAN encapsulation on TCC circuits with Gigabit Ethernet interfaces on which you want to use 802.1Q tagging. Extended Ethernet TCC is not currently supported on Fast Ethernet 48-port PICs.

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.

flexible-ethernet-services—For Gigabit Ethernet IQ interfaces and Gigabit Ethernet PICs with small form-factor pluggable transceivers (SFPs) only, 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. If you configure flexible Ethernet services encapsulation on the physical interface, VLAN IDs from 1 through 511 are no longer reserved for normal VLANs.

flexible-frame-relay—For IQ interfaces only, use flexible Frame Relay encapsulation when you want to configure multiple per-unit Frame Relay encapsulations. This encapsulation type allows you to configure any combination of TCC, CCC, and standard Frame Relay encapsulations on a single physical port. Also, each logical interface can have any DLCI value from 1 through 1022.

frame-relay—Use Frame Relay encapsulation.

frame-relay-ccc—Use Frame Relay encapsulation or Frame Relay encapsulation on CCC circuits.

frame-relay-port-ccc—Use Frame Relay port CCC encapsulation to transparently carry all the DLCIs between two CE routers without explicitly configuring each DLCI on the two provider edge (PE) routers with Frame Relay transport. When you use this encapsulation type, you can configure the family ccc only.

frame-relay-tcc—Use Frame Relay encapsulation on TCC circuits to connect unlike media.

multilink-frame-relay-uni-nni—Use MLFR user-to-network interface (UNI) network-to-network interface (NNI) encapsulation. This encapsulation is used only on link services and voice services interfaces functioning as FRF.16 bundles and their constituent T1 or E1 interfaces.

ppp—Use serial PPP encapsulation.

ppp-ccc—Use serial PPP encapsulation on CCC circuits. When you use this encapsulation type, you can configure the family ccc only.

ppp-tcc—Use serial PPP encapsulation on TCC circuits for connecting unlike media. When you use this encapsulation type, you can configure the family tcc only.

vlan-ccc—Use Ethernet VLAN encapsulation on CCC circuits.

vlan-vpls—Use VLAN VPLS encapsulation on Ethernet interfaces with VLAN tagging and VPLS enabled. Interfaces with VLAN VPLS encapsulation accept packets carrying standard TPID values only.

Usage Guidelines See "Configuring CCC Encapsulation on Interfaces" on page 82 or "Configuring TCC Encapsulation on Interfaces" on page 83.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

encapsulation-type

Syntax encapsulation-type *type*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn],
[edit routing-instances *routing-instance-name* protocols l2vpn]

Release Information Statement introduced before JUNOS Release 7.4.

Description Layer 2 protocol used for traffic from the customer edge (CE) router.

Options *type*—The following Layer 2 encapsulation types are supported:

- atm-aal5—ATM Adaptation Layer (AAL/5)
- atm-cell—ATM cell relay
- atm-cell-port-mode—ATM cell relay port promiscuous mode
- atm-cell-vc-mode—ATM VC cell relay nonpromiscuous mode
- atm-cell-vp-mode—ATM virtual path (VP) cell relay promiscuous mode
- cisco-hdlc—Cisco Systems-compatible HDLC
- ethernet—Ethernet
- ethernet-vlan—Ethernet VLAN
- frame-relay—Frame Relay
- frame-relay-port-mode—Frame Relay port mode
- interworking—Layer 2.5 interworking VPN
- ppp—PPP

Usage Guidelines See “Configuring the Encapsulation Type” on page 79.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

interface

Syntax

```
interface interface-name {
    description text;
    remote-site-id remote-site-id;
}
```

Hierarchy Level

```
[edit logical-systems logical-system-name routing-instances routing-instance-name
    protocols],
[edit routing-instances routing-instance-name protocols]
```

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure an interface to handle traffic for a circuit configured for the Layer 2 VPN.

Options *interface-name*—Name of the interface used for the Layer 2 VPN.

The remaining statements are explained separately.

Usage Guidelines See “Configuring the Site” on page 77 and “Configuring the Remote Site ID” on page 78.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

l2vpn

Syntax

```
l2vpn {
    (control-word | no-control-word);
    encapsulation-type type;
    traceoptions {
        file filename <files number> <size size> <world-readable | no-world-readable>;
        flag flag <flag-modifier> <disable>;
    }
    site site-name {
        site-identifier identifier;
        site-preference preference-value {
            backup;
            primary;
        }
        interface interface-name {
            description text;
            remote-site-id remote-site-id;
        }
    }
}
```

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols],
[edit routing-instances *routing-instance-name* protocols]

Release Information Statement introduced before JUNOS Release 7.4.

Description Enable a Layer 2 VPN routing instance on a PE router.

The remaining statements are explained separately.

Usage Guidelines See "Configuring a Layer 2 VPN Routing Instance" on page 77.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

no-control-word

See control-word

policer

Syntax

```
policer {
    input policer-template-name;
    output policer-template-name;
}
```

Hierarchy Level

[edit interfaces *interface-name* unit *logical-unit-number* family (ccc | inet | tcc)],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family (ccc | inet | tcc)]

Release Information

Statement introduced before JUNOS Release 7.4.

Description

Use policing to control the amount of traffic flowing over the interfaces servicing a Layer 2 VPN.

Options

input *policer-template-name*—Name of one policer to evaluate when packets are received on the interface.

output *policer-template-name*—Name of one policer to evaluate when packets are transmitted on the interface.

Usage Guidelines

See "Configuring Layer 2 VPN Policing on Interfaces" on page 84.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Topics

JUNOS Policy Framework Configuration Guide and *JUNOS Network Interfaces Configuration Guide*.

proxy

Syntax proxy inet-address *address*;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* family tcc],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family tcc]

Release Information Statement introduced before JUNOS Release 7.4.

Description For Layer 2.5 VPNs using an Ethernet interface as the TCC router, configure the IP address for which the TCC router is proxying. Ethernet TCC is supported on interfaces that carry IPv4 traffic only. Ethernet TCC encapsulation is supported on 1-port Gigabit Ethernet, 2-port Gigabit Ethernet, 4-port Gigabit Ethernet, and 4-port Fast Ethernet PICs only. Ethernet TCC is not supported on the T640 routing node.

Options inet-address *address*—IP address for which the TCC router is acting as a proxy.

Usage Guidelines See "Configuring TCC Encapsulation on Interfaces" on page 83.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

remote

Syntax remote (inet-address | mac-address) *address*;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* family tcc],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family tcc]

Release Information Statement introduced before JUNOS Release 7.4.

Description For Layer 2.5 VPNs employing an Ethernet interface as the TCC router, configure the location of the remote router. Ethernet TCC is supported on interfaces that carry IPv4 traffic only. Ethernet TCC encapsulation is supported on 1-port Gigabit Ethernet, 2-port Gigabit Ethernet, 4-port Gigabit Ethernet, and 4-port Fast Ethernet PICs only.

Options inet-address*address*—The IP address of the remote site.

mac-address *address*—The MAC address of the remote site.

Usage Guidelines See "Configuring TCC Encapsulation on Interfaces" on page 83.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

remote-site-id

Syntax remote-site-id *remote-site-ID*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn site *site-name* interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols l2vpn site *site-name* interface *interface-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Control the remote interface to which the interface should connect. If you do not explicitly configure the remote site ID, the order of the interfaces configured for the site determines the default value. This statement is optional.

Options *remote-site-ID*—Identifier specifying the interface on the remote PE router the Layer 2 VPN routing instance connects to.

Usage Guidelines See “Configuring the Remote Site ID” on page 78.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

site

Syntax

```
site site-name {
    site-identifier identifier;
    site-preference preference-value {
        backup;
        primary;
    }
    interface interface-name {
        description text;
        remote-site-id remote-site-ID;
    }
}
```

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn],
[edit routing-instances *routing-instance-name* protocols l2vpn]

Release Information Statement introduced before JUNOS Release 7.4.

Description Specify the site name, site identifier, and interfaces connecting to the site. Allows you to configure a remote site ID for remote sites.

Options site-identifier *identifier*—Numerical identifier for the site used as a default reference for the remote site ID.

site-name—Name of the site.

The remaining statements are explained separately.

Usage Guidelines See "Configuring the Site" on page 77.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

site-identifier

Syntax site-identifier *identifier*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn site *site-name*],
[edit routing-instances *routing-instance-name* protocols l2vpn site *site-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Specify the numerical identifier for the Layer 2 VPN site used as a default reference for the remote site ID.

Options *identifier*—The numerical identifier for the site, which can be any number from 1 through 65,534.

Usage Guidelines See "Configuring the Site" on page 77.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

site-preference

Syntax

```
site-preference preference-value {
    backup;
    primary;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn site *site-name*],
[edit routing-instances *routing-instance-name* protocols l2vpn site *site-name*],

Release Information

Statement introduced in JUNOS Release 9.1.

Description

Specify the preference value advertised for a particular Layer 2 VPN site. The site preference value is encoded in the BGP local preference attribute. When a PE router receives multiple advertisements with the same VE identifier, the advertisement with the highest local preference value is preferred. You can use this statement to enable multihoming for Layer 2 VPNs.

Options

preference-value—Specify the preference value advertised for a Layer 2 VPN.
Range: 1 through 65,535

backup—Set the preference value to 1.

primary—Set the preference value to 65,535.

Usage Guidelines

See "Configuring a Site Preference and Layer 2 VPN Multihoming" on page 80.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

traceoptions

Syntax

```
traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn],
[edit routing-instances *routing-instance-name* protocols l2vpn]

Release Information

Statement introduced before JUNOS Release 7.4.

Description

Trace traffic flowing through a Layer 2 VPN.

Options

disable—(Optional) Disable the tracing operation. You can use this option to disable a single operation when you have defined a broad group of tracing operations, such as all.

file *filename*—Name of the file to receive the output of the tracing operation. Enclose the name in quotation marks (" ").

files *number*—(Optional) Maximum number of trace files. When a trace file named *trace-file* reaches its maximum size, it is renamed *trace-file*.0, then *trace-file*.1, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum number of files, you also must specify a maximum file size with the size option.
Range: 2 through 1000 files
Default: 2 files

flag *flag*—Tracing operation to perform. To specify more than one tracing operation, include multiple flag statements.

- all—All Layer 2 VPN tracing options
- connections—Layer 2 connections (events and state changes)
- error—Error conditions
- general—General events
- nlri—Layer 2 advertisements received or sent by means of the BGP
- normal—Normal events
- policy—Policy processing
- route—Routing information
- state—State transitions
- task—Routing protocol task processing

- timer—Routing protocol timer processing
- topology—Layer 2 VPN topology changes caused by reconfiguration or advertisements received from other PE routers using BGP

flag-modifier—(Optional) Modifier for the tracing flag. You can specify the following modifier:

- detail—Provide detailed trace information
- receive—Trace received packets
- send—Trace transmitted packets

no-world-readable—(Optional) Prevents any user from reading the trace file.

size *size*—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named *trace-file* reaches this size, it is renamed *trace-file*.0. When *trace-file* again reaches its maximum size, *trace-file*.0 is renamed *trace-file*.1 and *trace-file* is renamed *trace-file*.0. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum file size, you also must specify a maximum number of trace files with the files option.
Syntax: *x*k to specify kilobytes, *x*m to specify megabytes, or *x*g to specify gigabytes
Range: 10 KB through the maximum file size supported on your system
Default: 1 MB

world-readable—(Optional) Allow any user to read the trace file. (Default is no-world-readable.)

Usage Guidelines See "Tracing Layer 2 VPN Traffic and Operations" on page 81.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Part 3
Layer 3 VPNs

Chapter 9

Layer 3 VPN Overview

The JUNOS software implements Layer 3 BGP/Multiprotocol Label Switching (BGP/MPLS) virtual private networks (VPNs) as defined in RFC 2547, *BGP/MPLS VPNs* and Internet draft draft-rosen-rfc2547bis, *BGP/MPLS VPNs* (also referred to as RFC 2547bis).

This chapter discusses the following topics that provide background information about Layer 3 VPNs:

Layer 3 VPN Introduction

In JUNOS software, Layer 3 VPNs are based on RFC 2547bis. RFC 2547bis defines a mechanism by which service providers can use their IP backbones to provide VPN services to their customers. A Layer 3 VPN is a set of sites that share common routing information and whose connectivity is controlled by a collection of policies. The sites that make up a Layer 3 VPN are connected over a provider's existing public Internet backbone.

RFC 2547bis VPNs 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 same private addresses used by other network users. MPLS/BGP VPNs solve this problem by adding a VPN identifier prefix 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.

Layer 3 VPN Standards

Layer 3 VPNs are defined in the following RFCs and IETF Internet drafts:

- RFC 1918, *Address Allocation for Private Internets*
- RFC 2685, *Virtual Private Networks Identifier*
- RFC 2858, *Multiprotocol Extensions for BGP4*
- RFC 4364, *BGP/MPLS IP Virtual Private Networks (VPNs)*
- RFC 4379, *Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures*
- RFC 4659, *BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN*
- RFC 4684, *Constrained Route Distribution for Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs)*

To access Internet RFCs and drafts, go to the IETF Web site at http://www.ietf.org.

Layer 3 VPN Platform Support

Layer 3 VPNs are supported on most combinations of Juniper Networks routing platforms and PICs capable of running the JUNOS software.

MX-series routers configured to be in Ethernet services mode can support some of the JUNOS software Layer 3 VPN features. For Layer 3 VPNs, Ethernet services mode supports configuring a loopback interface for a VPN routing and forwarding (VRF) instance. You can configure up to two VRF instances in Ethernet services mode. Each VRF instance can handle up to 10,000 routes. The ping mpls l3vpn operational mode command is also supported.

Layer 3 VPN Attributes

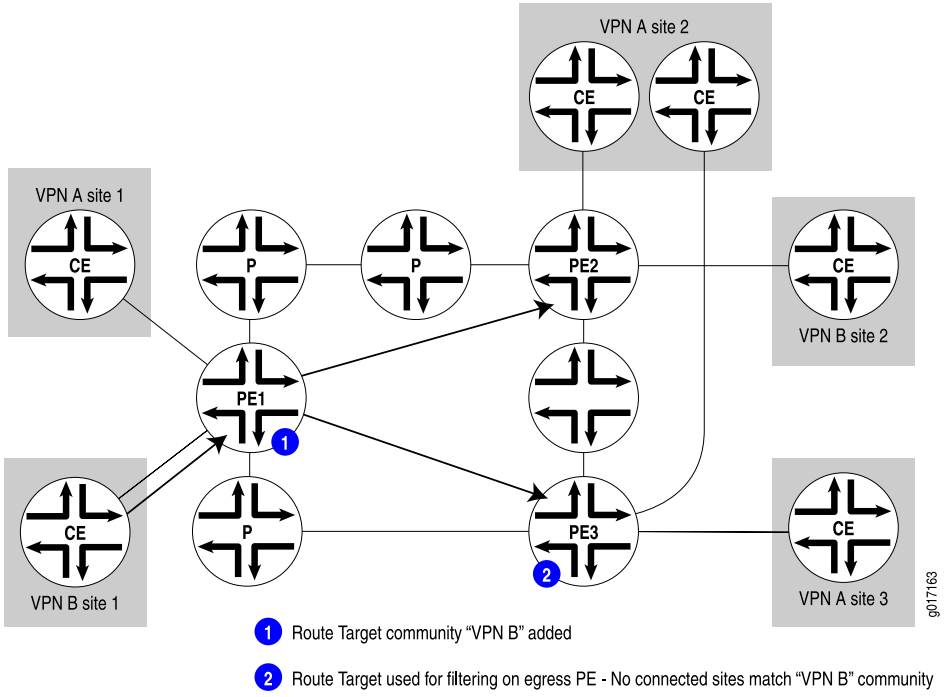
Route distribution within a VPN is controlled through BGP extended community attributes. RFC 2547 defines the following three attributes used by VPNs:

- Target VPN—Identifies a set of sites within a VPN to which a provider edge (PE) router distributes routes. This attribute is also called the *route target*. The route target is used by the egress PE router to determine whether a received route is destined for a VPN that the router services.

  Figure 8 on page 129 illustrates the function of the route target. PE Router PE1 adds the route target "VPN B" to routes received from the customer edge (CE) router at Site 1 in VPN B. When it receives the route, the egress router PE2 examines the route target, determines that the route is for a VPN that it services, and accepts the route. When the egress router PE3 receives the same route, it does not accept the route because it does not service any CE routers in VPN B.

- VPN of origin—Identifies a set of sites and the corresponding route as having come from one of the sites in that set.
- Site of origin—Uniquely identifies the set of routes that a PE router learned from a particular site. This attribute ensures that a route learned from a particular site through a particular PE-CE connection is not distributed back to the site through a different PE-CE connection. It is particularly useful if you are using BGP as the routing protocol between the PE and CE routers and if different sites in the VPN have been assigned the same autonomous system (AS) numbers.

Figure 8: VPN Attributes and Route Distribution

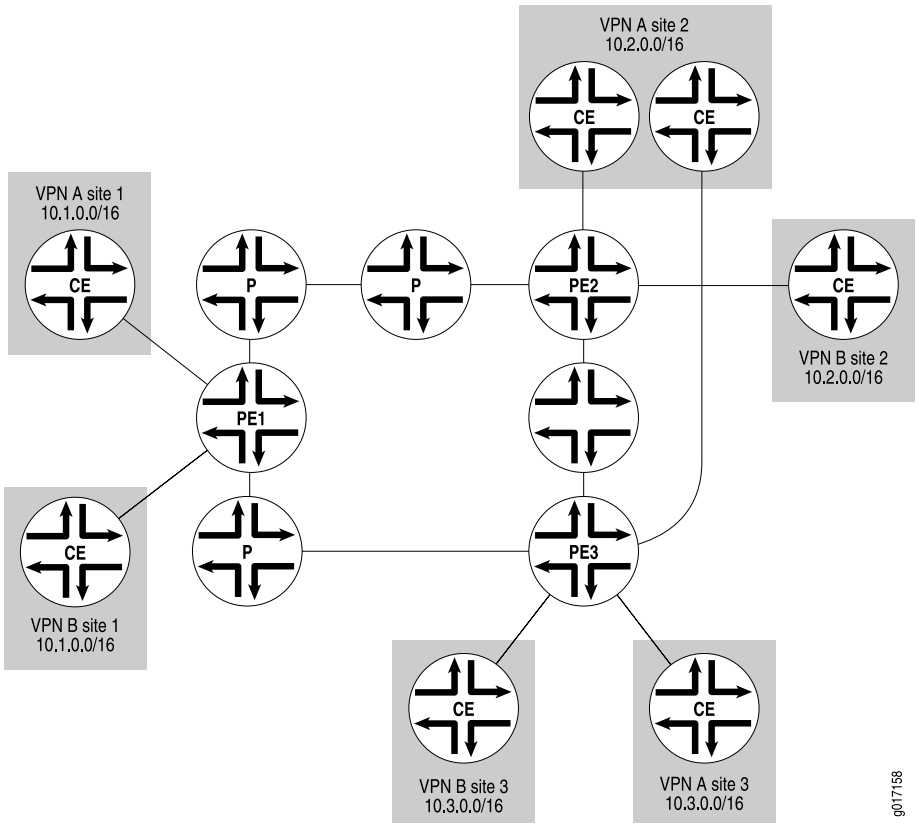


VPN-IPv4 Addresses and Route Distinguishers

Because Layer 3 VPNs connect private networks—which can use either public addresses or private addresses, as defined in RFC 1918 (*Address Allocation for Private Internets*)—over the public Internet infrastructure, when the private networks use private addresses, the addresses might overlap with the addresses of another private network.

Figure 9 on page 130 illustrates how private addresses of different private networks can overlap. Here, sites within VPN A and VPN B use the address spaces 10.1.0.0/16, 10.2.0.0/16, and 10.3.0.0/16 for their private networks.

Figure 9: Overlapping Addresses Among Different VPNs



To avoid overlapping private addresses, you can configure the network devices to use public addresses instead of private addresses. However, this is a large and complex undertaking. The solution provided in RFC 2547bis uses the existing private network numbers to create a new address that is unambiguous. The new address is part of the VPN-IPv4 address family, which is a BGP address family added as an extension to the BGP protocol. In VPN-IPv4 addresses, a value that identifies the VPN, called a route distinguisher, is prefixed to the private IPv4 address, providing an address that uniquely identifies a private IPv4 address.

Only the PE routers need to support the VPN-IPv4 address extension to BGP. When an ingress PE router receives an IPv4 route from a device within a VPN, it converts

it into a VPN-IPv4 route by adding the route distinguisher prefix to the route. The VPN-IPv4 addresses are used only for routes exchanged between PE routers. When an egress PE router receives a VPN-IPv4 route, it converts the VPN-IPv4 route back to an IPv4 route by removing the route distinguisher before announcing the route to its connected CE routers.

VPN-IPv4 addresses have the following format:

- Route distinguisher is a 6-byte value that you can specify in one of the following formats:
  - *as-number*:*number*, where *as-number* is an AS number (a 2-byte value) and *number* is any 4-byte value. The AS number can be in the range 1 through 65,535. We recommend that you use an Internet Assigned Numbers Authority (IANA)-assigned, nonprivate AS number, preferably the Internet service provider's (ISP's) own or the customer's own AS number.
  - *ip-address*:*number*, where *ip-address* is an IP address (a 4-byte value) and *number* is any 2-byte value. The IP address can be any globally unique unicast address. We recommend that you use the address that you configure in the router-id statement, which is a nonprivate address in your assigned prefix range.
- IPv4 address—4-byte address of a device within the VPN.

Figure 9 on page 130 illustrates how the AS number can be used in the route distinguisher. Suppose that VPN A is in AS 65535 and that VPN B is in AS 666 (both these AS numbers belong to the ISP), and suppose that the route distinguisher for Site 2 in VPN A is 65535:02 and that the route distinguisher for Site 2 in VPN B is 666:02. When Router PE2 receives a route from the CE router in VPN A, it converts it from its IP address of 10.2.0.0 to a VPN-IPv4 address of 65535:02:10.2.0.0. When the PE router receives a route from VPN B, which uses the same address space as VPN A, it converts it to a VPN-IPv4 address of 666:02:10.2.0.0.

If the IP address is used in the route distinguisher, suppose Router PE2's IP address is 172.168.0.1. When the PE router receives a route from VPN A, it converts it to a VPN-IPv4 address of 172.168.0.1:0:10.2.0.0/16, and it converts a route from VPN B to 172.168.0.0:1:10.2.0.0/16.

Route distinguishers are used only among PE routers to IPv4 addresses from different VPNs. The ingress PE router creates a route distinguisher and converts IPv4 routes received from CE routers into VPN-IPv4 addresses. The egress PE routers convert VPN-IPv4 routes into IPv4 routes before announcing them to the CE router.

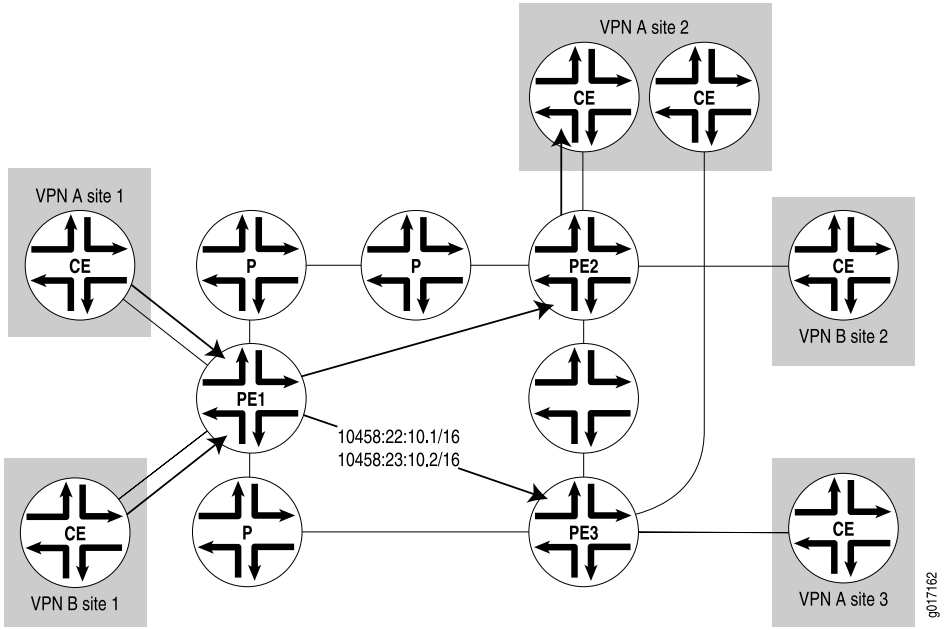
Because VPN-IPv4 addresses are a type of BGP address, you must configure internal BGP (IBGP) sessions between pairs of PE routers so that the PE routers can distribute VPN-IPv4 routes within the provider's core network. (All PE routers are assumed to be within the same AS.)

You define BGP communities to constrain the distribution of routes among the PE routers. Defining BGP communities does not, by itself, distinguish IPv4 addresses.

Figure 10 on page 132 illustrates how Router PE1 adds the route distinguisher 10458:22:10.1/16 to routes received from the CE router at Site 1 in VPN A and

forwards these routes to the other two PE routers. Similarly, Router PE1 adds the route distinguisher 10458:23:10.2/16 to routes received by the CE router at Site 1 in VPN B and forwards these routes to the other PE routers.

Figure 10: Route Distinguishers



IPv6 Layer 3 VPNs

The interfaces between the PE and CE routers of a Layer 3 VPN can be configured to carry IP version 6 (IPv6) traffic. IP allows numerous nodes on different networks to interoperate seamlessly. IPv4 is currently used in intranets and private networks, as well as the Internet. IPv6 is the successor to IPv4, and is based for the most part on IPv4.

In the Juniper Networks implementation of IPv6, the service provider implements an MPLS-enabled IPv4 backbone to provide VPN service for IPv6 customers. The PE routers have both IPv4 and IPv6 capabilities. They maintain IPv6 VPN routing and forwarding (VRF) tables for their IPv6 sites and encapsulate IPv6 traffic in MPLS frames that are then sent into the MPLS core network.

IPv6 for Layer 3 VPNs is supported for BGP and for static routes.

IPv6 over Layer 3 VPNs is described in RFC 4659, *BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN*.

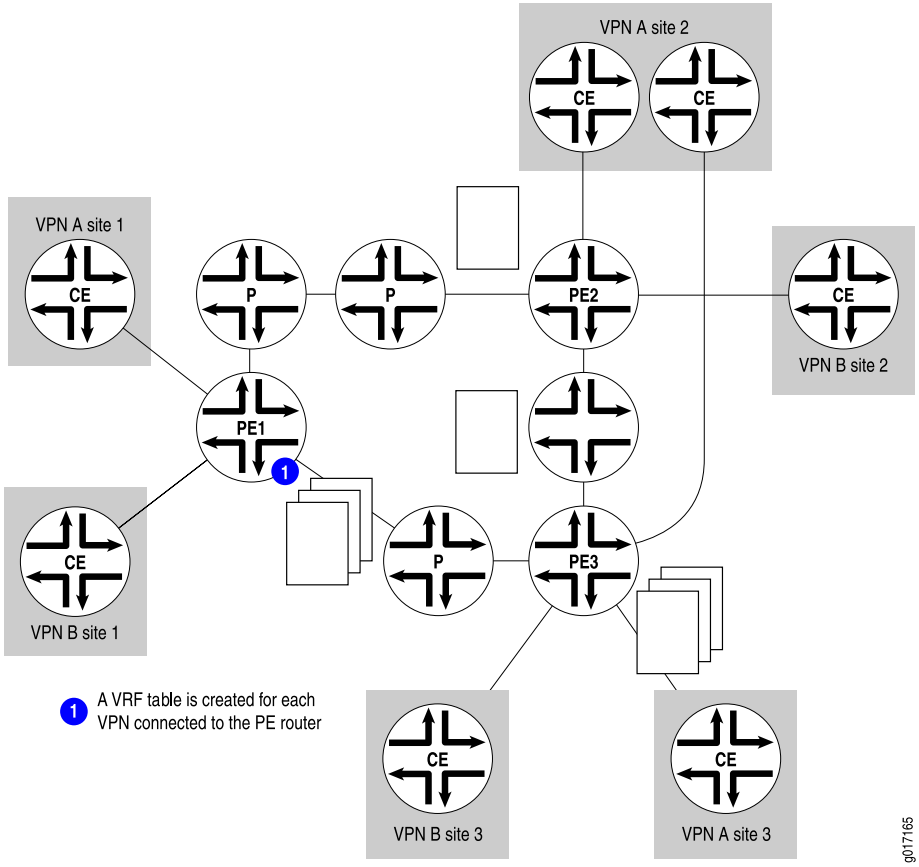
For more information about IPv6, see the *JUNOS Routing Protocols Configuration Guide*.

VPN Routing and Forwarding Tables

To separate a VPN's routes from routes in the public Internet or those in other VPNs, the PE router creates a separate routing table for each VPN, called a VPN routing and forwarding (VRF) table. The PE router creates one VRF table for each VPN that has a connection to a CE router. Any customer or site that belongs to the VPN can access only the routes in the VRF tables for that VPN.

Figure 11 on page 133 illustrates the VRF tables that are created on the PE routers. The three PE routers have connections to CE routers that are in two different VPNs, so each PE router creates two VRF tables, one for each VPN.

Figure 11: VRF Tables



Each VRF table is populated from routes received from directly connected CE sites associated with that VRF routing instance and from routes received from other PE routers that passed BGP community filtering and are in the same VPN.

Each PE router also maintains one global routing table (inet.0) to reach other routers in and outside the provider's core network.

Each customer connection (that is, each logical interface) is associated with one VRF table. Only the VRF table associated with a customer site is consulted for packets from that site.

You can configure the router so that if a next hop to a destination is not found in the VRF table, the router performs a lookup in the global routing table, which is used for Internet access.

The JUNOS software uses the following routing tables for VPNs:

- bgp.l3vpn.0—Stores all VPN-IPv4 unicast routes received from other PE routers. (This table does not store routes received from directly connected CE routers.) This table is present only on PE routers.

  When a PE router receives a route from another PE router, it places the route into its bgp.l3vpn.0 routing table. The route is resolved using the information in the inet.3 routing table. The resultant route is converted into IPv4 format and redistributed to all *routing-instance-name*.inet.0 routing tables on the PE router if it matches the VRF import policy.

  The bgp.l3vpn.0 table is also used to resolve routes over the MPLS tunnels that connect the PE routers. These routes are stored in the inet.3 routing table. PE-to-PE router connectivity must exist in inet.3 (not just in inet.0) for VPN routes to be resolved properly.

  To determine whether to add a route to the bgp.l3vpn.0 routing table, the JUNOS software checks it against the VRF instance import policies for all the VPNs configured on the PE router. If the VPN-IPv4 route matches one of the policies, it is added to the bgp.l3vpn.0 routing table. To display the routes in the bgp.l3vpn.0 routing table, use the show route table bgp.l3vpn.0 command.

- *routing-instance-name*.inet.0—Stores all unicast IPv4 routes received from directly connected CE routers in a routing instance (that is, in a single VPN) and all explicitly configured static routes in the routing instance. This is the VRF table and is present only on PE routers. For example, for a routing instance named VPN-A, the routing table for that instance is named VPN-A.inet.0.

  When a CE router advertises to a PE router, the PE router places the route into the corresponding *routing-instance-name*.inet.0 routing table and advertises the route to other PE routers if it passes a VRF export policy. Among other things, this policy tags the route with the route distinguisher (route target) that corresponds to the VPN site to which the CE belongs. A label is also allocated and distributed with the route. The bgp.l3vpn.0 routing table is not involved in this process.

  The *routing-instance-name*.inet.0 table also stores routes announced by a remote PE router that match the VRF import policy for that VPN. The remote PE router redistributed these routes from its bgp.l3vpn.0 table.

  Routes are not redistributed from the *routing-instance-name*.inet.0 table to the bgp.l3vpn.0 table; they are directly advertised to other PE routers.

  For each *routing-instance-name*.inet.0 routing table, one forwarding table is maintained in the router's Packet Forwarding Engine. This table is maintained in addition to the forwarding tables that correspond to the router's inet.0 and mpls.0 routing tables. As with the inet.0 and mpls.0 routing tables, the best routes from the *routing-instance-name*.inet.0 routing table are placed into the forwarding table.

To display the routes in the *routing-instance-name*.inet.0 table, use the show route table *routing-instance-name*.inet.0 command.

- inet.3—Stores all MPLS routes learned from Label Distribution Protocol (LDP) and Resource Reservation Protocol (RSVP) signaling done for VPN traffic. The routing table stores the MPLS routes only if the traffic-engineering bgp-igp option is not enabled.

  For VPN routes to be resolved properly, the inet.3 table must contain routes to all the PE routers in the VPN.

  To display the routes in the inet.3 table, use the show route table inet.3 command.

  Interior gateway protocol (IGP) shortcuts do not work in VPN environments and should not be configured. IGP shortcuts move routes in inet.3 to inet.0. VPN IBGP (family inet-vpn) relies on next-hops that are in the inet.3 table; thus, IGP shortcuts are incompatible with VPNs.

- inet.0—Stores routes learned by the IBGP sessions between the PE routers. To provide Internet access to the VPN sites, configure the *routing-instance-name*.inet.0 routing table to contain a default route to the inet.0 routing table.

  To display the routes in the inet.0 table, use the show route table inet.0 command.

The following routing policies, which are defined in VRF import and export statements, are specific to VRF tables.

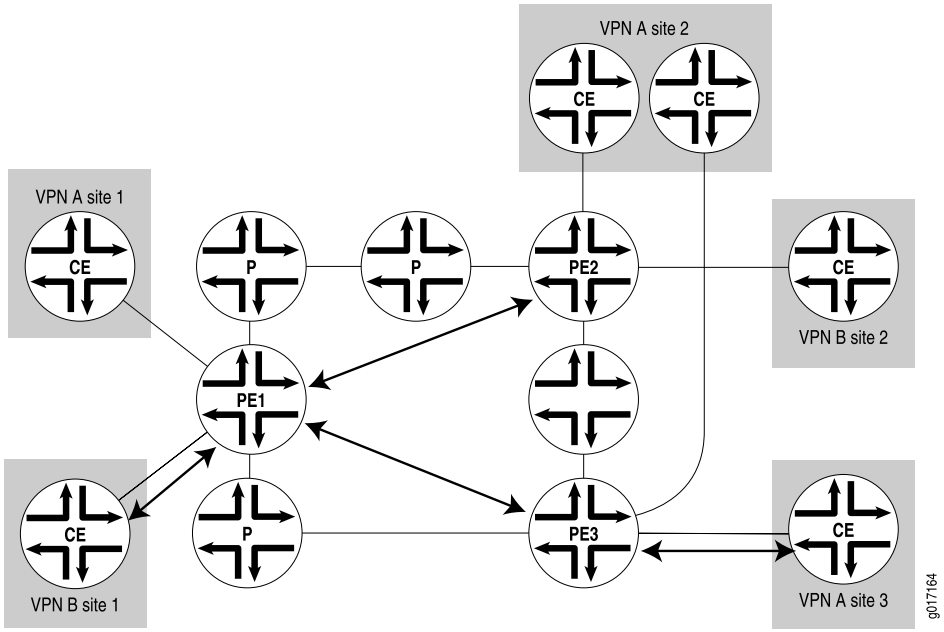
- Import policy—Applied to VPN-IPv4 routes learned from another PE router to determine whether the route should be added to the PE router's bgp.l3vpn.0 routing table. Each routing instance on a PE router has a VRF import policy.
- Export policy—Applied to VPN-IPv4 routes that are announced to other PE routers. The VPN-IPv4 routes are IPv4 routes that have been announced by locally connected CE routers.

VPN route processing differs from normal BGP route processing in one way. In BGP, routes are accepted if they are not explicitly rejected by import policy. However, because many more VPN routes are expected, the JUNOS software does not accept (and hence store) VPN routes unless the route matches at least one VRF import policy. If no VRF import policy explicitly accepts the route, it is discarded and not even stored in the bgp.l3vpn.0 table. As a result, if a VPN change occurs on a PE router—such as adding a new VRF table or changing a VRF import policy—the PE router sends a BGP route refresh message to the other PE routers (or to the route reflector if this is part of the VPN topology) to retrieve all VPN routes so they can be reevaluated to determine whether they should be kept or discarded.

Route Distribution Within a Layer 3 VPN

Within a VPN, the distribution of VPN-IPv4 routes occurs between the PE and CE routers and between the PE routers (see Figure 12 on page 136).

Figure 12: Route Distribution Within a VPN



This section discusses the following topics:

Distribution of Routes from CE to PE Routers

A CE router announces its routes to the directly connected PE router. The announced routes are in IPv4 format. The PE router places the routes into the VRF table for the VPN. In the JUNOS software, this is the *routing-instance-name*.inet.0 routing table, where *routing-instance-name* is the configured name of the VPN.

The connection between the CE and PE routers can be a remote connection (a WAN connection) or a direct connection (such as a Frame Relay or Ethernet connection).

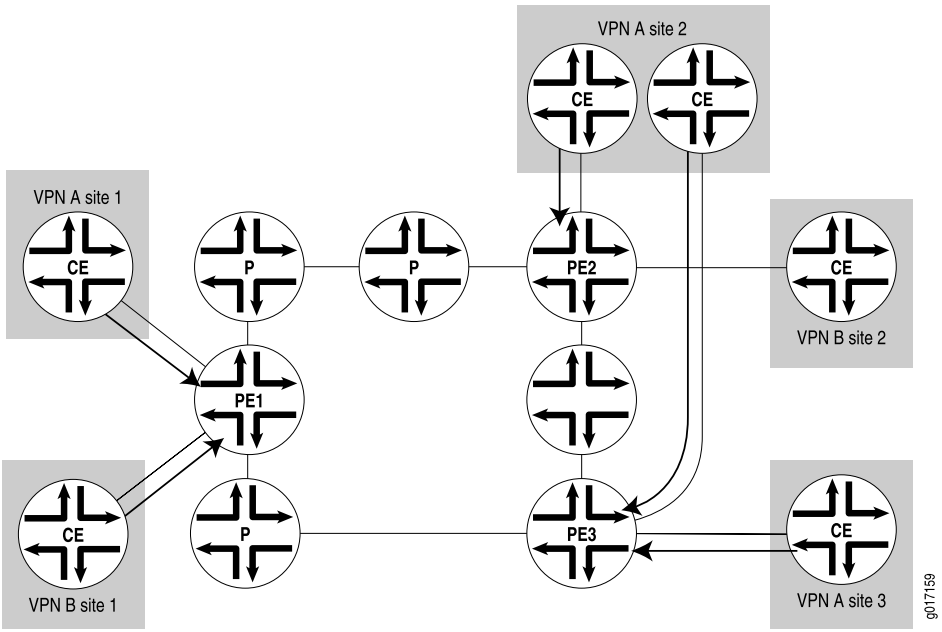
CE routers can communicate with PE routers using one of the following:

- Open Shortest Path First (OSPF)
- Routing Information Protocol (RIP)
- BGP
- Static route

Figure 13 on page 137 illustrates how routes are distributed from CE routers to PE routers. Router PE1 is connected to two CE routers that are in different VPNs.

Therefore, it creates two VRF tables, one for each VPN. The CE routers announce IPv4 routes. The PE router installs these routes into two different VRF tables, one for each VPN. Similarly, Router PE2 creates two VRF tables into which routes are installed from the two directly connected CE routers. Router PE3 creates one VRF table because it is directly connected to only one VPN.

Figure 13: Distribution of Routes from CE Routers to PE Routers



Distribution of Routes Between PE Routers

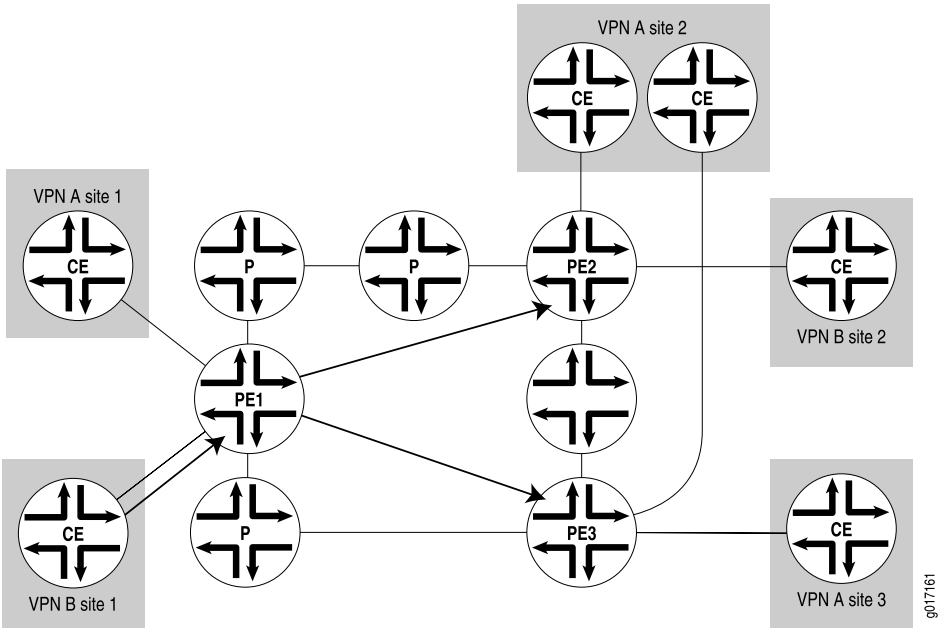
When one PE router receives routes advertised from a directly connected CE router, it checks the received route against the VRF export policy for that VPN. If it matches, the route is converted to VPN-IPv4 format—that is, the route distinguisher (route target) is added to the route. The PE router then announces the route in VPN-IPv4 format to the remote PE routers. The routes are distributed using IBGP sessions, which are configured in the provider's core network. If the route does not match, it is not exported to other PE routers, but can still be used locally for routing, for example, if two CE routers in the same VPN are directly connected to the same PE router.

The remote PE router places the route into its bgp.l3vpn.0 table if the route passes the import policy on the IBGP session between the PE routers. At the same time, it checks the route against the VRF import policy for the VPN. If it matches, the route distinguisher is removed from the route and it is placed into the VRF table (the *routing-instance-name*.inet.0 table) in IPv4 format.

Figure 14 on page 138 illustrates how Router PE1 distributes routes to the other PE routers in the provider's core network. Router PE2 and Router PE3 each have VRF

import policies that they use to determine whether to accept routes received over the IBGP sessions and install them in their VRF tables.

Figure 14: Distribution of Routes Between PE Routers



Distribution of Routes from PE to CE Routers

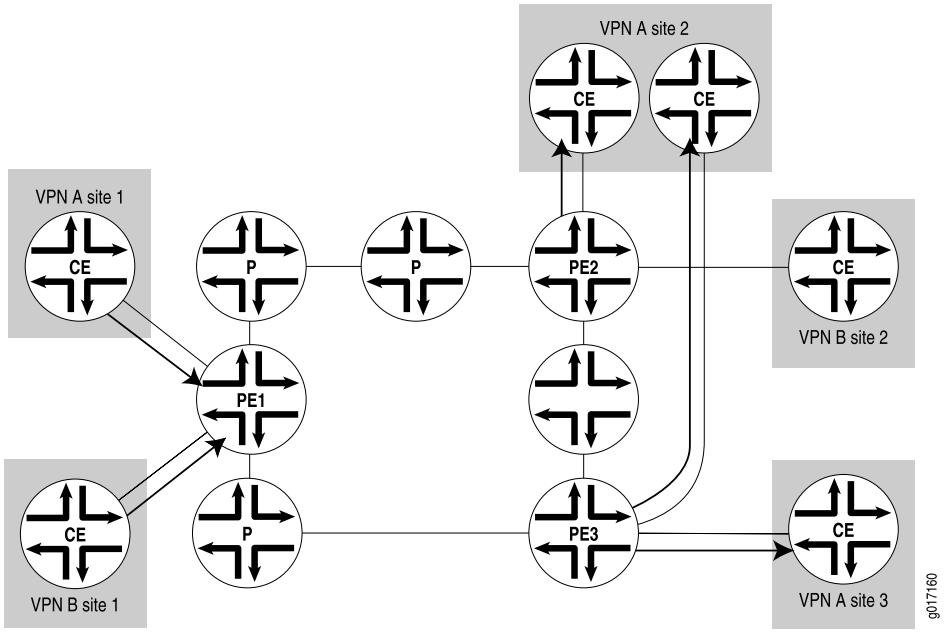
The remote PE router announces the routes in its VRF tables, which are in IPv4 format, to its directly connected CE routers.

PE routers can communicate with CE routers using one of the following routing protocols:

- OSPF
- RIP
- BGP
- Static route

Figure 15 on page 139 illustrates how the three PE routers announce their routes to their connected CE routers.

Figure 15: Distribution of Routes from PE Routers to CE Routers



Forwarding Across the Provider's Core Network

The PE routers in the provider's core network are the only routers that are configured to support VPNs and hence are the only routers to have information about the VPNs. From the point of view of VPN functionality, the provider (P) routers in the core—those P routers that are not directly connected to CE routers—are merely routers along the tunnel between the ingress and egress PE routers.

The tunnels can be either LDP or MPLS. Any P routers along the tunnel must support the protocol used for the tunnel, either LDP or MPLS.

When PE-router-to-PE router forwarding is tunneled over MPLS label-switched paths (LSPs), the MPLS packets have a two-level label stack (see Figure 16 on page 140):

- Outer label—Label assigned to the address of the BGP next hop by the IGP next hop
- Inner label—Label that the BGP next hop assigned for the packet's destination address

Figure 16: Using MPLS LSPs to Tunnel Between PE Routers

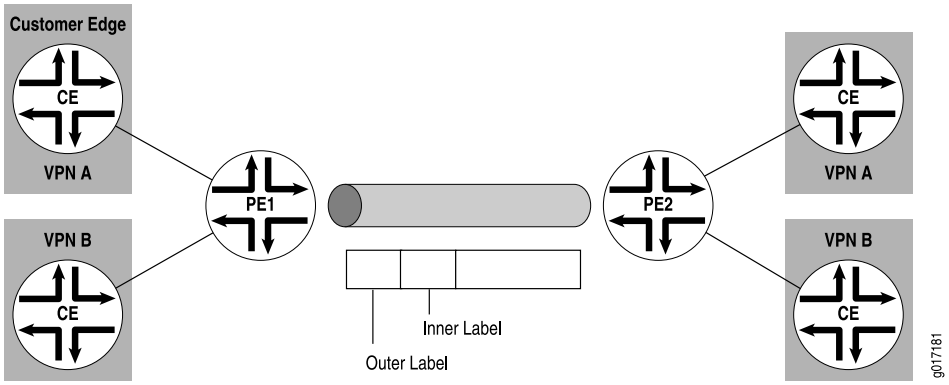
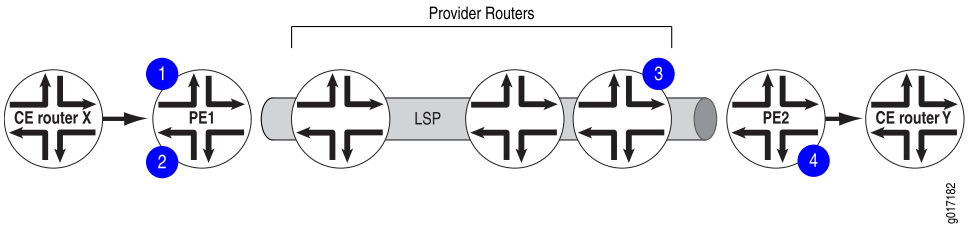


Figure 17 on page 140 illustrates how the labels are assigned and removed:

1. When CE Router X forwards a packet to Router PE1 with a destination of CE Router Y, the PE route identifies the BGP next hop to Router Y and assigns a label that corresponds to the BGP next hop and identifies the destination CE router. This label is the inner label.
2. Router PE1 then identifies the IGP route to the BGP next hop and assigns a second label that corresponds to the LSP of the BGP next hop. This label is the outer label.
3. The inner label remains the same as the packet traverses the LSP tunnel. The outer label is swapped at each hop along the LSP and is then popped by the penultimate hop router (the third P router).
4. Router PE2 pops the inner label from the route and forwards the packet to Router Y.

Figure 17: Label Stack



Routing Instances for VPNs

To implement Layer 3 VPNs in the JUNOS software, you configure one routing instance for each VPN. You configure the routing instances on PE routers only. Each VPN routing instance consists of the following components:

- VRF table—On each PE router, you configure one VRF table for each VPN.
- Set of interfaces that use the VRF table—The logical interface to each directly connected CE router must be associated with a VRF table. You can associate

more than one interface with the same VRF table if more than one CE router in a VPN is directly connected to the PE router.

- Policy rules—These control the import of routes into and the export of routes from the VRF table.
- One or more routing protocols that install routes from CE routers into the VRF table—You can use the BGP, OSPF, and RIP routing protocols, and you can use static routes.

Multicast over Layer 3 VPNs

You can configure multicast routing over a network running a Layer 3 VPN that complies with RFC 2547. This section describes this type of network application and includes these topics:

Multicast over Layer 3 VPNs Overview

In the unicast environment for Layer 3 VPNs, all VPN state information is contained within the PE routers. However, with multicast for Layer 3 VPNs, Protocol Independent Multicast (PIM) adjacencies are established in one of the following ways:

- You can set PIM adjacencies between the CE router and the PE router through a VRF instance at the [edit routing-instances *instance-name* protocols pim] hierarchy level. You must include the vpn-group-address statement at this hierarchy level, specifying a multicast group. The rendezvous point (RP) listed within the VRF-instance is the VPN customer RP (C-RP).
- You can also set the master PIM instance and the PE's IGP neighbors by configuring statements at the [edit protocols pim] hierarchy level. You must add the multicast group specified in the VRF instance to the master PIM instance. The set of master PIM adjacencies throughout the service provider network makes up the forwarding path that becomes an RP tree rooted at the service provider RP (SP-RP). Therefore, P routers within the provider core must maintain multicast state information for the VPNs.

For this to work properly, you need two types of RP routers for each VPN:

- A C-RP—An RP router located somewhere within the VPN (can be either a service provider router or a customer router).
- An SP-RP—An RP router located within the service provider network.

NOTE: A PE router can act as the SP-RP and the C-RP. Moving these multicast configuration tasks to service provider routers helps to simplify the multicast Layer 3 VPN configuration process for customers. However, configuration of both SP-RP and VPN C-RP on the same PE router is not supported.

To configure multicast over a Layer 3 VPN, you must install a Tunnel Services Physical Interface Card (PIC) on the following devices:

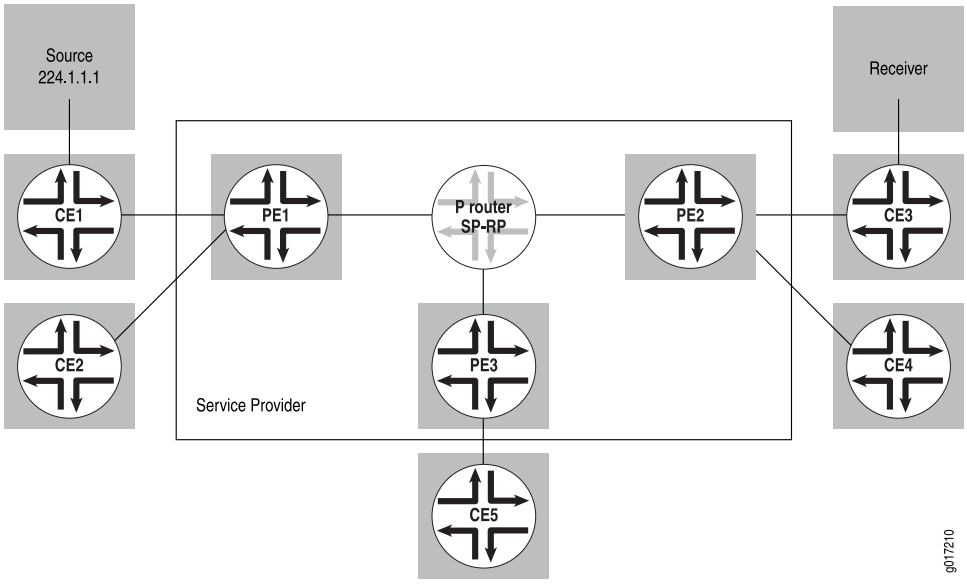
- P routers acting as RPs
- PE routers configured to run multicast routing
- CE routers acting as designated routers or as VPN-RPs

For more information about running multicast over Layer 3 VPNs, see the following documents:

- Internet draft draft-rosen-vpn-mcast-02.txt, *Multicast in MPLS/BGP VPNs*
- *JUNOS Multicast Protocols Configuration Guide*

The sections that follow describe the operation of a multicast VPN. Figure 18 on page 142 illustrates the network topology used.

Figure 18: Multicast Topology Overview



Sending PIM Hello Messages to the PE Routers

The first step in initializing multicast over a Layer 3 VPN is the distribution of a PIM Hello message from a PE router (called PE3 in this section) to all the other PE routers on which PIM is configured.

You configure PIM on the Layer 3 VPN routing instance on the PE3 router. If a Tunnel Services PIC is installed in the routing platform, a multicast interface is created. This interface is used to communicate between the PIM instance within the VRF routing instance and the master PIM instance.

The following occurs when a PIM Hello message is sent to the PE routers:

1. A PIM Hello message is sent from the VRF routing instance over the multicast interface. A generic routing encapsulation (GRE) header is prepended to the PIM Hello message. The header message includes the VPN group address and the loopback address of the PE3 router.
2. A PIM register header is prepended to the Hello message as the packet is looped through the PIM encapsulation interface. This header contains the destination address of the SP-RP and the loopback address of the PE3 router.
3. The packet is sent to the SP-RP.
4. The SP-RP removes the top header from the packet and sends the remaining GRE-encapsulated Hello message to all the PE routers.
5. The master PIM instance on each PE router handles the GRE encapsulated packet. Because the VPN group address is contained in the packet, the master instance removes the GRE header from the packet and sends the Hello message, which contains the proper VPN group address within the VRF routing instance, over the multicast interface.

Sending PIM Join Messages to the PE Routers

To receive a multicast broadcast from a multicast network, a CE router must send a PIM Join message to the C-RP. The process described in this section refers to Figure 18 on page 142.

The CE5 router needs to receive a multicast broadcast from multicast source 224.1.1.1. To receive the broadcast, it sends a PIM Join message to the C-RP (the PE3 router):

1. The PIM Join message is sent through the multicast interface, and a GRE header is prepended to the message. The GRE header contains the VPN group ID and the loopback address of the PE3 router.
2. The PIM Join message is then sent through the PIM encapsulation interface and a register header is prepended to the packet. The register header contains the IP address of the SP-RP and the loopback address of the PE3 router.
3. The PIM Join message is sent to the SP-RP by means of unicast routing.
4. On the SP-RP, the register header is stripped off (the GRE header remains) and the packet is sent to all the PE routers.
5. The PE2 router receives the packet, and because the link to the C-RP is through the PE2 router, it sends the packet through the multicast interface to remove the GRE header.
6. Finally, the PIM Join message is sent to the C-RP.

Receiving the Multicast Transmission

The steps that follow outline how a multicast transmission is propagated across the network:

1. The multicast source connected to the CE1 router sends the packet to group 224.1.1.1 (the VPN group address). The packet is encapsulated into a PIM register.
2. Because this packet already includes the PIM header, it is forwarded by means of unicast routing to the C-RP over the Layer 3 VPN.
3. The C-RP removes the packet and sends it out the downstream interfaces (which include the interface back to the CE3 router). The CE3 router also forwards this to the PE3 router.
4. The packet is sent through the multicast interface on the PE2 router; in the process, the GRE header is prepended to the packet.
5. Next, the packet is sent through the PIM encapsulation interface, where the register header is prepended to the data packet.
6. The packet is then forwarded to the SP-RP, which removes the register header, leaves the GRE header intact, and sends the packet to the PE routers.
7. PE routers remove the GRE header and forward the packet to the CE routers that requested the multicast broadcast by sending the PIM Join message.

NOTE: PE routers that have not received requests for multicast broadcasts from their connected CE routers still receive packets for the broadcast. These PE routers drop the packets as they are received.

Chapter 10

Configuring Layer 3 VPNs

To configure Layer 3 virtual private network (VPN) functionality, you must enable VPN support on the provider edge (PE) router. You must also configure any provider (P) routers that service the VPN, and you must configure the customer edge (CE) routers so that their routes are distributed into the VPN.

To configure Layer 3 VPNs, you include the following statements:

```
description text;
instance-type vrf;
interface interface-name;
route-distinguisher (as-number:id | ip-address:id);
vrf-export [ policy-names ];
vrf-import [ policy-names ];
vrf-target (community | export community-name | import community-name);
vrf-table-label;
protocols {
    bgp {
        group group-name {
            peer-as as-number;
            neighbor ip-address;
        }
        multihop ttl-value;
    }
    (ospf | ospf3) {
        area area {
            interface interface-name;
        }
        domain-id domain-id;
        domain-vpn-tag number;
        sham-link  {
            local address;
        }
        sham-link-remote address <metric number>;
    }
    pim {
        vpn-group-address address;
    }
    rip {
        rip-configuration;
    }
}
routing-options {
    autonomous-system autonomous-system {
```

```
        independent-domain;
        loops number;
    }
    forwarding-table {
        export [ policy-names ];
    }
    interface-routes {
        rib-group group-name ;
    }
    martians {
        destination-prefix match-type <allow>;
    }
    maximum-paths {
        path-limit;
        log-interval interval;
        log-only;
        threshold percentage;
    }
    maximum-prefixes {
        prefix-limit;
        log-interval interval;
        log-only;
        threshold percentage;
    }
    multipath {
        vpn-unequal-cost;
    }
    options {
        syslog (level level | upto level);
    }
    rib routing-table-name {
        martians {
            destination-prefix match-type <allow>;
        }
        multipath {
            vpn-unequal-cost;
        }
        static {
            defaults {
                static-options;
            }
            route destination-prefix {
                next-hop [next-hops];
                static-options;
            }
        }
    }
}
router-id address;
static {
    defaults {
        static-options;
    }
    route destination-prefix {
        policy [ policy-names ];
        static-options;
```

```
    }
}
```

You can include these statements at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*]

For Layer 3 VPNs, only some of the statements in the [edit routing-instances] hierarchy are valid. For the full hierarchy, see the *JUNOS Routing Protocols Configuration Guide*.

In addition to these statements, you must enable a signaling protocol, internal BGP (IBGP) sessions between the PE routers, and an interior gateway protocol (IGP) on the PE and P routers.

By default, Layer 3 VPNs are disabled.

For Layer 3 VPN configuration examples, see "Layer 3 VPN Configuration Examples" on page 201 and "Layer 3 VPN Internet Access Examples" on page 299.

Many of the configuration procedures for Layer 3 VPNs are common to all types of VPNs. These procedures are described in detail in "Configuring VPNs" on page 13.

This chapter describes how to configure Layer 3 VPNs, discussing the following topics:

Configuring VPN Routing Between the PE and CE Routers

For the PE router to distribute VPN-related routes to and from connected CE routers, you must configure routing within the VPN routing instance. You can configure a routing protocol—BGP, Open Shortest Path First (OSPF), or Routing Information Protocol (RIP)—or you can configure static routing. For the connection to each CE router, you can configure only one type of routing.

The following sections explain how to configure VPN routing between the PE and CE routers:

Configuring BGP Between the PE and CE Routers

To configure BGP as the routing protocol between the PE and the CE routers, include the bgp statement:

```
bgp {
   group group-name {
      peer-as as-number;
      neighbor ip-address;
   }
}
```

You can include the bgp statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols]

NOTE: Route reflectors and cluster IDs are not supported on a routing instance. Do not configure the cluster-id statement at the [edit routing-instances *routing-instance-name* protocols bgp group *group-name*] hierarchy level. Doing so causes the configuration to fail.

Configuring OSPF Between the PE and CE Routers

You can configure OSPF (version 2 or version 3) to distribute VPN-related routes between PE and CE routers.

The following sections describe how to configure OSPF as a routing protocol between the PE and the CE routers:

Configuring OSPF Version 2 Between the PE and CE Routers

To configure OSPF version 2 as the routing protocol between a PE and CE router, include the ospf statement:

```
ospf {
    area area {
        interface interface-name;
    }
}
```

You can include the ospf statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols]

Configuring OSPF Version 3 Between the PE and CE Routers

To configure OSPF version 3 as the routing protocol between a PE and CE router, include the ospf3 statement:

```
ospf3 {
    area area {
        interface interface-name;
    }
}
```

You can include the ospf3 statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols]

Configuring OSPF Sham Links for Layer 3 VPNs

When you configure OSPF between the PE and CE routers of a Layer 3 VPN, you can also configure OSPF sham links to compensate for issues related to OSPF intra-area links.

The following sections describe OSPF sham links and how to configure them:

OSPF Sham Links Overview

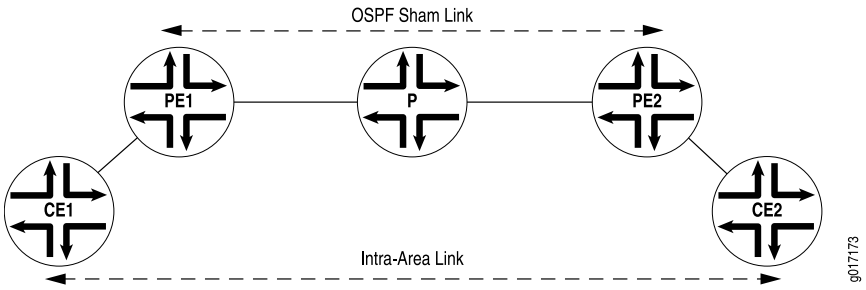
Figure 19 on page 150 provides an illustration of when you might configure an OSPF sham link. Router CE1 and Router CE2 are located in the same OSPF area. These CE routers are linked together by a Layer 3 VPN over Router PE1 and Router PE2. In

addition, Router CE1 and Router CE2 are connected by an intra-area link used as a backup.

OSPF treats the link through the Layer 3 VPN as an interarea link. By default, OSPF prefers intra-area links to interarea links, so OSPF selects the backup intra-area link as the active path. This is not acceptable in configurations where the intra-area link is not the expected primary path for traffic between the CE routers.

An OSPF sham link is also an intra-area link, except that it is configured between the PE routers as shown in Figure 19 on page 150. You can configure the metric for the sham link to ensure that the path over the Layer 3 VPN is preferred to a backup path over an intra-area link connecting the CE routers.

Figure 19: OSPF Sham Link



You should configure an OSPF sham link under the following circumstances:

- Two CE routers are linked together by a Layer 3 VPN.
- These CE routers are in the same OSPF area.
- An intra-area link is configured between the two CE routers.

If there is no intra-area link between the CE routers, you do not need to configure an OSPF sham link.

For more information on OSPF sham links, see the Internet draft draft-ietf-l3vpn-ospf-2547-01.txt, *OSPF as the PE/CE Protocol in BGP/MPLS VPNs*.

Configuring OSPF Sham Links

The sham link is an unnumbered point-to-point intra-area link and is advertised by means of a type 1 link-state advertisement (LSA). Sham links are valid only for routing instances and OSPF version 2.

Each sham link is identified by a combination of the local and remote sham link end-point address and the OSPF area to which it belongs. Sham links must be configured manually. You configure the sham link between two PE routers, both of which are within the same VRF routing instance.

You need to specify the address for the local end point of the sham link. This address is used as the source for the sham link packets and is also used by the remote PE router as the sham link remote end-point.

The OSPF sham link's local address must be specified with a loopback address for the local VPN. The route to this address must be propagated by BGP. Specify the address for the local end point using the local option of the sham-link statement:

```
sham-link {
   local address;
}
```

You can include the sham-link statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols ospf]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols ospf]

The OSPF sham link's remote address must be specified with a loopback address for the remote VPN. The route to this address must be propagated by BGP. To specify the address for the remote end point, include the sham-link-remote statement:

```
sham-link-remote address <metric number>;
```

You can include the sham-link-remote statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols ospf area *area-id*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols ospf area *area-id*]

Optionally, you can include the metric option to set a metric value for the remote end point. The metric value specifies the cost of using the link. Routes with lower total path metrics are preferred over those with higher path metrics.

You can configure a value from 1 through 65,535. The default value is 1.

OSPF Sham Links Example

This example shows how to enable OSPF sham links on a PE router.

The following is the loopback interface configuration on the PE router. The address configured is for the local end point of the OSPF sham link:

```
[edit]
interfaces {
   lo0 {
      unit 1 {
         family inet {
            address 10.1.1.1/32;
         }
      }
   }
}
```

The following is the routing instance configuration on the PE router, including the configuration for the OSPF sham link. The sham-link local statement is configured with the address for the local loopback interface:

```
[edit]
routing-instances {
   example-sham-links {
      instance-type vrf;
      interface e1-1/0/2.0;
      interface lo0.1;
      route-distinguisher 3:4;
      vrf-import vpn-red-import;
      vrf-export vpn-red-export;
      protocols {
         ospf {
            sham-link local 1-.1.1.1;
            area 0.0.0.0 {
               sham-link-remote 10.2.2.2 metric 1;
               interface e1-1/0/2.0 metric 1;
            }
         }
      }
   }
}
```

Configuring an OSPF Domain ID

For most OSPF configurations involving Layer 3 VPNs, you do not need to configure an OSPF domain ID. However, for a Layer 3 VPN connecting multiple OSPF domains, configuring OSPF domain IDs can help you control LSA translation (for Type 3 and Type 5 LSAs) between the OSPF domains and back-door paths. Each VPN routing and forwarding (VRF) table in a PE router associated with an OSPF instance is configured with the same OSPF domain ID. The default OSPF domain ID is the null value 0.0.0.0. As shown in Table 7 on page 152, a route with a null domain ID is handled differently from a route without any domain ID at all.

Table 7: How a PE Router Redistributes and Advertises Routes

Route Received	Domain ID of the Route Received	Domain ID on the Receiving Router	Route Redistributed and Advertised As
Type 3 route	A.B.C.D	A.B.C.D	Type 3 LSA
Type 3 route	A.B.C.D	E.F.G.H	Type 5 LSA
Type 3 route	0.0.0.0	0.0.0.0	Type 3 LSA
Type 3 route	Null	0.0.0.0	Type 3 LSA
Type 3 route	Null	Null	Type 3 LSA
Type 3 route	0.0.0.0	Null	Type 3 LSA
Type 3 route	A.B.C.D	Null	Type 5 LSA
Type 3 route	Null	A.B.C.D	Type 5 LSA
Type 5 route	Not applicable	Not applicable	Type 5 LSA

You can configure an OSPF domain ID for both version 2 and version 3 of OSPF. The only difference in the configuration is that you include statements at the [edit routing-instances *routing-instance-name* protocols ospf] hierarchy level for OSPF version 2 and at the [edit routing-instances *routing-instance-name* protocols ospf3] hierarchy level for OSPF version 3. The configuration descriptions that follow present the OSPF version 2 statement only. However, the substatements are also valid for OSPF version 3.

To configure an OSPF domain ID, include the domain-id statement:

```
domain-id domain-Id;
```

You can include this statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols ospf]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols ospf]

You can set a VPN tag for the OSPF external routes generated by the PE router to prevent looping. By default, this tag is automatically calculated and needs no configuration. However, you can configure the domain VPN tag for Type 5 LSAs explicitly by including the domain-vpn-tag number statement:

```
domain-vpn-tag number;
```

You can include this statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols ospf]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols ospf]

The range is 1 through 4,294,967,295 (232 - 1). If you set VPN tags manually, you must set the same value for all PE routers in the VPN.

For an example of this type of configuration, see “Configuring an OSPF Domain ID for a Layer 3 VPN” on page 263.

Hub-and-Spoke Layer 3 VPNs and OSPF Domain ID

The default behavior of an OSPF domain ID can cause the following problems for hub-and-spoke Layer 3 VPNs using OSPF between the PE and CE routers:

- PE routers set the down (DN) bit on all OSPF summary LSAs originating from area 0. PE routers are designated as area 0 by default because of the OSPF domain ID. When a PE router receives a summary LSA with the DN bit set, the LSA is not used in the OSPF calculation. This is done to prevent routing loops.

  For a hub-and-spoke Layer 3 VPN, when the hub PE router generates an OSPF summary LSA, it also sets the DN bit before sending it to the hub CE router. When the hub CE router sends the LSA back to the PE router, the PE router does not use the LSA in the OSPF calculation because the DN bit is set. Routes aggregated within the CE router are not affected.

- PE routers generating external LSAs learned from BGP updates set the domain-vpn-tag field to a value derived from the PE router's autonomous system (AS) number and an arbitrary tag. When a PE router receives an external LSA with a vpn-route-tag field that matches its own domain-vpn-tag field, the LSA is not used in the OSPF calculation. This is done to prevent routing loops.

  For a hub-and-spoke Layer 3 VPN, an external LSA originated by a hub PE router is sent to the hub CE router, which then sends it back to the same PE router. Because the vpn-route-tag field matches the PE router's domain-vpn-tag field, the LSA is not used in the OSPF calculation. Routes aggregated within the CE router are not affected.

  For hub-and-spoke Layer 3 VPNs using OSPF between the PE and CE routers to work, you need to configure the following on the hub PE router:

- Configure the disable statement at the [edit routing-instances *routing-instance-name* protocols ospf domain-id] hierarchy level on the routing instance for the hub CE router. This removes area 0 from the PE router, allowing the PE router to forward LSAs without setting the DN bit. When an LSA comes back from the hub CE router, the PE router can install it because the DN bit is not set.

- Configure 0 for the domain-vpn-tag statement at the [edit routing-instances *routing-instance-name* protocols ospf] hierarchy level on the routing instance for the spoke CE router. This removes any VPN route tags that are set on the external LSAs, preventing a VPN route tag match and allowing the PE router to install the LSA.

Configuring RIP Between the PE and CE Routers

For a Layer 3 VPN, you can configure RIP on the PE router to learn the routes of the CE router or to propagate the routes of the PE router to the CE router. RIP routes learned from neighbors configured at any [edit routing-instances] hierarchy level are added to the routing instance's inet table (*instance_name*.inet.0).

To configure RIP as the routing protocol between the PE and the CE router, include the rip statement:

```
rip {
```

```
    group group-name {
        export policy-names;
        neighbor interface-name;
    }
}
```

You can include the rip statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols]

By default, RIP does not advertise the routes it receives. To advertise routes from a PE router to a CE router, you need to configure an export policy on the PE router for RIP. For information on how to define an export policy, see the *JUNOS Policy Framework Configuration Guide*.

To specify an export policy for RIP, include the export statement:

```
export [ policy-names ];
```

You can include the export statement for RIP at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols rip group *group-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols rip group *group-name*]

To install routes learned from a RIP routing instance into multiple routing tables, include the rib-group and group statements:

```
rib-group inet group-name;
group group-name {
    neighbor interface-name;
}
```

You can include these statements at the following hierarchy levels:

- [edit protocols]
- [edit routing-instances *routing-instance-name* protocols]
- [edit logical-systems *logical-system-name* protocols]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols]

To configure a routing table group, include the rib-groups statement:

```
rib-groups group-name;
```

You can include the rib-groups statement at the following hierarchy levels:

- [edit routing-options]
- [edit logical-systems *logical-system-name* routing-options]

To add a routing table to a routing table group, include the import-rib statement. The first routing table name specified under the import-rib statement must be the name of the routing table you are configuring. For more information about how to configure routing tables and routing table groups, see the *JUNOS Routing Protocols Configuration Guide*.

```
import-rib [ group-names ]
```

You can include the import-rib statement at the following hierarchy levels:

- [edit routing-options rib-groups *group-name*]
- [edit logical-systems *logical-system-name* routing-options rib-groups *group-name*]

Configuring Static Routes Between the PE and CE Routers

To configure a static route between the PE and the CE routers, include the static statement:

```
static {
    route destination-prefix {
        next-hop [ next-hops ];
        static-options;
    }
}
```

You can include the static statement at the following hierarchy levels:

- [edit routing-options]
- [edit logical-systems *logical-system-name* routing-options]

For more information about configuring routing protocols and static routes, see the *JUNOS Routing Protocols Configuration Guide*.

Limiting the Paths and Prefixes Accepted from a CE Router

You can configure a maximum limit on the number of prefixes and paths that can be installed into the routing tables. Using prefix and path limits, you can curtail the number of prefixes and paths received from a CE router in a VPN. Prefix and path limits apply only to dynamic routing protocols, and are not applicable to static or interface routes.

To limit the number of paths accepted by a PE router from a CE router, include the maximum-paths statement:

```
maximum-paths path-limit <log-interval interval | log-only | threshold percentage>;
```

For a list of hierarchy levels at which you can configure this statement, see the statement summary section for this statement.

To limit the number of prefixes accepted by a PE router from a CE router, include the maximum-prefixes statement:

maximum-prefixes *prefix-limit* <log-interval *interval* | log-only | threshold *percentage*>;

For a list of hierarchy levels at which you can configure this statement, see the statement summary section for this statement.

A mandatory path or prefix limit, in addition to triggering a warning message, rejects any additional paths or prefixes once the limit is reached.

NOTE: Setting a path or prefix limit might result in unpredictable dynamic routing protocol behavior.

You can also configure the following options for both the maximum-paths and maximum-prefixes statements:

- log-interval—Specify the interval at which log messages are sent.
- log-only—Generate warning messages only. No limit is placed on the number of paths or prefixes stored in the routing tables.
- threshold—Generate warning messages after the specified percentage of the maximum paths or prefixes has been reached.

Configuring IPv6 Between the PE and CE Routers

You can configure IP version 6 (IPv6) between the PE and CE routers of a Layer 3 VPN. The PE router must have the PE router to PE router BGP session configured with the family inet6-vpn statement. The CE router must be capable of receiving IPv6 traffic. You can configure BGP or static routes between the PE and CE routers.

The following sections explains how to configure IPv6 VPNs between the PE routers:

Configuring IPv6 on the PE Router

To configure IPv6 between the PE and CE routers, include the family inet6-vpn statements on the PE router:

```
family inet6-vpn {
    (any | multicast | unicast) {
        aggregate-label community community-name;
        prefix-limit maximum prefix-limit;
        rib-group rib-group-name;
    }
}
```

For a list of hierarchy levels at which you can configure this statement, see the statement summary section for this statement.

You also must include the ipv6-tunneling statement:

```
ipv6-tunneling;
```

You can include the ipv6-tunneling statement at the following hierarchy levels:

- [edit protocols mpls]
- [edit logical-systems *logical-system-name* protocols mpls]

Configuring the Connection Between the PE and CE Routers

To support IPv6 routes, you must configure BGP, OSPF version 3, or static routes for the connection between the PE and CE routers in the Layer 3 VPN. You can configure BGP to handle just IPv6 routes or both IP version 4 (IPv4) and IPv6 routes.

For more information about IPv6, see the *JUNOS Routing Protocols Configuration Guide*.

The following sections explain how to configure BGP and static routes:

- Configuring BGP on the PE Router to Handle IPv6 Routes on page 158
- Configuring BGP on the PE Router for IPv4 and IPv6 Routes on page 158
- Configuring OSPF Version 3 on the PE Router on page 159
- Configuring Static Routes on the PE Router on page 159

Configuring BGP on the PE Router to Handle IPv6 Routes

To configure BGP in the Layer 3 VPN routing instance to handle IPv6 routes, include the bgp statement:

```
bgp {
    group group-name {
        local-address IPv6-address;
        family inet6 {
            unicast;
        }
        peer-as as-number;
        neighbor IPv6-address;
    }
}
```

You can include the bgp statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols]

Configuring BGP on the PE Router for IPv4 and IPv6 Routes

To configure BGP in the Layer 3 VPN routing instance to handle both IPv4 and IPv6 routes, include the bgp statement:

```
bgp {
   group group-name {
      local-address IPv4-address;
      family inet {
         unicast;
      }
      family inet6 {
         unicast;
      }
      peer-as as-number;
      neighbor address;
   }
}
```

You can include the bgp statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols]

Configuring OSPF Version 3 on the PE Router

To configure OSPF version 3 in the Layer 3 VPN routing instance to handle IPv6 routes, include the ospf3 statement:

```
ospf3 {
   area area-id {
      interface interface-name;
   }
}
```

You can include the ospf3 statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols]

For complete configuration guidelines for this statement, see the *JUNOS Routing Protocols Configuration Guide*.

Configuring Static Routes on the PE Router

To configure a static route to the CE router in the Layer 3 VPN routing instance, include the routing-options statement:

```
routing-options {
   rib routing-table.inet6.0 {
      static {
         defaults {
            static-options;
         }
      }
   }
```

```
}
```

You can include the routing-options statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*]

Configuring IPv6 on the Interfaces

You need to configure IPv6 on the PE router interfaces to the CE routers and on the CE router interfaces to the PE routers.

To configure the interface to handle IPv6 routes, include the family inet6 statement:

```
family inet6 {
   address ipv6-address;
}
```

You can include the family inet6 statement at the following hierarchy levels:

- [edit interfaces *interface-name* unit *unit-number*]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *unit-number*]

If you have configured the Layer 3 VPN to handle both IPv4 and IPv6 routes, configure the interface to handle both IPv4 and IPv6 routes by including the unit statement:

```
unit unit-number {
   family inet {
      address ipv4-address;
   }
   family inet6 {
      address ipv6-address;
   }
}
```

You can include the unit statement at the following hierarchy levels:

- [edit interfaces *interface-name*]
- [edit logical-systems *logical-system-name* interfaces *interface-name*]

Configuring EBGP or IBGP Multihop Between PE and CE Routers

You can configure an external BGP (EBGP) or IBGP multihop session between the PE and CE routers of a Layer 3 VPN. This allows you to have one or more routers between the PE and CE routers. Using IBGP between PE and CE routers does not require the configuration of any additional statements. However, using EBGP between the PE and CE routers requires the configuration of the multihop statement.

To configure an external BGP multihop session for the connection between the PE and CE routers, include the multihop statement on the PE router. To help prevent routing loops, you have to configure a time-to-live (TTL) value for the multihop session:

multihop *ttl-value*;

For the list of hierarchy levels at which you can configure this statement, see the summary section for this statement.

Configuring Layer 3 VPNs to Carry IBGP Traffic

When you configure BGP as the routing protocol between a PE router and a CE router in a Layer 3 VPN, you typically configure external peering sessions between the Layer 3 VPN service provider and the customer network ASs.

If the customer network has several sites advertising routes through an external BGP session to the service provider network and if the same AS is used by all the customer sites, the CE routers reject routes from the other CE routers. They detect a loop in the BGP AS path attribute.

To prevent the CE routers from rejecting each other's routes, you could configure the following:

- PE routers advertising routes received from remote PE routers can remap the customer network AS number to its own AS number.
- AS path loops can be configured.
- The customer network can be configured with different AS numbers at each site.

These types of configurations can work when there are no BGP routing exchanges between the customer network and other networks. However, they do have limitations for customer networks that use BGP internally for purposes other than carrying traffic between the CE routers and the PE routers. When those routes are advertised outside the customer network, the service provider ASs are present in the AS path.

To improve the transparency of Layer 3 VPN services for customer networks, you can configure the routing instance for the Layer 3 VPN to isolate the customer's network attributes from the service provider's network attributes.

When you include the independent-domain statement in the Layer 3 VPN routing instance configuration, BGP attributes received from the customer network (from the CE router) are stored in a BGP attribute (ATTRSET) that functions like a stack. When that route is advertised from the remote PE router to the remote CE router, the original BGP attributes are restored. This is the default behavior for BGP routes that are advertised to Layer 3 VPNs located in different domains.

This functionality is described in the Internet draft draft-marques-ppvpn-ibgp-version.txt, *RFC 2547bis Networks Using Internal BGP as PE-CE Protocol*.

To allow a Layer 3 VPN to transport IBGP traffic, include the independent-domain statement:

independent-domain;

You can include the statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* routing-options autonomous-system *number*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options autonomous-system *number*]

NOTE: All PE routers participating in a Layer 3 VPN configured with the independent-domain statement must be running JUNOS Release 6.3 or later.

Filtering Traffic Based on the IP Header

The vrf-table-label statement makes it possible to map the inner label to a specific VRF routing table; such mapping allows the examination of the encapsulated IP header at an egress VPN router. You might want to enable this functionality so that you can do either of the following:

- Forward traffic on a PE-router-to-CE-device interface, in a shared medium, where the CE device is a Layer 2 switch without IP capabilities (for example, a metro Ethernet switch).

  The first lookup is done on the VPN label to determine which VRF table to refer to, and the second lookup is done on the IP header to determine how to forward packets to the correct end hosts on the shared medium.

- Perform egress filtering at the egress PE router.

  The first lookup on the VPN label is done to determine which VRF routing table to refer to, and the second lookup is done on the IP header to determine how to filter and forward packets. You can enable this functionality by configuring output filters on the VRF interfaces.

  When you use the vrf-table-label statement to configure a VRF routing table, a label-switched interface (LSI) logical interface label is created and mapped to the VRF routing table.

  Any routes configured in a VRF routing table with the vrf-table-label statement are advertised with the LSI logical interface label allocated for the VRF routing table. When packets for this VPN arrive on a core-facing interface, they are treated as if the enclosed IP packet arrived on the LSI interface and are then forwarded and filtered based on the correct table.

The following sections describe how filter traffic based on the IP header:

- Configuring Traffic Filtering Based on the IP Header on page 162
- Applying MPLS EXP Classifiers to Routing Instances on page 167

Configuring Traffic Filtering Based on the IP Header

To filter traffic based on the IP header, include the vrf-table-label statement:

```
vrf-table-label;
```

You can include the vrf-table-label statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*]

You can configure the vrf-table-label statement for both IPv4 and IPv6 Layer 3 VPNs. If you configure the vrf-table-label statement for a dual-stack VRF routing table (where both IPv4 and IPv6 routes are supported), the vrf-table-label statement applies to both the IPv4 and IPv6 routes and the same label is advertised for both sets of routes.

For more information about traffic filtering based on the IP header, see the following sections:

Egress Filtering Options

You can enable egress filtering (which allows egress Layer 3 VPN PE routers to perform lookups on the VPN label and IP header at the same time) by including the vrf-table-label statement at the [edit routing-instances *instance-name*] hierarchy level. However, there are many limitations on when you can configure the vrf-table-label statement. For more information, see "Support for ATM and Frame Relay Interfaces" on page 164 and "Other Limitations" on page 166. There is no restriction on CE-router-to-PE-router interfaces.

You can also enable egress filtering by configuring a VPN tunnel (VT) interface on routing platforms equipped with a Tunnel Services Physical Interface Card (PIC). When you enable egress filtering this way, there is no restriction on the type of core-facing interface used. There is also no restriction on the type of CE-router-to-PE-router interface used.

Support for Ethernet, SONET/SDH, and T1/T3/E3 Interfaces

Support for the vrf-table-label statement over Ethernet, SONET/SDH, and DS3/T3 interfaces is available on the Juniper Networks routing platforms summarized in Table 8 on page 164.

Table 8: Support for Ethernet and SONET/SDH Interfaces

Interfaces	J-series	M-series Without an Enhanced FPC	M-series with an Enhanced FPC	M320	T-series
Ethernet	Yes	Yes	Yes	Yes	Yes
SONET/SDH	N/A	Yes	Yes	Yes	Yes
T1/T3/E3	Yes	Yes	Yes	Yes	Yes

Only the following Ethernet PICs support the vrf-table-label statement on M-series routers without enhanced FPCs:

- 1-port Gigabit Ethernet
- 2-port Gigabit Ethernet
- 4-port Fast Ethernet

Support for Aggregated and VLAN Interfaces

Support for the vrf-table-label statement over aggregated and VLAN interfaces is available on the Juniper Networks routing platforms summarized in Table 9 on page 164.

Table 9: Support for Aggregated and VLAN Interfaces

Interfaces	J-series	M-series Without an Enhanced FPC	M-series with an Enhanced FPC	M320	T-series
Aggregated	N/A	No	Yes	Yes	Yes
VLAN	Yes	No	Yes	Yes	Yes

NOTE: The vrf-table-label statement for aggregated Gigabit Ethernet, 10 Gigabit Ethernet, and VLAN physical interfaces is not supported on M120 routing platforms.

Support for ATM and Frame Relay Interfaces

Support for the vrf-table-label statement over Asynchronous Transfer Mode (ATM) and Frame Relay interfaces is available on the Juniper Networks routing platforms summarized in Table 10 on page 165.

Table 10: Support for ATM and Frame Relay Interfaces

Interfaces	J-series	M-series Without an Enhanced FPC	M-series with an Enhanced FPC	M320	T-series
ATM1	N/A	No	No	No	No
ATM2 intelligent queuing (IQ)	N/A	No	Yes	Yes	Yes
Frame Relay	Yes	No	Yes	Yes	Yes
Channelized	N/A	No	No	No	No

When you configure the vrf-table-label statement, be aware of the following limitations with ATM or Frame Relay interfaces:

- The vrf-table-label statement is supported on ATM interfaces, but with the following limitations:
  - ATM interfaces can be configured on a T-series routing platform, on an M320, or on an M-series router fitted with an enhanced FPC.
  - The interface can only be a PE router interface receiving traffic from a P router.
  - The router must have an ATM2 IQ PIC.
- The vrf-table-label statement is also supported with Frame Relay encapsulated interfaces, but with the following limitations:
  - Frame Relay interfaces can be configured on a T-series routing platform, on an M320, or on an M-series router fitted with an enhanced FPC.
  - The interface can only be a PE router interface receiving traffic from a P router.

Support for Multilink PPP and Multilink Frame Relay Interfaces

Support for the vrf-table-label statement over Multilink Point-to-Point Protocol (MLPPP) and Multilink Frame Relay (MLFR) interfaces is available on the Juniper Networks routing platforms summarized in Table 11 on page 165.

Table 11: Support for Multilink PPP and Multilink Frame Relay Interfaces

Interfaces	J-series	M-series Without an Enhanced FPC	M-series with an Enhanced FPC	M320	T-series	MX-series
MLPPP	Yes	No	Yes	No	No	No
End-to-End MLFR (FRF.15)	Yes	No	Yes	No	No	No
UNI/NNI MLFR (FRF.16)	Yes	No	No	No	No	No

M-series routing platforms require an AS PIC to support the vrf-table-label statement over MLPPP and MLFR interfaces. The vrf-table-label statement over MLPPP interfaces is not supported on M120 routing platforms.

Support for Packets with Null Top Labels

You can configure the vrf-table-label statement on core-facing interfaces receiving MPLS packets with a null top label, which might be transmitted by some vendors' equipment. These packets can be received only on M320 and T-series routing platforms using one of the following PICs:

- 1-port Gigabit Ethernet with SFP
- 2-port Gigabit Ethernet with SFP
- 4-port Gigabit Ethernet with SFP
- 10–port Gigabit Ethernet with SFP
- 1-port SONET STM4
- 4-port SONET STM4
- 1-port SONET STM16
- 1-port SONET STM16 (non-SFP)
- 4-port SONET STM16
- 1-port SONET STM64

The following PICs can receive packets with null top labels, but only when installed in an M120 router or an M320 router with an Enhanced III FPC:

- 1–port 10 Gigabit Ethernet
- 1–port 10 Gigabit Ethernet IQ2

Other Limitations

When you configure the vrf-table-label statement, be aware of the following other limitations:

- The time-to-live (TTL) value in the MPLS header is not copied back to the IP header of packets sent from the PE router to the CE router.
- You cannot configure a virtual loopback tunnel interface and the vrf-table-label statement on the same routing instance. Doing so causes the commit to fail.
- Do not use the vrf-table-label statement for source class usage/destination class usage (SCU/DCU) configurations. For information on SCU/DCU configuration, see the *JUNOS Network Interfaces Configuration Guide*.
- You can configure the vrf-table-label statement on Multilink Frame Relay (MLFR FRF.16) encapsulated PE-router-to-P-router interfaces, but only on J-series routing platforms.
- When you configure the vrf-table-label statement, MPLS packets with label-switched interface (LSI) labels that arrive on core-facing ATM or Frame

Relay interfaces, or on aggregated Ethernet interfaces configured with VLANs or Ethernet interfaces configured with VLANs, are not counted at the logical interface level.

- You cannot configure the vrf-table-label statement within a VRF routing instance if the PE-router-to-P-router interface is any of the following:
  - Aggregated SONET/SDH interfaces
  - All channelized interfaces
  - All tunnel interfaces (for example, generic routing encapsulation [GRE] and IP Security [IPSec])
  - Circuit cross-connect (CCC) and translational cross-connect (TCC) encapsulated interfaces
  - Logical tunnel interfaces
  - Virtual private LAN service (VPLS) encapsulated interfaces

NOTE: All CE-router-to-PE-router and PE-router-to-CE-router interfaces are supported.

- You cannot configure the vrf-table-label statement within a VRF routing instance if the PE-router-to-P-router PIC is one of the following:
  - 10-port E1 PIC
  - 8-port Fast Ethernet PIC
  - 12-port Fast Ethernet PIC
  - 48-port Fast Ethernet PIC
  - All ATM PICs, except the ATM2 IQ PIC

Applying MPLS EXP Classifiers to Routing Instances

When you configure the vrf-table-label statement, and you do not explicitly apply a classifier configuration to the routing instance, the default MPLS EXP classifier is applied to the routing instance.

For PICs that are installed on Enhanced FPCs, you can override the default MPLS EXP classifier and apply a custom classifier to the routing instance. Detailed instructions for this procedure are provided in the *JUNOS Network Interfaces Configuration Guide*. The following instructions summarize how to apply a custom classifier to a routing instance:

1. Filter traffic based on the IP header by including the vrf-table-label statement at the [edit routing-instances *routing-instance-name*] hierarchy level:

   vrf-table-label;

2. Configure a custom MPLS EXP classifier by including the following statements in the configuration. See the *JUNOS Network Interfaces Configuration Guide* for information on how to do this.

3. Configure the routing instance for CoS by including the routing-instances statement at the [edit class-of-service] hierarchy level:

```
routing-instances routing-instance-name {
    classifiers {
        exp (classifier-name | default);
    }
}
```

4. Configure the routing instance to use the custom MPLS EXP classifier by including the classifiers statement at the [edit class-of-service routing-instances *routing-instance-name*] hierarchy level:

```
classifiers {
    exp classifier-name;
}
```

To display the MPLS EXP classifiers associated with all routing instances, issue the show class-of-service routing-instances command.

NOTE: The following caveats apply to custom MPLS EXP classifiers for routing instances:

- An Enhanced FPC is required.
- Logical systems are not supported.

Configuring a VPN Tunnel for VRF Table Lookup

You can configure a VPN tunnel to facilitate VRF table lookup based on MPLS labels. You might want to enable this functionality to forward traffic on a PE-router-to-CE-device interface in a shared medium, where the CE device is a Layer 2 switch without IP capabilities (for example, a metro Ethernet switch), or to perform egress filtering at the egress PE router.

For more information on VPN tunnels and VT interfaces, see the *JUNOS Services Interfaces Configuration Guide*.

Configuring a Logical Unit on the Loopback Interface

For Layer 3 VPNs (VRF routing instances), you can configure a logical unit on the loopback interface into each VRF routing instance that you have configured on the router. Associating a VRF routing instance with a logical unit on the loopback interface allows you to easily identify the VRF routing instance.

Doing this is useful for troubleshooting:

- It allows you to ping a remote CE router from a local PE router in a Layer 3 VPN. For more information, see "Pinging the Remote CE Router from the Local PE Router" on page 197.
- It ensures that a path maximum transmission unit (MTU) check on traffic originating on a VRF or virtual-router routing instance functions properly. For more information, see "Configuring a Path MTU Check for VPNs" on page 39.

You can also configure a firewall filter for the logical unit on the loopback interface; this configuration allows you to filter traffic for the VRF routing instance associated with it.

The following describes how firewall filters affect the VRF routing instance depending on whether they are configured on the default loopback interface, the VRF routing instance, or some combination of the two. The "default loopback interface" refers to lo0.0 (associated with the default routing table), and the "VRF loopback interface" refers to lo0.*n*, which is configured in the VRF routing instance.

- If you configure Filter A on the default loopback interface and Filter B on the VRF loopback interface, the VRF routing instance uses Filter B.
- If you configure Filter A on the default loopback interface but do not configure a filter on the VRF loopback interface, the VRF routing instance does not use a filter.
- If you configure Filter A on the default loopback interface but do not even configure a VRF loopback interface, the VRF routing instance uses Filter A.

To configure a logical unit on the loopback interface, include the unit statement:

```
unit number {
    family inet {
        address address;
    }
}
```

You can include the unit statement at the following hierarchy levels:

- [edit interfaces lo0]
- [edit logical-systems *logical-system-name* interfaces lo0]

To associate a firewall filter with the logical unit on the loopback interface, include the filter statement:

```
filter {
    input filter-name;
}
```

You can include the filter statement at the following hierarchy levels:

- [edit interfaces lo0 unit *unit-number* family inet]
- [edit logical-systems *logical-system-name* interfaces lo0 unit *unit-number* family inet]

To include the lo0.*n* interface (where *n* specifies the logical unit) in the configuration for the VRF routing instance, include the following statement:

```
interface lo0.n;
```

You can include this statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*]

For more information on how to configure firewall filters, see the *JUNOS Policy Framework Configuration Guide*.

Configuring Multicast over Layer 3 VPNs

You can configure two types of multicast Layer 3 VPNs using the JUNOS software:

- Draft Rosen multicast VPNs—Draft Rosen multicast VPNs are 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* (expired April 2004).
- Next generation multicast VPNs—Next generation multicast VPNs are described in Internet drafts draft-ietf-l3vpn-2547bis-mcast-bgp-03.txt, *BGP Encodings for Multicast in MPLS/BGP IP VPNs* and draft-ietf-l3vpn-2547bis-mcast-02.txt, *Multicast in MPLS/BGP IP VPNs*.

This section describes how to configure draft Rosen multicast VPNs. This information is provided to you in case you already have dual PIM multicast VPNs configured on your network. For information on how to configure next generation multicast VPNs, see “Multicast VPNs” on page 349.

You can configure a Layer 3 VPN to support multicast traffic using the Protocol Independent Multicast (PIM) routing protocol. To support multicast, you need to configure PIM on routers within the VPN and within the service provider’s network.

Each PE router configured to run multicast over Layer 3 VPNs must have a Tunnel Services PIC. A Tunnel Services PIC is also required on the P routers that act as rendezvous points (RPs). Tunnel Services PICs are also needed on all the CE routers acting as designated routers (first-hop/last-hop routers) or as RPs, just as they are in non-VPN PIM environments.

Configure the master PIM instance at the [edit protocols pim] hierarchy level on the CE and PE routers. This master PIM instance configuration on the PE router should match the configuration on the service providers core routers.

You also need to configure a PIM instance for the Layer 3 VPN at the [edit routing-instances *routing-instance-name* protocols pim] hierarchy level on the PE router. This creates a PIM instance for the indicated routing instance. The configuration of the PIM instance on the PE router should match the PIM instance configured on the CE router the PE router is connected to.

For information about how to configure PIM, see the *JUNOS Multicast Protocols Configuration Guide*.

You use the vpn-apply-export statement to configure the group address designated for the VPN in the service provider's network. This address should be unique for each VPN and configured on the VRF routing instance of all PE routers connecting to the same VPN. It ensures that multicast traffic is transmitted only to the specified VPN.

Include the vpn-apply-export statement:

```
vpn-apply-export address;
```

You can include the vpn-apply-export statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols pim]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim]

The rest of the Layer 3 VPN configuration for multicast is conventional and is described in other sections of this manual. Most of the specific configuration tasks needed to activate multicast in a VPN environment involve PIM. For more information about how to configure PIM and multicast in JUNOS, including an example of how to configure multicast over Layer 3 VPNs, see the *JUNOS Multicast Protocols Configuration Guide*.

Configuring Packet Forwarding for Layer 3 VPNs

You can configure the router to support packet forwarding for IPv4 traffic in Layer 2 and Layer 3 VPNs. Packet forwarding is handled in one of the following ways, depending on the type of helper service configured:

- BOOTP service—Clients send Bootstrap Protocol (BOOTP) requests through the router configured with BOOTP service to a server in the specified routing instance. The server recognizes the client address and sends a response back to the router configured with BOOTP service. This router forwards the reply to the correct client address in the specified routing instance.
- Other services—Clients send requests through the router configured with the service to a server in the specified routing instance. The server recognizes the client address and sends a response to the correct client address in the specified routing instance.

To enable packet forwarding for VPNs, include the helpers statement:

```
helpers {
    service {
        description description-of-service;
        server {
            address address {
                routing-instance routing-instance-names;
            }
        }
```

```
        interface interface-name {
            description description-of-interface;
            no-listen;
            server {
                address address {
                    routing-instance routing-instance-names;
                }
            }
        }
    }
}
```

You can include the helpers statement at the following hierarchy levels:

- [edit forwarding-options]
- [edit logical-systems *logical-system-name* forwarding-options]
- [edit routing-instances *routing-instance-name* forwarding-options]

NOTE: You can enable packet forwarding for multiple VPNs. However, the client and server must be within the same VPN. Any Juniper Networks routing platforms with packet forwarding enabled along the path between the client and server must also reside within the same VPN.

The address and routing instance together constitute a unique server. This has implications for routers configured with BOOTP service, which can accept multiple servers.

For example, a BOOTP service can be configured as follows:

```
[edit forwarding-options helpers bootp]
server address 10.2.3.4 routing-instance [instance-A instance-B];
```

Even though the addresses are identical, the routing instances are different. A packet coming in for BOOTP service on instance-A is forwarded to 10.2.3.4 in the instance-A routing instance, while a packet coming in on instance-B is forwarded in the instance-B routing instance. Other services can only accept a single server, so this configuration does not apply in those cases.

For more information about the statements configured at the [edit forwarding-options] hierarchy level, see the *JUNOS Policy Framework Configuration Guide*.

Configuring GRE Tunnels for Layer 3 VPNs

JUNOS software allows you to configure a generic routing encapsulation (GRE) tunnel between the PE and CE routers for a Layer 3 VPN. The GRE tunnel can have one or more hops.

For more information about how to configure tunnel interfaces, see the *JUNOS Services Interfaces Configuration Guide*.

You can configure the GRE tunnels manually or configure the JUNOS software to instantiate GRE tunnels dynamically.

The following sections describe how to configure GRE tunnels manually and dynamically:

Configuring GRE Tunnels Manually Between PE and CE Routers

The following sections explain how to configure a GRE tunnel between the PE and CE routers for a Layer 3 VPN:

Configuring the GRE Tunnel Interface on the PE Router

You configure the GRE tunnel as a logical interface on the PE router. To configure the GRE tunnel interface, include the unit statement:

```
unit logical-unit-number {
    tunnel {
        source source-address;
        destination destination-address;
        routing-instance {
            destination routing-instance-name;
        }
    }
    family inet {
        address address;
    }
}
```

You can include the unit statement at the following hierarchy levels:

- [edit interfaces *interface-name*]
- [edit logical-systems *logical-system-name* interfaces *interface-name*]

As part of the GRE tunnel interface configuration, you need to include the following statements:

- source *source-address*—Specify the source or origin of the GRE tunnel.
- destination *destination-address*—Specify the destination or end point of the GRE tunnel.

By default, the tunnel destination address is assumed to be in the default Internet routing table, inet.0. If the tunnel destination address is not in inet.0, you need to specify which routing table to search for the tunnel destination address by configuring the routing-instance statement. This is the case if the tunnel encapsulating interface is also configured under the routing instance.

- destination *routing-instance-name*—Specify the name of the routing instance when configuring the GRE tunnel interface on the PE router.

To complete the GRE tunnel interface configuration, include the interface statement for the GRE interface under the appropriate routing instance:

```
interface interface-name;
```

You can include the interface statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*]

Configuring the GRE Tunnel Interface on the CE Router

To configure the GRE tunnel interface on the CE router, include the unit statement:

```
unit logical-unit-number {
    tunnel {
        source address;
        destination address;
    }
    family inet {
        address address;
    }
}
```

You can include the unit statement at the following hierarchy levels:

- [edit interfaces *interface-name*]
- [edit logical-systems *logical-system-name* interfaces *interface-name*]

Configuring GRE Tunnels Dynamically

When the router receives a VPN route to a BGP next-hop address but no MPLS path is available, a GRE tunnel can be dynamically generated to carry the VPN traffic across the BGP network. The GRE tunnel is generated and then its routing information is copied into the inet.3 routing table.

NOTE: IPv4 routes are the only type of routes supported for dynamic GRE tunnels. Also, the routing platform must have a tunnel PIC.

To generate GRE tunnels dynamically, include the dynamic-tunnels statement:

```
dynamic-tunnels tunnel-name {
    destination-networks prefix;
    source-address address;
    tunnel-type gre;
}
```

You can include this statement at the following hierarchy levels:

- [edit routing-options]
- [edit logical-systems *logical-system-name* routing-options]

Specify the IPv4 prefix range (for example, 10/8 or 11.1/16) for the destination network by including the destination-networks statement. Only tunnels within the specified IPv4 prefix range are allowed to be initiated.

```
destination-networks prefix;
```

You can include this statement at the following hierarchy levels:

- [edit routing-options dynamic-tunnels *tunnel-name*]
- [edit logical-systems *logical-system-name* routing-options dynamic-tunnels *tunnel-name*]

Specify the source address for the GRE tunnels by including the source-address statement. The source address specifies the address used as the source for the local tunnel endpoint. This could be any local address on the router (typically the router ID or the loopback address).

```
source-address address;
```

You can include this statement at the following hierarchy levels:

- [edit routing-options dynamic-tunnels *tunnel-name*]
- [edit logical-systems *logical-system-name* routing-options dynamic-tunnels *tunnel-name*]

Specify the type of tunnel to be dynamically created by including the tunnel-type statement. The only currently valid value is gre (for GRE tunnels).

```
tunnel-type gre;
```

You can include this statement at the following hierarchy levels:

- [edit routing-options dynamic-tunnels *tunnel-name*]
- [edit logical-systems *logical-system-name* routing-options dynamic-tunnels *tunnel-name*]

Configuring an ES Tunnel Interface for Layer 3 VPNs

An ES tunnel interface allows you to configure an IP Security (IPSec) tunnel between the PE and CE routers of a Layer 3 VPN. The IPSec tunnel can include one or more hops.

The following sections explain how to configure an ES tunnel interface between the PE and CE routers of a Layer 3 VPN:

Configuring the ES Tunnel Interface on the PE Router

To configure the ES tunnel interface on the PE router, include the unit statement:

```
unit logical-unit-number {
    tunnel {
        source source-address;
        destination destination-address;
    }
    family inet {
        address address;
        ipsec-sa security-association-name;
    }
}
```

You can include the unit statement at the following hierarchy levels:

- [edit interfaces *interface-name*]
- [edit logical-systems *logical-system-name* interfaces *interface-name*]

By default, the tunnel destination address is assumed to be in the default Internet routing table, inet.0. For IPSec tunnels using manual security association (SA), if the tunnel destination address is not in the default inet.0 routing table, you need to specify which routing table to search for the tunnel destination address by configuring the routing-instance statement. This is the case if the tunnel encapsulating interface is also configured under the routing instance.

```
unit logical-unit-number {
    tunnel {
        source address;
        destination address;
        routing-instance {
            destination routing-instance-name;
        }
        family inet {
            address address;
            ipsec-sa security-association-name;
        }
        family mpls;
    }
}
```

You can include these statements at the following hierarchy levels:

- [edit interfaces *interface-name*]
- [edit logical-systems *logical-system-name* interfaces *interface-name*]

NOTE: For IPSec tunnels using dynamic SA, the tunnel destination address must be in the default Internet routing table, inet.0.

To complete the ES tunnel interface configuration, include the interface statement for the ES interface under the appropriate routing instance:

```
interface interface-name;
```

You can include the interface statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*]

Configuring the ES Tunnel Interface on the CE Router

To configure the ES tunnel interface on the CE router, include the unit statement:

```
unit 0 {
    tunnel {
        source address;
        destination address;
    }
    family inet {
        address address;
        ipsec-sa security-association-name;
    }
}
```

You can include the unit statement at the following hierarchy levels:

- [edit interfaces *interface-name*]
- [edit logical-systems *logical-system-name* interfaces *interface-name*]

For more information about how to configure tunnel interfaces, see the *JUNOS Services Interfaces Configuration Guide*.

For more information about how to configure IPSec interfaces, see the *JUNOS System Basics Configuration Guide*.

Configuring IPSec Instead of MPLS Between PE Routers

A conventional Layer 3 BGP/MPLS VPN requires the configuration of MPLS label-switched paths (LSPs) between the PE routers. When a PE router receives a packet from a CE router, it performs a lookup in a specific VRF table for the IP destination address and obtains a corresponding MPLS label stack. The label stack is used to forward the packet to the egress PE router, where the bottom label is removed and the packet is forwarded to the specified CE router.

You can provide Layer 3 BGP/MPLS VPN service without an MPLS backbone. Instead of configuring MPLS LSPs between the PE routers, you configure GRE and IPSec tunnels between the PE routers. The MPLS information for the VPN (the VPN label) is encapsulated within an IP header and an IPSec header. The source address of the IP header is the address of the ingress PE router. The destination address has the BGP next hop, the address of the egress PE router.

NOTE: The IPSec tunnel requires the use of an ES PIC. The GRE tunnel requires the use of a Tunnel Services PIC.

To configure IPSec between PE routers, follow these steps:

1. Configure an IPSec tunnel between the PE routers. The source address is that of the ingress PE router, and the destination address is that of the egress PE router:

```
es-interface-name {
    unit unit-number {
        tunnel {
            source source-address;
            destination destination-address;
        }
        family inet {
            ipsec-sa sa-esp-dynamic;
            address address;
        }
        family mpls;
    }
}
```

   You can include these statements at the following hierarchy levels:

   - [edit interfaces]
   - [edit logical-systems *logical-system-name* interfaces]

2. Configure IPSec on the PE router. For information about how to configure IPSec, see the *JUNOS System Basics Configuration Guide*.

3. Configure a GRE tunnel between the PE routers. Again, the source address is that of the ingress PE router, and the destination address is that of the egress PE router:

```
gr-interface-name {
    unit unit-number {
        family inet {
            address address;
        }
        family mpls;
        tunnel {
            source source-address;
            destination destination-address;
        }
    }
}
```

You can include these statements at the following hierarchy levels:

- [edit interfaces]
- [edit logical-systems *logical-system-name* interfaces]

4. Configure BGP between the PE routers:

```
bgp {
   group pe {
      type internal;
      local-address local-address;
      family inet {
         unicast;
      }
      family inet-vpn {
         unicast;
      }
      peer-as as-number;
      neighbor address;
   }
}
```

You can include these statements at the following hierarchy levels:

- [edit protocols]
- [edit logical-systems *logical-system-name* protocols]

5. Configure the routing instance:

```
instance-type vrf;
interface interface-name;
route-distinguisher address;
vrf-import import-policy-name;
vrf-export export-policy-name;
protocols {
   bgp {
      group routing-instance-name {
         type external;
         peer-as as-number;
         as-override;
         neighbor address;
      }
   }
}
```

You can include these statements at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*]

6. Configure the policy options:

```
policy-statement import-policy-name {
   term 1 {
      from {
         protocol bgp;
```

```
            community community-name;
        }
        then accept;
    }
    term 2 {
        then reject;
    }
}
policy-statement export-policy-name {
    term 1 {
        from protocol [ bgp direct ];
        then {
            community add community-name;
            accept;
        }
    }
    term 2 {
        then reject;
    }
}
community community-name members target:target;
```

You can include these statements at the following hierarchy levels:

- [edit policy-options]
- [edit logical-systems *logical-system-name* policy-options]

7. Configure routing table groups to enable VPN route resolution in the **inet.3** routing table:

```
interface-routes {
    rib-group inet if-rib;
}
rib inet.3 {
    static {
        route BGP-address-for-remote-PE next-hop gre-interface-name;
    }
}
rib-groups {
    if-rib {
        import-rib [ inet.0 inet.3 ];
    }
}
```

You can include these statements at the following hierarchy levels:

- [edit routing-options]
- [edit logical-systems *logical-system-name* routing-options]

Configuring SCU and DCU for Layer 3 VPNs

For information on how to configure source class usage (SCU) for a Layer 3 VPN loopback interface, see the *JUNOS Network Management Configuration Guide*.

For information on how to configure SCU and destination class usage (DCU) to count packets on Layer 3 VPNs, see the *JUNOS Network Interfaces Configuration Guide*.

Protocol-Independent Load Balancing for Layer 3 VPNs

Protocol-independent load balancing for Layer 3 VPNs allows the forwarding next hops of both the active route and alternative paths to be used for load balancing. Protocol-independent load balancing works in conjunction with Layer 3 VPNs. It supports the load balancing of VPN routes independently of the assigned route distinguisher. When protocol-independent load balancing is enabled, both routes to other PE routers and routes to directly connected CE routers are load-balanced.

When load-balancing information is created for a given route, the active path is marked as Routing Use Only in the output of the show route table command.

The following sections describe how to configure protocol-independent load balancing and how this configuration can affect routing policies:

- Configuring Load Balancing for Layer 3 VPNs on page 181
- Configuring Load Balancing and Routing Policies on page 182

Configuring Load Balancing for Layer 3 VPNs

To configure protocol-independent load balancing for Layer 3 VPNs, include the multipath statement:

```
multipath {
   vpn-unequal-cost equal-external-internal;
}
```

If you include the multipath statement at the following hierarchy levels, protocol-independent load balancing is applied to the default routing table for that routing instance (*routing-instance-name*.inet.0):

- [edit routing-instances *routing-instance-name* routing-options]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options]

If you include the multipath statement at the following hierarchy levels, protocol-independent load balancing is applied to the specified routing table:

- [edit routing-instances *routing-instance-name* routing-options rib *routing-table-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options rib *routing-table-name*]

The vpn-unequal-cost statement is optional:

- If you do not configure the vpn-unequal-cost statement, protocol-independent load balancing is applied to VPN routes that are equal until the router identifier with regard to route selection.

- If you configure the vpn-unequal-cost statement, protocol-independent load balancing is applied to VPN routes that are equal until the IGP metric with regard to route selection.

The equal-external-internal statement is also optional. If you configure the equal-external-internal statement, protocol-independent load balancing is applied to both internal and external BGP paths.

Configuring Load Balancing and Routing Policies

If you enable protocol-independent load balancing for Layer 3 VPNs by including the multipath statement and if you also include the load-balance per-packet statement in the routing policy configuration, packets are not load-balanced.

For example, a PE router has the following VRF routing instance configured:

```
[edit routing-instances]
load-balance-example {
   instance-type vrf;
   interface fe-0/1/1.0;
   interface fe-0/1/1.1;
   route-distinguisher 2222:2;
   vrf-target target:2222:2;
   routing-options {
      multipath;
   }
   protocols {
      bgp {
         group group-example {
            import import-policy;
            family inet {
               unicast;
            }
            export export-policy;
            peer-as 4444;
            local-as 3333;
            multipath;
            as-override;
            neighbor 10.12.33.22;
         }
      }
   }
}
```

The PE router also has the following policy statement configured:

```
[edit policy-options policy-statement export-policy]
from protocol bgp;
then {
   load-balance per-packet;
}
```

When you include the multipath statement in the VRF routing instance configuration, the paths are no longer marked as BGP paths but are instead marked as multipath paths. Packets from the PE router are not load-balanced.

To ensure that VPN load-balancing functions as expected, do not include the from protocol statement in the policy statement configuration. The policy statement should be configured as follows:

```
[edit policy-options policy-statement export-policy]
then {
    load-balance per-packet;
}
```

For more information on how to configure per-packet load balancing, see the *JUNOS Policy Framework Configuration Guide*.

Configuring Layer 3 VPN Policing on Interfaces

You can use policing to control the amount of traffic flowing over the interfaces servicing a Layer 3 VPN. If policing is disabled on an interface, all the available bandwidth on a Layer 3 VPN tunnel can be used by a single CCC or TCC interface.

For more information about the policer statement, see the *JUNOS Policy Framework Configuration Guide*.

To enable Layer 3 VPN policing on an interface, include the policer statement:

```
policer {
    input policer-template-name;
    output policer-template-name;
}
```

If you configure CCC encapsulation, you can include the policer statement at the following hierarchy levels:

- [edit interfaces *interface-name* unit *logical-unit-number* family ccc]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family ccc]

If you configure TCC encapsulation, you can include the policer statement at the following hierarchy levels:

- [edit interfaces *interface-name* unit *logical-unit-number* family tcc]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family tcc]

Sending RADIUS Messages Through a Layer 3 VPN

You can send RADIUS messages through a Layer 3 VPN routing instance to customer RADIUS servers in a private network. To configure, include the routing-instance statement at the [edit access profile *profile-name* radius-server] hierarchy level and

apply the profile to an interface with the **access-profile** statement at the **[edit interfaces** *interface-name* **unit** *logical-unit-number* **ppp-options chap]** hierarchy level. For more information, see the *JUNOS System Basics Configuration Guide*.

Chapter 11

Troubleshooting Layer 3 VPNs

This chapter discusses the following strategies and tools for troubleshooting Layer 3 virtual private network (VPN) configurations:

Diagnosing Common Problems

When problems arise in a Layer 3 VPN configuration, the best way to troubleshoot is to start at one end of the VPN (the local customer edge [CE] router) and follow the routes to the other end of the VPN (the remote CE router). The following troubleshooting steps should help you diagnose common problems:

1. If you configured a routing protocol between the local provider edge (PE) and CE routers, check that the peering and adjacency are fully operational. When you do this, be sure to specify the name of the routing instance. For example, to check Open Shortest Path First (OSPF) adjacencies, enter the show ospf neighbor instance *routing-instance-name* command on the PE router.

   If the peering and adjacency are not fully operational, check the routing protocol configuration on the CE router and check the routing protocol configuration for the associated VPN routing instance on the PE router.

2. Check that the local CE and PE routers can ping each other.

   To check that the local CE router can ping the VPN interface on the local PE router, use a ping command in the following format, specifying the IP address or name of the PE router:

   user@host> **ping** (*ip-address* | *host-name*)

   To check that the local PE router can ping the CE router, use a ping command in the following format, specifying the IP address or name of the CE router, the name of the interface used for the VPN, and the source IP address (the local address) in outgoing ECHO_REQUEST packets:

   user@host> **ping** *ip-address* **interface** *interface* **local** *echo-address*

Often, the peering or adjacency between the local CE and local PE routers must come up before a ping command is successful. To check that a link is operational in a lab setting, remove the interface from the VPN routing and forwarding (VRF) by deleting the interface statement from the [edit routing-instance *routing-instance-name*] hierarchy level and recommitting the configuration. Doing this removes the interface from the VPN. Then try the ping command again. If the command is successful, configure the interface back into the VPN and check the routing protocol configuration on the local CE and PE routers again.

3. On the local PE router, check that the routes from the local CE router are in the VRF table (*routing-instance-name*.inet.0):

   user@host> **show route table** *routing-instance-name*.**inet.0 <detail>**

   The following example shows the routing table entries. Here, the loopback address of the CE router is 10.255.14.155/32 and the routing protocol between the PE and CE routers is BGP. The entry looks like any ordinary BGP announcement.

```
10.255.14.155/32 (1 entry, 1 announced)
        *BGP    Preference: 170/-101
                Nexthop: 192.168.197.141 via fe-1/0/0.0, selected
                State: <Active Ext>
                Peer AS:     1
                Age: 45:46
                Task: BGP_1.192.168.197.141+179
                Announcement bits (2): 0-BGP.0.0.0.0+179 1-KRT
                AS path: 1 I
                Localpref: 100
                Router ID: 10.255.14.155
```

   If the routes from the local CE router are not present in the VRF routing table, check that the CE router is advertising routes to the PE router. If static routing is used between the CE and PE routers, make sure the proper static routes are configured.

4. On a remote PE router, check that the routes from the local CE router are present in the bgp.l3vpn.0 routing table:

```
user@host> show route table bgp.l3vpn.0 extensive

10.255.14.175:3:10.255.14.155/32 (1 entry, 0 announced)
        *BGP    Preference: 170/-101
                Route Distinguisher: 10.255.14.175:3
                Source: 10.255.14.175
                Nexthop: 192.168.192.1 via fe-1/1/2.0, selected
                label-switched-path vpn07-vpn05
                Push 100004, Push 100005(top)
                State: <Active Int Ext>
                Local AS:    69 Peer AS:    69
                Age: 15:27      Metric2: 338
                Task: BGP_69.10.255.14.175+179
                AS path: 1 I
                Communities: target:69:100
                BGP next hop: 10.255.14.175
                Localpref: 100
```

```
Router ID: 10.255.14.175
Secondary tables: VPN-A.inet.0
```

The output of the show route table bgp.l3vpn.0 extensive command contains the following information specific to the VPN:

- In the prefix name (the first line of the output), the route distinguisher is added to the route prefix of the local CE router. Because the route distinguisher is unique within the Internet, the concatenation of the route distinguisher and IP prefix provides unique VPN-IP version 4 (IPv4) routing entries.
- The Route Distinguisher field lists the route distinguisher separately from the VPN-IPv4 address.
- The label-switched-path field shows the name of the label-switched path (LSP) used to carry the VPN traffic.
- The Push field shows both labels being carried in the VPN-IPv4 packet. The first label is the inner label, which is the VPN label that was assigned by the PE router. The second label is the outer label, which is a Resource Reservation Protocol (RSVP) label.
- The Communities field lists the target community.
- The Secondary tables field lists other routing tables on this router into which this route has been installed.

If routes from the local CE router are not present in the bgp.l3vpn.0 routing table on the remote PE router, do the following:

- Check the VRF import filter on the remote PE router, which is configured in the vrf-import statement. (On the local PE router, you check the VRF export filter, which is configured with the vrf-export statement.)
- Check that there is an operational LSP or a Label Distribution Protocol (LDP) path between the PE routers. To do this, check that the internal BGP (IBGP) next-hop addresses are in the inet.3 table.
- Check that the IBGP session between the PE routers is established and configured properly.
- Check for "hidden" routes, which usually means that routes were not labeled properly. To do this, use the show route table bgp.l3vpn.0 hidden command.
- Check that the inner label matches the inner VPN label that is assigned by the local PE router. To do this, use the show route table mpls command.

The following example shows the output of this command on the remote PE router. Here, the inner label is 100004.

```
...
Push 100004, Push 10005 (top)
```

The following example shows the output of this command on the local PE router, which shows that the inner label of 100004 matches the inner label on the remote PE router:

```
...
100004             *[VPN/7] 06:56:25, metric 1
 > to 192.168.197.141 via fe-1/0/0.0, Pop
```

5. On the remote PE router, check that the routes from the local CE router are present in the VRF table (*routing-instance-name*.inet.0):

```
user@host> show route table routing-instance-name.inet.0 detail

10.255.14.155/32 (1 entry, 1 announced)
        *BGP    Preference: 170/-101
                Route Distinguisher: 10.255.14.175:3
                Source: 10.255.14.175
                Nexthop: 192.168.192.1 via fe-1/1/2.0, selected
                label-switched-path vpn07-vpn05
                Push 100004, Push 100005(top)
                State: <Secondary Active Int Ext>
                Local AS:    69 Peer AS:    69
                Age: 1:16:22    Metric2: 338
                Task: BGP_69.10.255.14.175+179
                Announcement bits (2): 1-KRT 2-VPN-A-RIP
                AS path: 1 I
                Communities: target:69:100
                BGP next hop: 10.255.14.175
                Localpref: 100
                Router ID: 10.255.14.175
                Primary Routing Table bgp.l3vpn.0
```

In this routing table, the route distinguisher is no longer prepended to the prefix. The last line, Primary Routing Table, lists the table from which this route was learned.

If the routes are not present in this routing table, but were present in Step 4, the routes might have not passed the VRF import policy on the remote PE router.

If a VPN-IPv4 route matches no vrf-import policy, the route does not show up in the bgp.l3vpn table at all and hence is not present in the VRF table. If this occurs, it might indicate that on the PE router, you have configured another vrf-import statement on another VPN (with a common target), and the routes show up in the bgp.l3vpn.0 table, but are imported into the wrong VPN.

6. On the remote CE router, check that the routes from the local CE router are present in the routing table (inet.0):

```
user@host> show route
```

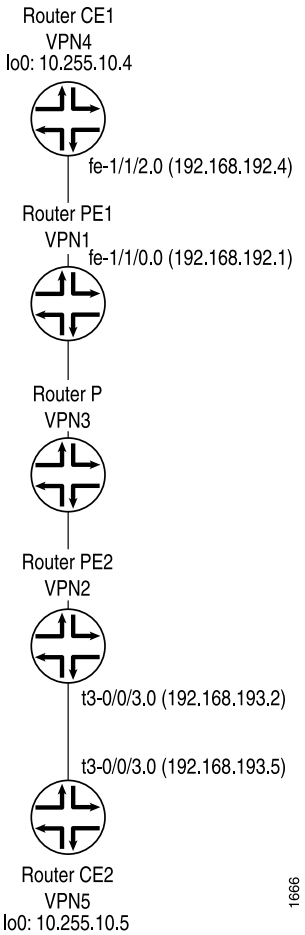
If the routes are not present, check the routing protocol configuration between the remote PE and CE routers, and make sure that peers and adjacencies (or static routes) between the PE and CE routers are correct.

7. If, in Step 1 through Step 6, you have determined that routes originated from the local CE router are correct, check the routes originated from the remote CE router by repeating Step 1 through Step 6.

Troubleshooting Layer 3 VPNs Using ping and traceroute

This section provides examples of how to use the ping command to check the accessibility of various routers in a VPN topology, and how to use the traceroute command to check the path that packets travel between the VPN routers. The topology shown in Figure 20 on page 189 illustrates these commands.

Figure 20: Layer 3 VPN Topology for ping and traceroute Examples



The following sections describe how to use the ping and traceroute commands to troubleshoot Layer 3 VPN topologies:

Pinging the CE Router from Another CE Router

You can ping one CE router from the other by specifying the other CE router's loopback address as the IP address in the ping command. This ping command succeeds if the loopback addresses have been announced by the CE routers to their directly connected PE routers. The success of these ping commands also means that Router CE1 can ping any network devices beyond Router CE2, and vice versa.
Figure 20 on page 189 shows the topology referenced in the following examples:

Pinging Router CE2 from Router CE1

Ping Router CE2 (VPN5) from Router CE1 (VPN4):

```
user@vpn4> ping 10.255.10.5 local 10.255.10.4 count 3
PING 10.255.10.5 (10.255.10.5): 56 data bytes
64 bytes from 10.255.10.5: icmp_seq=0 ttl=253 time=1.086 ms
64 bytes from 10.255.10.5: icmp_seq=1 ttl=253 time=0.998 ms
64 bytes from 10.255.10.5: icmp_seq=2 ttl=253 time=1.140 ms
--- 10.255.10.5 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.998/1.075/1.140/0.059 ms
```

Using traceroute from Loopback to Loopback

To determine the path from Router CE1's loopback interface to Router CE2's loopback interface, use the traceroute command:

```
user@vpn4> traceroute 10.255.10.5 source 10.255.10.4
traceroute to 10.255.10.5 (10.255.10.5) from 10.255.10.4, 30 hops max, 40 byte
packets
 1  vpn1-fe-110.isp-core.net (192.168.192.1)  0.680 ms  0.491 ms  0.456 ms
 2  vpn2-t3-001.isp-core.net (192.168.192.110)  0.857 ms  0.766 ms  0.754 ms
     MPLS Label=100005 CoS=0 TTL=1 S=1
 3  vpn5.isp-core.net (10.255.10.5)  0.825 ms  0.886 ms  0.732 ms
```

When you use the traceroute command to examine the path used by a Layer 3 VPN, the provider (P) routers in the service provider's network are not displayed. As shown above, the jump from Router VPN1 to Router VPN2 is displayed as a single hop. The P router (VPN3) shown in Figure 20 on page 189 is not displayed.

Pinging Router CE1 from Router CE2

Ping Router CE1 (VPN4) from Router CE2 (VPN5):

```
user@vpn5> ping 10.255.10.4 local 10.255.10.5 count 3
PING 10.255.10.4 (10.255.10.4): 56 data bytes
64 bytes from 10.255.10.4: icmp_seq=0 ttl=253 time=1.042 ms
64 bytes from 10.255.10.4: icmp_seq=1 ttl=253 time=0.998 ms
64 bytes from 10.255.10.4: icmp_seq=2 ttl=253 time=0.954 ms
--- 10.255.10.4 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.954/0.998/1.042/0.036 ms
```

Using traceroute from Router CE2 to Router CE1

To determine the path from Router CE2 to Router CE1, use the traceroute command:

```
user@vpn5> traceroute 10.255.10.4 source 10.255.10.5
traceroute to 10.255.10.4 (10.255.10.4) from 10.255.10.5, 30 hops max, 40 byte
packets
 1  vpn-08-t3-003.isp-core.net (192.168.193.2)  0.686 ms  0.519 ms  0.548 ms
 2  vpn1-so-100.isp-core.net (192.168.192.100)  0.918 ms  0.869 ms  0.859 ms
     MPLS Label=100021 CoS=0 TTL=1 S=1
 3  vpn4.isp-core.net (10.255.10.4)  0.878 ms  0.760 ms  0.739 ms
```

Pinging the Remote PE and CE Routers from the Local CE Router

From the local CE router, you can ping the VPN interfaces on the remote PE and CE routers, which are point-to-point interfaces. Figure 20 on page 189 shows the topology referenced in the following examples:

- Pinging Router CE2 from Router CE1 on page 191
- Using traceroute from Router CE1 to Router CE2 on page 192
- Pinging Router PE2 from Router CE1 on page 192
- Using traceroute from Router CE1 to Router PE2 on page 192
- Pinging a CE Router from a Multiaccess Interface on page 192

Pinging Router CE2 from Router CE1

Ping Router CE2 (VPN5) from Router CE1 (VPN4):

```
user@vpn4> ping 192.168.193.5 local 10.255.10.4 count 3
PING 192.168.193.5 (192.168.193.5): 56 data bytes
64 bytes from 192.168.193.5: icmp_seq=0 ttl=253 time=1.040 ms
64 bytes from 192.168.193.5: icmp_seq=1 ttl=253 time=0.891 ms
```

```
64 bytes from 192.168.193.5: icmp_seq=2 ttl=253 time=0.944 ms
--- 192.168.193.5 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.891/0.958/1.040/0.062 ms
```

Using traceroute from Router CE1 to Router CE2

To determine the path from Router CE1's loopback interface to Router CE2's directly connected interface, use the traceroute command:

```
user@vpn4> traceroute 192.168.193.5 source 10.255.10.4
traceroute to 192.168.193.5 (192.168.193.5) from 10.255.10.4, 30 hops max, 40
byte packets
 1  vpn1-fe-110.isp-core.net (192.168.192.1)  0.669 ms  0.508 ms  0.457 ms
 2  vpn2-t3-001.isp-core.net (192.168.192.110)  0.851 ms  0.769 ms  0.750 ms
     MPLS Label=100000 CoS=0 TTL=1 S=1
 3  vpn5-t3-003.isp-core.net (192.168.193.5)  0.829 ms  0.838 ms  0.731 ms
```

Pinging Router PE2 from Router CE1

Ping Router PE2 (VPN2) from Router CE1 (VPN4). In this case, packets that originate at Router CE1 go to Router PE2, then to Router CE2, and back to Router PE2 before Router PE2 can respond to Internet Control Message Protocol (ICMP) requests. You can verify this by using the traceroute command.

```
user@vpn4> ping 192.168.193.2 local 10.255.10.4 count 3
PING 192.168.193.2 (192.168.193.2): 56 data bytes
64 bytes from 192.168.193.2: icmp_seq=0 ttl=254 time=1.080 ms
64 bytes from 192.168.193.2: icmp_seq=1 ttl=254 time=0.967 ms
64 bytes from 192.168.193.2: icmp_seq=2 ttl=254 time=0.983 ms
--- 192.168.193.2 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.967/1.010/1.080/0.050 ms
```

Using traceroute from Router CE1 to Router PE2

To determine the path from Router CE1 to Router PE2, use the traceroute command:

```
user@vpn4> traceroute 192.168.193.2 source 10.255.10.4
traceroute to 192.168.193.2 (192.168.193.2) from 10.255.10.4, 30 hops max, 40
byte packets
 1  vpn1-fe-110.isp-core.net (192.168.192.1)  0.690 ms  0.490 ms  0.458 ms
 2  vpn2-t3-003.isp-core.net (192.168.193.2)  0.846 ms  0.768 ms  0.749 ms
     MPLS Label=100000 CoS=0 TTL=1 S=1
 3  vpn5-t3-003.isp-core.net (192.168.193.5)  0.643 ms  0.703 ms  0.600 ms
 4  vpn-08-t3-003.isp-core.net (192.168.193.2)  0.810 ms  0.739 ms  0.729 ms
```

Pinging a CE Router from a Multiaccess Interface

You cannot ping one CE router from the other if the VPN interface is a multiaccess interface, such as the fe-1/1/2.0 interface on Router CE1. To ping Router CE1 from Router CE2, you must either configure the vrf-table-label statement at the [edit routing-instances *routing-instance-name*] hierarchy level on Router PE1 or configure a

static route on Router PE1 to the VPN interface of Router CE1. If you configure the vrf-table-label statement to ping a router, you cannot configure a static route.

If you configure a static route on Router PE1 to the VPN interface of Router CE1, its next hop must point to Router CE1 (at the [edit routing-instance *routing-instance-name*] hierarchy level), and this route must be announced from Router PE1 to Router PE2 as shown in the following configuration:

```
[edit]
routing-instances {
    direct-multipoint {
        instance-type vrf;
        interface fe-1/1/0.0;
        route-distinguisher 69:1;
        vrf-import direct-import;
        vrf-export direct-export;
        routing-options {
            static {
                route 192.168.192.4/32 next-hop 192.168.192.4;
            }
        }
        protocols {
            bgp {
                group to-vpn4 {
                    peer-as 1;
                    neighbor 192.168.192.4;
                }
            }
        }
    }
    policy-options {
        policy-statement direct-export {
            term a {
                from protocol bgp;
                then {
                    community add direct-comm;
                    accept;
                }
            }
            term b {
                from {
                    protocol static;
                    route-filter 192.168.192.4/32 exact;
                }
                then {
                    community add direct-comm;
                    accept;
                }
            }
            term d {
                then reject;
            }
        }
    }
}
```

Now you can ping Router CE1 from Router CE2:

```
user@vpn5> ping 192.168.192.4 local 10.255.10.5 count 3
PING 192.168.192.4 (192.168.192.4): 56 data bytes
64 bytes from 192.168.192.4: icmp_seq=0 ttl=253 time=1.092 ms
64 bytes from 192.168.192.4: icmp_seq=1 ttl=253 time=1.019 ms
64 bytes from 192.168.192.4: icmp_seq=2 ttl=253 time=1.031 ms
--- 192.168.192.4 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.019/1.047/1.092/0.032 ms
```

To determine the path between these two interfaces, use the traceroute command:

```
user@vpn5> traceroute 192.168.192.4 source 10.255.10.5
traceroute to 192.168.192.4 (192.168.192.4) from 10.255.10.5, 30 hops max, 40
byte packets
 1  vpn-08-t3003.isp-core.net (192.168.193.2)  0.678 ms  0.549 ms  0.494 ms
 2  vpn1-so-100.isp-core.net (192.168.192.100)  0.873 ms  0.847 ms  0.844 ms
     MPLS Label=100021 CoS=0 TTL=1 S=1
 3  vpn4-fe-112.isp-core.net (192.168.192.4)  0.825 ms  0.743 ms  0.764 ms
```

Pinging the Directly Connected PE Routers from the CE Routers

From the loopback interfaces on the CE routers, you can ping the VPN interface on the directly connected PE router. Figure 20 on page 189 shows the topology referenced in the following examples:

- Pinging Router PE1 from the Loopback Interface on Router CE1 on page 194
- Using traceroute from the Loopback Interface on Router CE1 to PE1 on page 194
- Pinging Router PE2 from the Loopback Interface on Router CE2 on page 195
- Using traceroute from the Loopback Interface on Router CE2 to PE2 on page 195

Pinging Router PE1 from the Loopback Interface on Router CE1

From the loopback interface on Router CE1 (VPN4), ping the VPN interface, fe-1/1/0.0, on Router PE1:

```
user@vpn4> ping 192.168.192.1 local 10.255.10.4 count 3
PING 192.168.192.1 (192.168.192.1): 56 data bytes
64 bytes from 192.168.192.1: icmp_seq=0 ttl=255 time=0.885 ms
64 bytes from 192.168.192.1: icmp_seq=1 ttl=255 time=0.757 ms
64 bytes from 192.168.192.1: icmp_seq=2 ttl=255 time=0.734 ms
--- 192.168.192.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.734/0.792/0.885/0.066 ms
```

Using traceroute from the Loopback Interface on Router CE1 to PE1

To determine the path from the loopback interface on Router CE1 to the VPN interfaces on Router PE1, use the traceroute command:

```
user@vpn4> traceroute 192.168.192.1 source 10.255.10.4
```

```
traceroute to 192.168.192.1 (192.168.192.1) from 10.255.10.4, 30 hops max, 40
byte packets
 1  vpn1-fe-110.isp-core.net (192.168.192.1)  0.828 ms  0.657 ms  1.972 ms
```

Pinging Router PE2 from the Loopback Interface on Router CE2

From the loopback interface on Router CE2 (VPN5), ping the VPN interface, t3-0/0/3.0, on Router PE2:

```
user@vpn5> ping 192.168.193.2 local 10.255.10.5 count 3
PING 192.168.193.2 (192.168.193.2): 56 data bytes
64 bytes from 192.168.193.2: icmp_seq=0 ttl=255 time=0.998 ms
64 bytes from 192.168.193.2: icmp_seq=1 ttl=255 time=0.834 ms
64 bytes from 192.168.193.2: icmp_seq=2 ttl=255 time=0.819 ms
--- 192.168.193.2 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.819/0.884/0.998/0.081 ms
```

Using traceroute from the Loopback Interface on Router CE2 to PE2

To determine the path from the loopback interface on Router CE2 to the VPN interfaces on Router PE2, use the traceroute command:

```
user@vpn5> traceroute 192.168.193.2 source 10.255.10.5
traceroute to 192.168.193.2 (192.168.193.2) from 10.255.10.5, 30 hops max, 40
byte packets
 1  vpn-08-t3003.isp-core.net (192.168.193.2)  0.852 ms  0.670 ms  0.656 ms
```

Pinging the Directly Connected CE Routers from the PE Routers

From the VPN and loopback interfaces on the PE routers, you can ping the VPN interface on the directly connected CE router. Figure 20 on page 189 shows the topology referenced in the following examples:

- Pinging the VPN Interface on Router CE1 from Router PE1 on page 195
- Pinging the Loopback Interface on Router CE1 from Router PE1 on page 196
- Using traceroute from Router PE1 to Router CE1 on page 196
- Pinging the VPN Interface on Router CE2 from Router PE2 on page 196
- Pinging the Loopback Interface on Router CE2 from Router PE2 on page 197
- Using traceroute from Router PE2 to Router CE2 on page 197

Pinging the VPN Interface on Router CE1 from Router PE1

From the VPN interface on the PE router, you can ping the VPN or loopback interface on the directly connected CE router.

From the VPN interface on Router PE1 (VPN1), ping the VPN interface, fe-1/1/0.0, on Router CE1:

```
user@vpn1> ping 192.168.192.4 interface fe-1/1/0.0 local 192.168.192.1 count 3
```

```
PING 192.168.192.4 (192.168.192.4): 56 data bytes
64 bytes from 192.168.192.4: icmp_seq=0 ttl=255 time=0.866 ms
64 bytes from 192.168.192.4: icmp_seq=1 ttl=255 time=0.728 ms
64 bytes from 192.168.192.4: icmp_seq=2 ttl=255 time=0.753 ms
--- 192.168.192.4 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.728/0.782/0.866/0.060 ms
```

Pinging the Loopback Interface on Router CE1 from Router PE1

From the VPN interface on Router PE1 (VPN1), ping the loopback interface, 10.255.10.4, on Router CE1:

```
user@vpn1> ping 10.255.10.4 interface fe-1/1/0.0 local 192.168.192.1 count 3
PING 10.255.10.4 (10.255.10.4): 56 data bytes
64 bytes from 10.255.10.4: icmp_seq=0 ttl=255 time=0.838 ms
64 bytes from 10.255.10.4: icmp_seq=1 ttl=255 time=0.760 ms
64 bytes from 10.255.10.4: icmp_seq=2 ttl=255 time=0.771 ms
--- 10.255.10.4 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.760/0.790/0.838/0.034 ms
```

Using traceroute from Router PE1 to Router CE1

To determine the path from the VPN interface on Router PE1 to the VPN and loopback interfaces on Router CE1, respectively, use the following traceroute commands:

```
user@vpn1> traceroute 10.255.10.4 interface fe-1/1/0.0 source 192.168.192.1
traceroute to 10.255.10.4 (10.255.10.4) from 192.168.192.1, 30 hops max, 40 byte
 packets
 1  vpn4.isp-core.net (10.255.10.4)  0.842 ms  0.659 ms  0.621 ms
user@vpn1>  traceroute 192.168.192.4 interface fe-1/1/0.0 source 192.168.192.1

traceroute to 192.168.192.4 (192.168.192.4) from 192.168.192.1, 30 hops max, 40
byte packets
 1  vpn4-fe-112.isp-core.net (192.168.192.4)  0.810 ms  0.662 ms  0.640 ms
```

Pinging the VPN Interface on Router CE2 from Router PE2

From the VPN interface on Router PE2 (VPN2), ping the VPN interface, t3-0/0/3.0, on Router CE2:

```
user@vpn2> ping 192.168.193.5 interface t3-0/0/3.0 local 192.168.193.2 count 3
PING 192.168.193.5 (192.168.193.5): 56 data bytes
64 bytes from 192.168.193.5: icmp_seq=0 ttl=255 time=0.852 ms
64 bytes from 192.168.193.5: icmp_seq=1 ttl=255 time=0.909 ms
64 bytes from 192.168.193.5: icmp_seq=2 ttl=255 time=0.793 ms
--- 192.168.193.5 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.793/0.851/0.909/0.047 ms
```

Pinging the Loopback Interface on Router CE2 from Router PE2

From the VPN interface on Router PE2 (VPN2), ping the loopback interface, 10.255.10.5, on Router CE2:

```
user@vpn2> ping 10.255.10.5 interface t3-0/0/3.0 local 192.168.193.2 count 3
PING 10.255.10.5 (10.255.10.5): 56 data bytes
64 bytes from 10.255.10.5: icmp_seq=0 ttl=255 time=0.914 ms
64 bytes from 10.255.10.5: icmp_seq=1 ttl=255 time=0.888 ms
64 bytes from 10.255.10.5: icmp_seq=2 ttl=255 time=1.066 ms
--- 10.255.10.5 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.888/0.956/1.066/0.079 ms
```

Using traceroute from Router PE2 to Router CE2

To determine the path from the VPN interface on Router PE2 to the VPN and loopback interfaces on Router CE2, respectively, use the following traceroute commands:

```
user@vpn2> traceroute 10.255.10.5 interface t3-0/0/3.0 source 192.168.193.2
traceroute to 10.255.10.5 (10.255.10.5) from 192.168.193.2, 30 hops max, 40 byte
 packets
 1  vpn5.isp-core.net (10.255.10.5)  1.009 ms  0.677 ms  0.633 ms
user@vpn2>  traceroute 192.168.193.5 interface t3-0/0/3.0 source 192.168.193.2
traceroute to 192.168.193.5 (192.168.193.5) from 192.168.193.2, 30 hops max, 40
byte packets
 1  vpn5-t3-003.isp-core.net (192.168.193.5)  0.974 ms  0.665 ms  0.619 ms
```

Pinging the Remote CE Router from the Local PE Router

The following procedure is effective for Layer 3 VPNs only. To ping a remote CE router from a local PE router in a Layer 3 VPN, you need to configure the following:

1. Configure a logical unit for the loopback interface.

   To configure an additional logical unit on the loopback interface of the PE router, configure the unit statement at the [edit interfaces lo0] hierarchy level:

   ```
   [edit interfaces]
   lo0 {
      unit number {
         family inet {
            address address;
         }
      }
   }
   ```

2. Configure the loopback interface for the Layer 3 VPN routing instance on the local PE router. You can associate one logical loopback interface with each Layer 3 VPN routing instance, enabling you to ping a specific routing instance on a router.

   Specify the loopback interface you configured in Step 1 using the interface statement at the [edit routing-instances *routing-instance-name*] hierarchy level:

   ```
   [edit routing-instances routing-instance-name]
   ```

interface *interface-name*;

The *interface-name* is the logical unit on the loopback interface (for example, lo0.1).

3. From the VPN interface on PE router, you can now ping the logical unit on the loopback interface on the remote CE router:

   user@host> **ping interface *interface host***

   Use *interface* to specify the new logical unit on the loopback interface (for example, lo0.1). For more information on how to use the ping interface command, see the *JUNOS Interfaces Command Reference*.

Limitation on Pinging a Remote CE Router from a PE Router

If you attempt to ping a remote CE router from a PE router, ICMP echo requests are sent from the PE router, with the PE router's VPN interface as the source. Other PE routers have a route back to that address with a VPN label. When the echo replies return, they include a label. The PE router pops the VPN label and sends the packet from the VPN interface to the local CE router. The local CE router sends it back to the PE router, its actual destination.

When a Juniper Networks routing platform receives a labeled packet, the label is popped (depending on the label operation specified), and the packet is forwarded to an interface, even if the packet is destined for that particular PE router. Labeled packets are not analyzed further for the IP information under the label.

If there is a problem with the connection to the local CE router, packets are sent out but do not return to the PE router, and the ping fails. If the connection between your PE router and local CE router is down, sending a ping to the remote CE router fails even though the connection to the remote CE router might be functional.

Pinging a Layer 3 VPN

You can ping from a PE router to a PE router in a Layer 3 VPN using the ping mpls l3vpn *l3vpn-name* prefix *prefix* <count *count*> command. For more information, see "Pinging VPNs and Layer 2 Circuits" on page 37.

For a detailed description of the ping mpls command, see the *JUNOS Routing Protocols and Policies Command Reference*.

Disabling Normal TTL Decrementing for Layer 3 VPNs

For information on how to disable normal TTL decrementing for Layer 3 VPNs, see "Disabling Normal TTL Decrementing for VPNs" on page 82.

Troubleshooting RSVP and LDP LSPs

You can use the show mpls lsp command to determine whether an LSP is up and running. However, this command displays information on RSVP LSPs only. If you

have configured LDP LSPs, use the show route protocol ldp command. For more information on how to use show commands to troubleshoot RSVP LSPs, see the *JUNOS MPLS Network Operations Guide*.

Troubleshooting Inconsistently Advertised Routes from Gigabit Ethernet Interfaces

For direct routes on a LAN in a VRF, the JUNOS software attempts to locate a CE that can be designated as the next hop. If this cannot be done, advertised routes from Gigabit Ethernet interfaces are dropped.

In such instances, do one of the following:

- Use the static statement at the [edit routing-options] or [edit logical-systems *logical-system-name* routing-options] hierarchy levels in the VRF routing instance to a CE router on the LAN subnet, configuring the CE as the next hop. All traffic to directly destinations on this LAN will go to the CE. You can add two static routes to two CEs on the LAN for redundancy.
- Configure the vrf-table-label statement at the [edit routing-instances *routing-instance-name*] hierarchy levels to map the inner label of a packet to a specific VRF routing table. This allows the examination of the encapsulated IP header to force IP lookups on the VRF routing instance for all traffic.

NOTE: The vrf-table-label statement is not available for every core-facing interface; for example, channelized interfaces are not supported. See "Support for Ethernet, SONET/SDH, and T1/T3/E3 Interfaces" on page 163 for information about support for the vrf-table-label statement over Ethernet and SONET/SDH interfaces.

Chapter 12

Layer 3 VPN Configuration Examples

The examples in this chapter show only the portions of the configuration that establish VPN functionality. You must also configure other router functionality, including all router interfaces, for a router configuration to work properly.

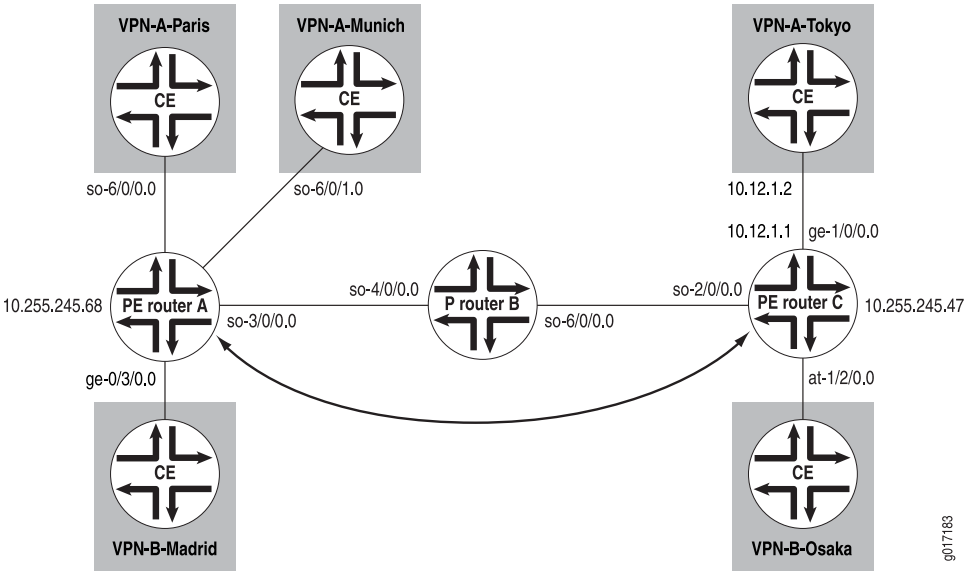
This chapter provides the following examples of Layer 3 virtual private network (VPN) configurations:

Configuring a Simple Full-Mesh VPN Topology

This example shows how to set up a simple full-mesh service provider VPN configuration, which consists of the following components (see Figure 21 on page 202):

- Two separate VPNs (VPN-A and VPN-B)
- Two provider edge (PE) routers, both of which service VPN-A and VPN-B
- Resource Reservation Protocol (RSVP) as the signaling protocol
- One RSVP label-switched path (LSP) that tunnels between the two PE routers through one provider (P) router

Figure 21: Example of a Simple VPN Topology



In this configuration, route distribution in VPN A from Router VPN-A-Paris to Router VPN-A-Tokyo occurs as follows:

1. The customer edge (CE) router VPN-A-Paris announces routes to the PE router Router A.
2. Router A installs the received announced routes into its VPN routing and forwarding (VRF) table, VPN-A.inet.0.
3. Router A creates a Multiprotocol Label Switching (MPLS) label for the interface between it and Router VPN-A-Paris.
4. Router A checks its VRF export policy.
5. Router A converts the Internet Protocol version 4 (IPv4) routes from Router VPN-A-Paris into VPN IPv4 format using its route distinguisher and announces these routes to PE Router C over the internal BGP (IBGP) between the two PE routers.
6. Router C checks its VRF import policy and installs all routes that match the policy into its bgp.l3vpn.0 routing table. (Any routes that do not match are discarded.)
7. Router C checks its VRF import policy and installs all routes that match into its VPN-A.inet.0 routing table. The routes are installed in IPv4 format.
8. Router C announces its routes to the CE router Router VPN-A-Tokyo, which installs them into its master routing table. (For routing platforms running JUNOS software, the master routing table is inet.0.)
9. Router C uses the LSP between it and Router A to route all packets from Router VPN-A-Tokyo that are destined for Router VPN-A-Paris.

The final section in this example, “Simple VPN Configuration Summarized by Router” on page 211, consolidates the statements needed to configure VPN functionality on each of the service P routers shown in Figure 21 on page 202.

NOTE: In this example, a private autonomous system (AS) number is used for the route distinguisher and the route target. This number is used for illustration only. When you are configuring VPNs, you should use an assigned AS number.

The following sections explain how to configure the VPN functionality on the PE and P routers. The CE routers have no information about the VPN, so you configure them normally.

Enabling an IGP on the PE and P Routers

To allow the PE and P routers to exchange routing information among themselves, you must configure an interior gateway protocol (IGP) on all these routers or you must configure static routes. You configure the IGP on the master instance of the routing protocol process (rpd) (that is, at the [edit protocols] hierarchy level), not within the VPN routing instance (that is, not at the [edit routing-instances] hierarchy level).

You configure the IGP in the standard way. This configuration example does not include this portion of the configuration.

Enabling RSVP and MPLS on the P Router

On the P router, Router B, you must configure RSVP and MPLS because this router exists on the MPLS LSP path between the two PE routers, Router A and Router C:

```
[edit]
protocols {
    rsvp {
        interface so-4/0/0.0;
        interface so-6/0/0.0;
    }
    mpls {
        interface so-4/0/0.0;
        interface so-6/0/0.0;
    }
}
```

Configuring the MPLS LSP Tunnel Between the PE Routers

In this configuration example, RSVP is used for VPN signaling. Therefore, in addition to configuring RSVP, you must enable traffic engineering support in an IGP and you must create an MPLS LSP to tunnel the VPN traffic.

On PE Router A, enable RSVP and configure one end of the MPLS LSP tunnel. In this example, traffic engineering support is enabled for Open Shortest Path First (OSPF). When configuring the MPLS LSP, include interface statements for all interfaces participating in MPLS, including the interfaces to the PE and CE routers. The statements for the interfaces between the PE and CE routers are needed so that the PE router can create an MPLS label for the private interface. In this example, the first interface statement configures MPLS on the interface connected to the LSP, and the remaining three configure MPLS on the interfaces that connect the PE router to the CE routers.

```
[edit]
protocols {
    rsvp {
        interface so-3/0/0.0;
    }
    mpls {
        label-switched-path RouterA-to-RouterC {
            to 10.255.245.47;
        }
        interface so-3/0/0.0;
        interface so-6/0/0.0;
        interface so-6/0/1.0;
        interface ge-0/3/0.0;
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface so-3/0/0.0;
        }
    }
}
```

On PE Router C, enable RSVP and configure the other end of the MPLS LSP tunnel. Again, traffic engineering support is enabled for OSPF, and you configure MPLS on the interfaces to the LSP and the CE routers.

```
[edit]
protocols {
    rsvp {
        interface so-2/0/0.0;
    }
    mpls {
        label-switched-path RouterC-to-RouterA {
            to 10.255.245.68;
        }
        interface so-2/0/0.0;
        interface ge-1/0/0.0;
        interface at-1/2/0.0;
    }
```

```
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface so-2/0/0.0;
        }
    }
}
```

Configuring IBGP on the PE Routers

On the PE routers, configure an IBGP session with the following properties:

- VPN family—To indicate that the IBGP session is for the VPN, include the **family inet-vpn** statement.
- Loopback address—Include the **local-address** statement, specifying the local PE router's loopback address. The IBGP session for VPNs runs through the loopback address. You must also configure the **lo0** interface at the **[edit interfaces]** hierarchy level. The example does not include this part of the router's configuration.
- Neighbor address—Include the **neighbor** statement, specifying the IP address of the neighboring PE router, which is its loopback (**lo0**) address.

On PE Router A, configure IBGP:

```
[edit]
protocols {
    bgp {
        group PE-RouterA-to-PE-RouterC {
            type internal;
            local-address 10.255.245.68;
            family inet-vpn {
                unicast;
            }
            neighbor 10.255.245.47;
        }
    }
}
```

On PE Router C, configure IBGP:

```
[edit]
protocols {
    bgp {
        group PE-RouterC-to-PE-RouterA {
            type internal;
            local-address 10.255.245.47;
            family inet-vpn {
                unicast;
            }
            neighbor 10.255.245.68;
        }
    }
}
```

Configuring Routing Instances for VPNs on the PE Routers

Both PE routers service VPN-A and VPN-B, so you must configure two routing instances on each router, one for each VPN. For each VPN, you must define the following in the routing instance:

- Route distinguisher, which must be unique for each routing instance on the PE router.
- It is used to distinguish the addresses in one VPN from those in another VPN.
- Instance type of vrf, which creates the VRF table on the PE router.
- Interfaces connected to the CE routers.
- VRF import and export policies, which must be the same on each PE router that services the same VPN. Unless an import policy contains only a then reject statement, it must include reference to a community. Otherwise, when you try to commit the configuration, the commit fails.

NOTE: In this example, a private AS number is used for the route distinguisher. This number is used for illustration only. When you are configuring VPNs, you should use an assigned AS number.

- Routing between the PE and CE routers, which is required for the PE router to distribute VPN-related routes to and from connected CE routers. You can configure a routing protocol—BGP, OSPF, or Routing Information Protocol (RIP)—or you can configure static routing.

On PE Router A, configure the following routing instance for VPN-A. In this example, Router A uses static routes to distribute routes to and from the two CE routers to which it is connected.

```
[edit]
routing-instance {
   VPN-A-Paris-Munich {
      instance-type vrf;
      interface so-6/0/0.0;
      interface so-6/0/1.0;
      route-distinguisher 65535:0;
      vrf-import VPN-A-import;
      vrf-export VPN-A-export;
      routing-options {
         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;
         }
      }
   }
}
```

On PE Router C, configure the following routing instance for VPN-A. In this example, Router C uses BGP to distribute routes to and from the CE router to which it is connected.

```
[edit]
routing-instance {
   VPN-A-Tokyo {
      instance-type vrf;
      interface ge-1/0/0.0;
      route-distinguisher 65535:1;
      vrf-import VPN-A-import;
      vrf-export VPN-A-export;
      protocols {
         bgp {
            group VPN-A-Site2 {
               peer-as 1;
               neighbor 10.12.1.2;
            }
         }
      }
   }
}
```

On PE Router A, configure the following routing instance for VPN-B. In this example, Router A uses OSPF to distribute routes to and from the CE router to which it is connected.

```
[edit]
routing-instance {
   VPN-B-Madrid {
      instance-type vrf;
      interface ge-0/3/0.0;
      route-distinguisher 65535:2;
      vrf-import VPN-B-import;
      vrf-export VPN-B-export;
      protocols {
         ospf {
            export bgp-to-ospf;
            area 0.0.0.0 {
               interface ge-0/3/0;
            }
         }
      }
   }
}
```

On PE Router C, configure the following routing instance for VPN-B. In this example, Router C uses RIP to distribute routes to and from the CE router to which it is connected.

```
[edit]
routing-instance {
   VPN-B-Osaka {
      instance-type vrf;
      interface at-1/2/0.0;
      route-distinguisher 65535:3;
      vrf-import VPN-B-import;
      vrf-export VPN-B-export;
      protocols {
         rip {
```

```
            group PE-C-to-VPN-B {
               export bgp-to-rip;
               neighbor at-1/2/0;
            }
         }
      }
   }
}
```

Configuring VPN Policy on the PE Routers

Configure the VPN import and export policies on each PE router so that the appropriate routes are installed in the PE router's VRF tables. The VRF table is used to forward packets within a VPN. For VPN-A, the VRF table is VPN-A.inet.0, and for VPN-B it is VPN-B.inet.0.

In the VPN policy, you also configure VPN target communities.

NOTE: In this example, a private AS number is used for the route target. This number is used for illustration only. When you are configuring VPNs, you should use an assigned AS number.

On PE Router A, configure the following VPN import and export policies:

NOTE: The policy qualifiers shown in this example are only those needed for the VPN to function. You can configure additional qualifiers, as needed, to any policies that you configure.

```
[edit]
policy-options {
   policy-statement VPN-A-import {
      term a {
         from {
            protocol bgp;
            community VPN-A;
         }
         then accept;
      }
      term b {
         then reject;
      }
   }
   policy-statement VPN-A-export {
      term a {
         from protocol static;
         then {
            community add VPN-A;
            accept;
         }
      }
```

```
        term b {
            then reject;
        }
    }
    policy-statement VPN-B-import {
        term a {
            from {
                protocol bgp;
                community VPN-B;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    policy-statement VPN-B-export {
        term a {
            from protocol ospf;
            then {
                community add VPN-B;
                accept;
            }
        }
        term b {
            then reject;
        }
    }
    community VPN-A members target:65535:4;
    community VPN-B members target:65535:5;
}
```

On PE Router C, configure the following VPN import and export policies:

```
[edit]
policy-options {
    policy-statement VPN-A-import {
        term a {
            from {
                protocol bgp;
                community VPN-A;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    policy-statement VPN-A-export {
        term a {
            from protocol bgp;
            then {
                community add VPN-A;
                accept;
            }
        }
```

```
        term b {
            then reject;
        }
    }
    policy-statement VPN-B-import {
        term a {
            from {
                protocol bgp;
                community VPN-B;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    policy-statement VPN-B-export {
        term a {
            from protocol rip;
            then {
                community add VPN-B;
                accept;
            }
        }
        term b {
            then reject;
        }
    }
    community VPN-A members target:65535:4;
    community VPN-B members target:65535:5;
}
```

To apply the VPN policies on the routers, include the vrf-export and vrf-import statements when you configure the routing instance. For both VPNs, the VRF import and export policies handle the route distribution across the IBGP session running between the PE routers.

To apply the VPN policies on PE Router A, include the following statements:

```
[edit]
routing-instance {
    VPN-A-Paris-Munich {
        vrf-import VPN-A-import;
        vrf-export VPN-A-export;
    }
    VPN-B-Madrid {
        vrf-import VPN-B-import;
        vrf-export VPN-B-export;
    }
}
```

To apply the VPN policies on PE Router C, include the following statements:

```
[edit]
routing-instance {
    VPN-A-Tokyo {
```

```
        vrf-import VPN-A-import;
        vrf-export VPN-A-export;
    }
    VPN-B-Osaka {
        vrf-import VPN-B-import;
        vrf-export VPN-B-export;
    }
}
```

Simple VPN Configuration Summarized by Router

Router A (PE Router)

Routing Instance for VPN-A

```
routing-instance {
    VPN-A-Paris-Munich {
        instance-type vrf;
        interface so-6/0/0.0;
        interface so-6/0/1.0;
        route-distinguisher 65535:0;
        vrf-import VPN-A-import;
        vrf-export VPN-A-export;
    }
}
```

Instance Routing Protocol

```
routing-options {
    static {
        route 172.16.0.0/16 next-hop so-6/0/0.0;
        route 172.17.0.0/16 next-hop so-6/0/1.0;
    }
}
```

Routing Instance for VPN-B

```
routing-instance {
    VPN-B-Madrid {
        instance-type vrf;
        interface ge-0/3/0.0;
        route-distinguisher 65535:2;
        vrf-import VPN-B-import;
        vrf-export VPN-B-export;
    }
}
```

Instance Routing Protocol

```
protocols {
    ospf {
        area 0.0.0.0 {
            interface ge-0/3/0;
        }
    }
}
```

Master Protocol Instance

```
protocols {
}
```

Enable RSVP

```
rsvp {
```

```
    interface so-3/0/0.0;
}
```

Configure an MPLS LSP

```
mpls {
    label-switched-path RouterA-to-RouterC {
        to 10.255.245.47;
    }
    interface so-3/0/0.0;
    interface so-6/0/0.0;
    interface so-6/0/1.0;
    interface ge-0/3/0.0;
}
```

Configure IBGP

```
bgp {
    group PE-RouterA-to-PE-RouterC {
        type internal;
        local-address 10.255.245.68;
        family inet-vpn {
            unicast;
        }
        neighbor 10.255.245.47;
    }
}
```

Configure OSPF for Traffic Engineering Support

```
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface so-3/0/0.0;
    }
}
```

Configure VPN Policy

```
policy-options {
    policy-statement VPN-A-import {
        term a {
            from {
                protocol bgp;
                community VPN-A;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    policy-statement VPN-A-export {
        term a {
            from protocol static;
            then {
                community add VPN-A;
                accept;
            }
        }
        term b {
            then reject;
```

```
        }
    }
    policy-statement VPN-B-import {
        term a {
            from {
                protocol bgp;
                community VPN-B;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    policy-statement VPN-B-export {
        term a {
            from protocol ospf;
            then {
                community add VPN-B;
                accept;
            }
        }
        term b {
            then reject;
        }
    }
    community VPN-A members target:65535:4;
    community VPN-B members target:65535:5;
}
```

Router B (P Router)

Master Protocol Instance

```
protocols {
}
```

Enable RSVP

```
rsvp {
    interface so-4/0/0.0;
    interface so-6/0/0.0;
}
```

Enable MPLS

```
mpls {
    interface so-4/0/0.0;
    interface so-6/0/0.0;
}
```

Router C (PE Router)

Routing Instance for VPN-A

```
routing-instance {
    VPN-A-Tokyo {
        instance-type vrf;
        interface ge-1/0/0.0;
        route-distinguisher 65535:1;
        vrf-import VPN-A-import;
```

```
        vrf-export VPN-A-export;
    }
}
```

Instance Routing Protocol

```
protocols {
    bgp {
        group VPN-A-Site2 {
            peer-as 1;
            neighbor 10.12.1.2;
        }
    }
}
```

Routing Instance for VPN-B

```
VPN-B-Osaka {
    instance-type vrf;
    interface at-1/2/0.0;
    route-distinguisher 65535:3;
    vrf-import VPN-B-import;
    vrf-export VPN-B-export;
}
```

Instance Routing Protocol

```
protocols {
    rip {
        group PE-C-to-VPN-B {
            neighbor at-1/2/0;
        }
    }
}
```

Master Protocol Instance

```
protocols {
}
```

Enable RSVP

```
rsvp {
    interface so-2/0/0.0;
}
```

Configure an MPLS LSP

```
mpls {
    label-switched-path RouterC-to-RouterA {
        to 10.255.245.68;
    }
    interface so-2/0/0.0;
    interface ge-1/0/0.0;
    interface at-1/2/0.0;
}
```

Configure IBGP

```
bgp {
    group PE-RouterC-to-PE-RouterA {
        type internal;
        local-address 10.255.245.47;
        family inet-vpn {
            unicast;
        }
        neighbor 10.255.245.68;
```

```
    }
}
```

Configure OSPF for Traffic Engineering Support

```
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface so-2/0/0.0;
    }
}
```

Configure VPN Policy

```
policy-options {
    policy-statement VPN-A-import {
        term a {
            from {
                protocol bgp;
                community VPN-A;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    policy-statement VPN-A-export {
        term a {
            from protocol bgp;
            then {
                community add VPN-A;
                accept;
            }
        }
        term b {
            then reject;
        }
    }
    policy-statement VPN-B-import {
        term a {
            from {
                protocol bgp;
                community VPN-B;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    policy-statement VPN-B-export {
        term a {
            from protocol rip;
            then {
                community add VPN-B;
                accept;
            }
        }
```

```
        term b {
            then reject;
        }
    }
    community VPN-A members target:65535:4;
    community VPN-B members target:65535:5;
}
```

Configuring a Full-Mesh VPN Topology with Route Reflectors

This example is a variation of the full-mesh VPN topology example (described in "Configuring a Simple Full-Mesh VPN Topology" on page 201) in which one of the PE routers is a BGP route reflector. In this variation, Router C in Figure 21 on page 202 is a route reflector. The only change to its configuration is that you need to include the cluster statement when configuring the BGP group:

```
[edit]
protocols {
    bgp {
        group PE-RouterC-to-PE-RouterA {
            type internal;
            local-address 10.255.245.47;
            family inet-vpn {
                unicast;
            }
            neighbor 10.255.245.68;
            cluster 4.3.2.1;
        }
    }
}
```

For the complete configuration example of Router C, see "Router C (PE Router)" on page 213.

Configuring Hub-and-Spoke VPN Topologies: One Interface

Use a one-interface configuration to advertise a default route from a hub or hubs.

Figure 22: Example of a Hub-and-Spoke VPN Topology with One Interface

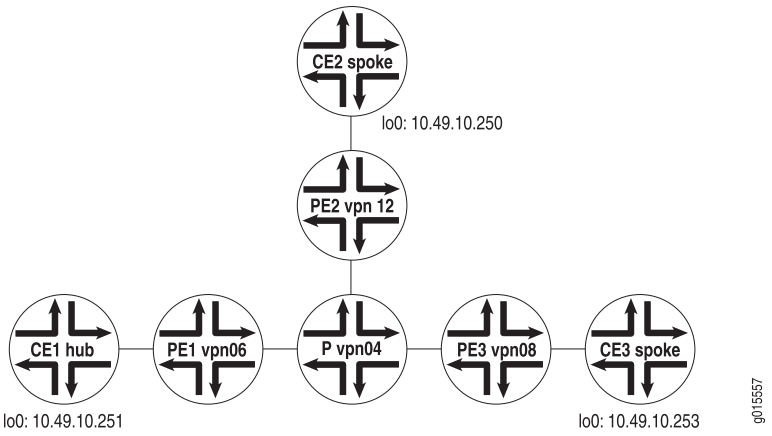


Figure 22 on page 217 illustrates a Layer 3 VPN hub-and-spoke application where there is only one interface between the hub CE (CE1) and the hub PE (PE1). This is the recommended way of configuring hub-and-spoke topologies.

In this configuration, a default route is advertised from the hub to the spokes. If more specific spoke CE routes need to be exchanged between spoke CE routers, then two interfaces are needed between the hub CE and hub PE. See "Configuring Hub-and-Spoke VPN Topologies: Two Interfaces" on page 229 for a two-interface example.

In this configuration example, spoke route distribution is as follows:

1. Spoke CE2 advertises its routes to spoke PE2.
2. Spoke PE2 installs routes from CE2 into its VPN routing and forwarding (VRF) table.
3. Spoke PE2 checks its VRF export policy, adds the route target community, and announces the routes to hub PE1.
4. Hub PE1 checks its VRF import policy and installs routes that match the import policy into table bgp.l3vpn.0.
5. Hub PE1 installs routes from table bgp.l3vpn.0 into the hub VRF table.
6. Hub PE1 announces routes from the hub VRF table to the hub CE1.

In this configuration example, default route distribution is as follows:

1. Hub CE1 announces a default route to hub PE1.
2. Hub PE1 installs the default route into the hub VRF table.
3. Hub PE1 checks its VRF export policy, adds the route target community and announces the default route to spoke PE2 and PE3.
4. Spoke PE2 and PE3 check their VRF import policy and install the default route into table bgp.l3vpn.0.

5. Spoke PE2 and PE3 install the routes from table bgp.l3vpn.0 into their spoke VRF tables.
6. Spoke PE2 and PE3 announce the default route from the spoke VRF table to spoke CE2 and CE3.

The following sections describe how to configure a hub-and-spoke topology with one interface based on the topology illustrated in Figure 22 on page 217:

Configuring Hub CE1

Configure hub CE1 as follows:

```
[edit routing-options]
static {
   route 0.0.0.0/0 discard;
}
autonomous-system 100;
[edit protocols]
bgp {
   group hub {
      type external;
      export default;
      peer-as 200;
      neighbor 10.49.4.1;
   }
}
[edit policy-statement]
default {
   term 1 {
      from {
         protocol static;
         route-filter 0.0.0.0/0 exact;
      }
      then accept;
   }
   term 2 {
      then reject;
   }
}
```

Configuring Hub PE1

Configure hub PE1 as follows:

```
[edit]
routing-instances {
    hub {
        instance-type vrf;
        interface t3-0/0/0 {
            encapsulation frame-relay;
            unit 0 {
                dlci 16;
                family inet {
                    address 10.49.4.1/30;
                }
            }
        }
        vrf-target {
            import target:200:100;
            export target:200:101;
        }
        protocols {
            bgp {
                group hub {
                    type external;
                    peer-as 100;
                    as-override;
                    neighbor 10.49.4.2;
                }
            }
        }
    }
}
```

Configuring the P Router

Configure the P Router as follows:

```
[edit]
interfaces {
    t3-0/1/1 {
        unit 0 {
            family inet {
                address 10.49.2.1/30;
            }
            family mpls;
        }
    }
    t3-0/1/3 {
        unit 0 {
            family inet {
                address 10.49.0.2/30;
            }
            family mpls;
        }
```

```
    }
    t1-0/2/0 {
        unit 0 {
            family inet {
                address 10.49.1.2/30;
            }
            family mpls;
        }
    }
}
[edit]
protocols {
    ospf {
        area 0.0.0.0 {
            interface t3-0/1/3.0;
            interface t1-0/2/0.0;
            interface t3-0/1/1.0;
            interface lo0.0 {
                passive;
            }
        }
    }
    ldp {
        interface t3-0/1/1.0;
        interface t3-0/1/3.0;
        interface t1-0/2/0.0;
    }
}
```

Configuring Spoke PE2

Configure spoke PE2 as follows:

```
[edit]
interfaces {
    t3-0/0/0 {
        unit 0 {
            family inet {
                address 10.49.0.1/30;
            }
            family mpls;
        }
    }
    t1-0/1/2 {
        unit 0 {
            family inet {
                address 10.49.3.1/30;
            }
        }
    }
}
[edit protocols]
bgp {
    group ibgp {
        type internal;
```

```
        local-address 10.255.14.182;
        peer-as 200;
        neighbor 10.255.14.176 {
            family inet-vpn {
                unicast;
            }
        }
    }
}
ospf {
    area 0.0.0.0 {
        interface t3-0/0/0.0;
        interface lo0.0 {
            passive;
        }
    }
}
ldp {
    interface t3-0/0/0.0;
}
[edit]
routing-instances {
    spoke {
        instance-type vrf;
        interface t1-0/1/2.0;
        vrf-target {
            import target:200:101;
            export target:200:100;
        }
        protocols {
            bgp {
                group spoke {
                    type external;
                    peer-as 100;
                    as-override;
                    neighbor 10.49.3.2;
                }
            }
        }
    }
}
```

Configuring Spoke PE3

Configure spoke PE3 as follows:

```
[edit]
interfaces {
    t3-0/0/0 {
        unit 0 {
            family inet {
                address 10.49.6.1/30;
            }
        }
    }
```

```
    t3-0/0/1 {
        unit 0 {
            family inet {
                address 10.49.2.2/30;
            }
            family mpls;
        }
    }
}
[edit protocols}
bgp {
    group ibgp {
        type internal;
        local-address 10.255.14.178;
        peer-as 200;
        neighbor 10.255.14.176 {
            family inet-vpn {
                unicast;
            }
        }
    }
}
ospf {
    area 0.0.0.0 {
        interface t3-0/0/1.0;
        interface lo0.0 {
            passive;
        }
    }
}
ldp {
    interface t3-0/0/1.0;
}
[edit]
routing-instances {
    spoke {
        instance-type vrf;
        interface t3-0/0/0.0;
        vrf-target {
            import target:200:101;
            export target:200:100;
        }
        protocols {
            bgp {
                group spoke {
                    type external;
                    peer-as 100;
                    as-override;
                    neighbor 10.49.6.2;
                }
            }
        }
    }
}
```

Configuring Spoke CE2

Configure spoke CE2 as follows:

```
[edit routing-options]
autonomous-system 100;
{edit protocols]
bgp {
   group spoke {
      type external;
      export loopback;
      peer-as 200;
      neighbor 10.49.3.1;
   }
}
```

Configuring Spoke CE3

Configure spoke CE3 as follows:

```
[edit routing-options]
autonomous-system 100;
[edit protocols]
bgp {
   group spoke {
      type external;
      export loopback;
      peer-as 200;
      neighbor 10.49.6.1;
   }
}
```

In this configuration example, traffic forwarding is as follows between spoke CE2 and hub CE1:

1. Spoke CE2 forwards traffic using the default route learned from spoke PE2 through BGP.

```
        0.0.0.0/0          *[BGP/170] 02:24:15, localpref 100
                              AS path: 200 200 I
                            > to 10.49.3.1 via t1-3/0/1.0
```

2. Spoke PE2 performs a route lookup in the spoke VRF table and forwards the traffic to hub PE2 (through the P router—PE2 pushes two labels) using the default route learned through BGP.

```
        0.0.0.0/0          *[BGP/170] 01:35:45, localpref 100, from
    10.255.14.176
                              AS path: 100 I
                            > via t3-0/0/1.0, Push 100336, Push 100224(top)
```

3. Hub PE1 does a route lookup in the mpls.0 table for the VPN label 100336.

```
100336             *[VPN/170] 01:37:03
                    > to 10.49.4.2 via t3-0/0/0.0, Pop
```

4. Hub PE1 forwards the traffic out the interface t3-0/0/0.0 to hub CE1.

In this configuration example, traffic forwarding is as follows between hub CE1 and spoke CE2:

1. Hub CE1 forwards traffic to the hub PE1 using the route learned through BGP.

```
10.49.10.250/32    *[BGP/170] 02:28:46, localpref 100
                      AS path: 200 200 I
                    > to 10.49.4.1 via t3-3/1/0.0
```

2. Hub PE1 does a route lookup in the hub VRF table and forwards the traffic to spoke PE2 (through the P router—PE1 pushes two labels).

```
    10.49.10.250/32    *[BGP/170] 01:41:05, localpref 100, from
10.255.14.182
                         AS path: 100 I
                       > via t1-0/1/0.0, Push 100352, Push 100208(top)
```

3. Spoke PE2 does a route lookup in the mpls.0 table for the VPN label 100352.

```
100352             *[VPN/170] 02:31:39
                    > to 10.49.3.2 via t1-0/1/2.0, Pop
```

4. Spoke PE2 forwards the traffic out the interface t1-0/1/2.0 to spoke CE2.

In this configuration example, traffic forwarding is as follows between spoke CE2 and spoke CE3:

1. Spoke CE2 forwards traffic using the default route learned from spoke PE2 through BGP.

```
0.0.0.0/0          *[BGP/170] 02:24:15, localpref 100
                      AS path: 200 200 I
                    > to 10.49.3.1 via t1-3/0/1.0
```

2. Spoke PE2 does a route lookup in the spoke VRF table and forwards the traffic to hub PE1 (through the P router—PE2 pushes two labels) using the default route learned through BGP.

```
    0.0.0.0/0          *[BGP/170] 01:35:45, localpref 100, from
10.255.14.176
                         AS path: 100 I
                       > via t3-0/0/1.0, Push 100336, Push 100224(top)
```

3. Hub PE1 does a route lookup in the mpls.0 table for the VPN label 100336.

```
100336             *[VPN/170] 01:37:03
                    > to 10.49.4.2 via t3-0/0/0.0, Pop
```

4. Hub PE1 forwards the traffic out the interface t3-0/0/0.0 to the hub CE1.
5. Hub CE1 forwards the traffic to hub PE1 using the router learned through BGP.

```
10.49.10.253/32    *[BGP/170] 02:40:03, localpref 100
                      AS path: 200 200 I
                    > to 10.49.4.1 via t3-3/1/0.0
```

6. Hub PE1 does a route lookup in the hub VRF table and forwards the traffic to spoke PE3 (through the P router—PE1 pushes two labels).

```
    10.49.10.253/32    *[BGP/170] 01:41:05, localpref 100, from
10.255.14.178
                         AS path: 100 I
                       > via t1-0/1/0.0, Push 100128, Push 100192(top)
```

7. Spoke PE3 does a route lookup in the mpls.0 table for VPN label 100128.

```
100128             *[VPN/170] 02:41:30
                    > to 10.49.6.2 via t3-0/0/0.0, Pop
```

8. Spoke PE3 forwards the traffic out the interface t3-0/0/0.0 to spoke CE3.

If egress features are needed on the hub PE that require an IP forwarding lookup on the hub VRF routing table, see "Enabling Egress Features on the Hub PE Router" on page 225.

Enabling Egress Features on the Hub PE Router

This example is provided in conjunction with "Configuring Hub-and-Spoke VPN Topologies: One Interface" on page 216. This example also uses the topology illustrated in Figure 22 on page 217.

If egress features are needed on the hub PE that require an IP forwarding lookup on the hub VRF routing table, the configuration detailed in "Configuring Hub-and-Spoke VPN Topologies: One Interface" on page 216 will not work. Applying the vrf-table-label statement on the hub routing instance forces traffic from a remote spoke PE to be forwarded to the hub PE and forces an IP lookup to be performed. Because specific spoke routes are in the hub VRF table, traffic will be forwarded to a spoke PE without going through the hub CE.

The hub PE advertises the default route as follows, using VPN label 1028:

```
hub.inet.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
* 0.0.0.0/0 (1 entry, 1 announced)
 BGP group ibgp type Internal
     Route Distinguisher: 10.255.14.176:2
     VPN Label: 1028
     Nexthop: Self
```

```
Localpref: 100
AS path: 100 I
Communities: target:200:101
```

Incoming traffic is forwarded using VPN label 1028. The mpls.0 table shows that an IP lookup in the table hub.inet.0 is required:

```
1028               *[VPN/0] 00:00:27
                      to table hub.inet.0, Pop
```

However, the hub VRF table hub.inet.0 contains specific spoke routes:

```
10.49.10.250/32    *[BGP/170] 00:00:05, localpref 100, from 10.255.14.182
                      AS path: 100 I
                    > via t1-0/1/0.0, Push 100352, Push 100208(top)
10.49.10.253/32    *[BGP/170] 00:00:05, localpref 100, from 10.255.14.178
                      AS path: 100 I
                    > via t1-0/1/0.0, Push 100128, Push 100192(top)
```

Because of this, traffic is forwarded directly to the spoke PEs without going through the hub CE. To prevent this, you must configure a secondary routing instance for downstream traffic in the hub PE1.

Configuring Hub PE1

Configure hub PE1 as follows:

```
[edit]
routing-instances {
   hub {
      instance-type vrf;
      interface t3-0/0/0.0;
      vrf-target {
         import target:200:100;
         export target:200:101;
      }
      no-vrf-advertise;
      routing-options {
         auto-export;
      }
      protocols {
         bgp {
            group hub {
               type external;
               peer-as 100;
               as-override;
               neighbor 10.49.4.2;
            }
         }
      }
   }
   hub-downstream {
      instance-type vrf;
      vrf-target target:200:101;
```

```
        vrf-table-label;
        routing-options {
            auto-export;
        }
    }
}
```

When the no-vrf-advertise statement is used at the [edit routing-instances hub] hierarchy level, no routing table groups or VRF export policies are required. The no-vrf-advertise statement configures the hub PE not to advertise VPN routes from the primary routing-instance hub. These routes are instead advertised from the secondary routing instance hub_downstream. See the routing instances configuration guidelines in the *JUNOS Routing Protocols Configuration Guide* for more information about the no-vrf-advertise statement.

The auto-export statement at the [edit routing-instances hub-downstream routing-options] hierarchy level identifies routes exported from the hub instance to the hub-downstream instance by looking at the route targets defined for each routing instance. See the routing instances configuration guidelines in the *JUNOS Routing Protocols Configuration Guide* for more information about using the auto-export statement. See "Configuring Overlapping VPNs Using Automatic Route Export" on page 280 for more examples of export policy.

With this configuration on hub PE, spoke-to-spoke CE traffic goes through the hub CE and permits egress features (such as filtering) to be enabled on the hub PE.

In this configuration example, traffic forwarding is as follows between spoke CE2 and spoke CE3:

1. Spoke CE2 forwards traffic using the default route learned from spoke PE2 through BGP.

```
0.0.0.0/0          *[BGP/170] 02:24:15, localpref 100
                      AS path: 200 200 I
                    > to 10.49.3.1 via t1-3/0/1.0
```

2. Spoke PE2 does a route lookup in the spoke VRF table and forwards the traffic to hub PE1 (through the P router—PE2 pushes two labels) using the default route learned through BGP.

```
spoke.inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)

+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[BGP/170] 00:00:09, localpref 100, from
10.255.14.176
                      AS path: 100 I
                    > via t3-0/0/0.0, Push 1029, Push 100224(top)
```

3. Hub PE1 does a route lookup in the mpls.0 table for the VPN label 1029.

```
mpls.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
1029               *[VPN/0] 00:11:49
                      to table hub_downstream.inet.0, Pop
```

The VPN label 1029 is advertised because:

a. The vrf-table-label statement is applied at the [edit routing-instances hub_downsteam] hierarchy level in the hub PE1 configuration.

b. The no-vrf-advertise statement is applied at the [edit routing-instances hub] hierarchy level, instructing the router to advertise the route from the secondary table.

Therefore, IP lookups are performed in the hub_downstream.inet.0 table, not in the hub.inet.0 table.

Issue the show route advertising-protocol command on the hub PE to a spoke PE to verify the VPN label 1029 advertisement:

```
user@host> show route advertising-protocol

   hub_downstream.inet.0: 2 destinations, 2 routes (2 active, 0 holddown,
0 hidden)
    * 0.0.0.0/0 (1 entry, 1 announced)
     BGP group ibgp type Internal
         Route Distinguisher: 10.255.14.176:3
         VPN Label: 1029
         Nexthop: Self
         Localpref: 100
         AS path: 100 I
         Communities: target:200:101
```

4. Hub PE1 performs an IP lookup in the hub_downstream.inet.0 table and forwards the traffic out interface t3-0/0/0.0 to hub CE1.

```
   hub_downstream.inet.0: 2 destinations, 2 routes (2 active, 0 holddown,
0 hidden)
   0.0.0.0/0 (1 entry, 1 announced)
           *BGP    Preference: 170/-101
                   Next-hop reference count: 4
                   Source: 10.49.4.2
                   Next hop: 10.49.4.2 via t3-0/0/0.0, selected
                   State: <Secondary Active Ext>
                   Peer AS:   100
                   Age: 3:03
                   Task: BGP_100.10.49.4.2+1707
                   Announcement bits (2): 0-KRT 2-BGP.0.0.0.0+179
                   AS path: 100 I
                   Communities: target:200:101
                   Localpref: 100
                   Router ID: 10.49.10.251
                   Primary Routing Table hub.inet.0
```

The primary routing table is hub.inet.0, indicating that this route was exported from table hub.inet.0 into this hub_downstream.inet.0 table as a result of the

no-vrf-advertise statement at the [edit routing-instances hub] hierarchy level and the auto-export statement at the [edit routing-instances hub-downstream routing-options] hierarchy level in the hub PE1 configuration.

5. Hub CE1 forwards the traffic back to hub PE1 using the router learned through BGP.

```
10.49.10.253/32    *[BGP/170] 02:40:03, localpref 100
                      AS path: 200 200 I
                    > to 10.49.4.1 via t3-3/1/0.0
```

6. Hub PE1 performs a route lookup in the hub VRF table and forwards the traffic to spoke PE3 (through the P router—PE1 pushes two labels).

```
    10.49.10.253/32    *[BGP/170] 01:41:05, localpref 100, from
10.255.14.178
                         AS path: 100 I
                       > via t1-0/1/0.0, Push 100128, Push 100192(top)
```

7. Spoke PE3 performs a route lookup in the mpls.0 table for VPN label 100128.

```
100128             *[VPN/170] 02:41:30
                    > to 10.49.6.2 via t3-0/0/0.0, Pop
```

8. Spoke PE3 forwards traffic out interface t3-0/0/0.0 to spoke CE3.

Configuring Hub-and-Spoke VPN Topologies: Two Interfaces

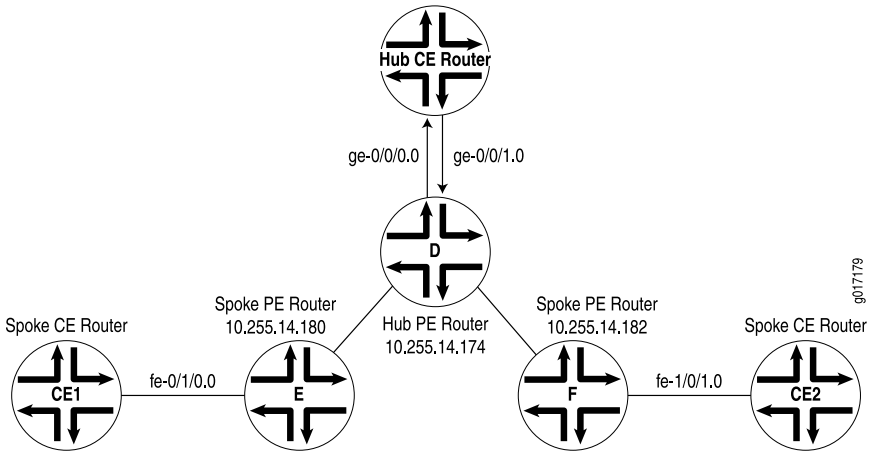
Use a two-interface configuration to propagate routes from spoke to spoke.

The example in this section configures a hub-and-spoke topology with two interfaces using the following components (see Figure 23 on page 230):

- One hub PE router (Router D).
- One hub CE router connected to the hub PE router. For this hub-and-spoke VPN topology to function properly, there must be two interfaces connecting the hub PE router to the hub CE router, and each interface must have its own VRF table on the PE router:
  - The first interface (here, interface ge-0/0/0.0) is used to announce spoke routes to the hub CE router. The VRF table associated with this interface contains the routes being announced by the spoke PE routers to the hub CE router.
  - The second interface (here, interface ge-0/0/1.0) is used to receive route announcements from the hub CE that are destined for the hub-and-spoke routers. The VRF table associated with this interface contains the routes announced by the hub CE router to the spoke PE routers. For this example, two separate physical interfaces are used. It would also work if you were to configure two separate logical interfaces sharing the same physical interface between the hub PE router and the hub CE router.
- Two spoke PE routers (Router E and Router F).

- Two spoke CE routers (CE1 and CE2), one connected to each spoke PE router.
- Label Distribution Protocol (LDP) as the signaling protocol.

Figure 23: Example of a Hub-and-Spoke VPN Topology with Two Interfaces



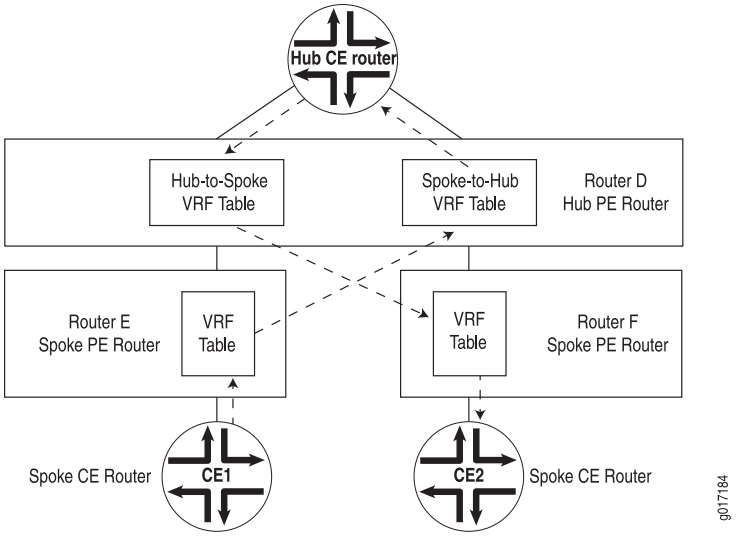
In this configuration, route distribution from spoke CE Router CE1 occurs as follows:

1. Spoke Router CE1 announces its routes to spoke PE Router E.
2. Router E installs the routes from CE1 into its VRF table.
3. After checking its VRF export policy, Router E adds the spoke target community to the routes from Router CE1 that passed the policy and announces them to the hub PE router, Router D.
4. Router D checks the VRF import policy associated with interface ge-0/0/0.0 and places all routes from spoke PE routers that match the policy into its bgp.l3vpn routing table. (Any routes that do not match are discarded.)
5. Router D checks its VRF import policy associated with interface ge-0/0/0.0 and installs all routes that match into its spoke VRF table. The routes are installed with the spoke target community.
6. Router D announces routes to the hub CE over interface ge-0/0/0.
7. The hub CE router announces the routes back to the hub PE Router D over the second interface to the hub router, interface ge-0/0/1.
8. The hub PE router installs the routes learned from the hub CE router into its hub VRF table, which is associated with interface ge-0/0/1.
9. The hub PE router checks the VRF export policy associated with interface ge-0/0/1.0 and announces all routes that match to all spokes after adding the hub target community.

Figure 24 on page 231 illustrates how routes are distributed from this spoke router to the other spoke CE router, Router CE2. The same path is followed if you issue a traceroute command from Router CE1 to Router CE2.

The final section in this example, "Hub-and-Spoke VPN Configuration Summarized by Router" on page 239, consolidates the statements needed to configure VPN functionality for each of the service provider routers shown in Figure 23 on page 230.

Figure 24: Route Distribution Between Two Spoke Routers



The following sections explain how to configure the VPN functionality for a hub-and-spoke topology on the hub-and-spoke PE routers. The CE routers do not have any information about the VPN, so you configure them normally.

Enabling an IGP on the Hub-and-Spoke PE Routers

To allow the hub-and-spoke PE routers to exchange routing information, you must configure an IGP on all these routers or you must configure static routes. You configure the IGP on the master instance of the routing protocol process (rpd) (that is, at the [edit protocols] hierarchy level), not within the routing instance (that is, not at the [edit routing-instances] hierarchy level).

You configure the IGP in the standard way. This configuration example does not include this portion of the configuration.

In the route distribution in a hub-and-spoke topology, if the protocol used between the CE and PE routers at the hub site is BGP, the hub CE router announces all routes received from the hub PE router and the spoke routers back to the hub PE router and all the spoke routers. This means that the hub-and-spoke PE routers receive

routes that contain their AS number. Normally, when a route contains this information, it indicates that a routing loop has occurred and the router rejects the routes. However, for the VPN configuration to work, the hub PE router and the spoke routers must accept these routes. To enable this, include the **loops** option when configuring the AS at the **[edit routing-options]** hierarchy level on the hub PE router and all the spoke routers. For this example configuration, you specify a value of 1. You can specify a number from 0 through 10.

```
[edit routing-options]
autonomous-system as-number loops 1;
```

Configuring LDP on the Hub-and-Spoke PE Routers

Configure LDP on the interfaces between the hub-and-spoke PE routers that participate in the VPN.

On hub PE Router D, configure LDP:

```
[edit protocols]
ldp {
   interface so-1/0/0.0;
   interface t3-1/1/0.0;
}
```

On spoke PE Router E, configure LDP:

```
[edit protocols]
ldp {
   interface fe-0/1/2.0;
}
```

On spoke PE router Router F, configure LDP:

```
[edit protocols]
ldp {
   interface fe-1/0/0.0;
}
```

Configuring IBGP on the PE Routers

On the hub-and-spoke PE routers, configure an IBGP session with the following properties:

- VPN family—To indicate that the IBGP session is for the VPN, include the **family inet-vpn** statement.
- Loopback address—Include the **local-address** statement, specifying the local PE router's loopback address. The IBGP session for VPNs runs through the loopback address. You must also configure the **lo0** interface at the **[edit interfaces]** hierarchy level. The example does not include this part of the router's configuration.
- Neighbor address—Include the **neighbor** statement. On the hub router, specify the IP address of each spoke PE router, and on the spoke router, specify the address of the hub PE router.

For the hub router, you configure an IBGP session with each spoke, and for each spoke router, you configure an IBGP session with the hub. There are no IBGP sessions between the two spoke routers.

On hub Router D, configure IBGP. The first neighbor statement configures an IBGP session to spoke Router E, and the second configures a session to spoke Router F.

```
[edit protocols]
bgp {
    group Hub-to-Spokes {
        type internal;
        local-address 10.255.14.174;
        family inet-vpn {
            unicast;
        }
        neighbor 10.255.14.180;
        neighbor 10.255.14.182;
    }
}
```

On spoke Router E, configure an IBGP session to the hub router:

```
[edit protocols]
bgp {
    group Spoke-E-to-Hub {
        type internal;
        local-address 10.255.14.180;
        neighbor 10.255.14.174 {
            family inet-vpn {
                unicast;
            }
        }
    }
}
```

On spoke Router F, configure an IBGP session to the hub router:

```
[edit protocols]
bgp {
    group Spoke-F-to-Hub {
        type internal;
        local-address 10.255.14.182;
        neighbor 10.255.14.174 {
            family inet-vpn {
                unicast;
            }
        }
    }
}
```

Configuring VPN Routing Instances on the Hub-and-Spoke PE Routers

For the hub PE router to be able to distinguish between packets going to and coming from the spoke PE routers, you must configure it with two routing instances:

- One routing instance (in this example, Spokes-to-Hub-CE) is associated with the interface that carries packets from the hub PE router to the hub CE router (in this example, interface ge-0/0/0.0). Its VRF table contains the routes being announced by the spoke PE routers and the hub PE router to the hub CE router.
- The second routing instance (in this example, Hub-CE-to-Spokes) is associated with the interface that carries packets from the hub CE router to the hub PE router (in this example, interface ge-0/0/1.0). Its VRF table contains the routes being announced from the hub CE router to the hub-and-spoke PE routers.

On each spoke router, you must configure one routing instance.

You must define the following in the routing instance:

- Route distinguisher, which is used to distinguish the addresses in one VPN from those in another VPN.
- Instance type of vrf, which creates the VRF table on the PE router.
- Interfaces that are part of the VPN and that connect the PE routers to their CE routers.
- VRF import and export policies. Both import policies must include reference to a community. Otherwise, when you try to commit the configuration, the commit fails. (The exception to this is if the import policy contains only a then reject statement.) In the VRF export policy, spoke PE routers attach the spoke target community.
- Routing between the PE and CE routers, which is required for the PE router to distribute VPN-related routes to and from connected CE routers. You can configure a routing protocol—BGP, OSPF, or RIP—or you can configure static routing.

For a hub-and-spoke topology, you must configure different policies in each routing instance on the hub CE router. For the routing instance associated with the interface that carries packets from the hub PE router to the hub CE router (in this example, Spokes-to-Hub-CE), the import policy must accept all routes received on the IBGP session between the hub-and-spoke PE routers, and the export policy must reject all routes received from the hub CE router. For the routing instance associated with the interface that carries packets from the hub CE router to the hub PE router (in this example, Hub-CE-to-Spokes), the import policy must reject all routes received from the spoke PE routers, and the export policy must export to all the spoke routers.

On hub PE Router D, configure the following routing instances. Router D uses OSPF to distribute routes to and from the hub CE router.

```
[edit]
routing-instance {
   Spokes-to-Hub-CE {
      instance-type vrf;
      interface ge-0/0/0.0;
      route-distinguisher 10.255.1.174:65535;
```

```
        vrf-import spoke;
        vrf-export null;
        protocols {
            ospf {
                export redistribute-vpn;
                area 0.0.0.0 {
                    interface ge-0/0/0;
                }
            }
        }
    }
    Hub-CE-to-Spokes {
        instance-type vrf;
        interface ge-0/0/1.0;
        route-distinguisher 10.255.1.174:65535;
        vrf-import null;
        vrf-export hub;
        protocols {
            ospf {
                export redistribute-vpn;
                area 0.0.0.0 {
                    interface ge-0/0/1.0;
                }
            }
        }
    }
}
```

On spoke PE Router E, configure the following routing instances. Router E uses OSPF to distribute routes to and from spoke CE Router CE1.

```
[edit]
routing-instance {
    Spoke-E-to-Hub {
        instance-type vrf;
        interface fe-0/1/0.0;
        route-distinguisher 10.255.14.80:65535;
        vrf-import hub;
        vrf-export spoke;
        protocols {
            ospf {
                export redistribute-vpn;
                area 0.0.0.0 {
                    interface fe-0/1/0.0;
                }
            }
        }
    }
}
```

On spoke PE Router F, configure the following routing instances. Router F uses OSPF to distribute routes to and from spoke CE Router CE2.

```
[edit]
routing-instance {
    Spoke-F-to-Hub {
```

```
        instance-type vrf;
        interface fe-1/0/1.0;
        route-distinguisher 10.255.14.182:65535;
        vrf-import hub;
        vrf-export spoke;
        protocols {
            ospf {
                export redistribute-vpn;
                area 0.0.0.0 {
                    interface fe-1/0/1.0;
                }
            }
        }
    }
}
```

Configuring VPN Policy on the PE Routers

You must configure VPN import and export policies on each of the hub-and-spoke PE routers so that they install the appropriate routes in the VRF tables, which they use to forward packets within each VPN.

On the spoke routers, you define policies to exchange routes with the hub router.

On the hub router, you define policies to accept routes from the spoke PE routers and distribute them to the hub CE router, and vice versa. The hub PE router has two VRF tables:

- Spoke-to-hub VRF table—Handles routes received from spoke routers and announces these routes to the hub CE router. For this VRF table, the import policy must check that the spoke target name is present and that the route was received from the IBGP session between the hub PE and the spoke PE routers. This VRF table must not export any routes, so its export policy should reject everything.
- Hub-to-spoke VRF table—Handles routes received from the hub CE router and announces them to the spoke routers. For this VRF table, the export policy must add the hub target community. This VRF table must not import any routes, so its import policy should reject everything.

In the VPN policy, you also configure the VPN target communities.

On hub PE Router D, configure the following policies to apply to the VRF tables:

- spoke—Accepts routes received from the IBGP session between it and the spoke PE routers that contain the community target spoke, and rejects all other routes.
- hub—Adds the community target hub to all routes received from OSPF (that is, from the session between it and the hub CE router). It rejects all other routes.
- null—Rejects all routes.
- redistribute-vpn—Redistributes OSPF routes to neighbors within the routing instance.

```
[edit]
```

```
    policy-options {
      policy-statement spoke {
        term a {
          from {
            protocol bgp;
            community spoke;
          }
          then accept;
        }
        term b {
          then reject;
        }
      }
      policy-statement hub {
        term a {
          from protocol ospf;
          then {
            community add hub;
            accept;
          }
        }
        term b {
          then reject;
        }
      }
      policy-statement null {
        then reject;
      }
      policy-statement redistribute-vpn {
        term a {
          from protocol bgp;
          then accept;
        }
        term b {
          then reject;
        }
      }
      community hub members target:65535:1;
      community spoke members target:65535:2;
    }
```

To apply the VRF policies on Router D, include the **vrf-export** and **vrf-import** statements when you configure the routing instances:

```
[edit]
routing-instance {
  Spokes-to-Hub-CE {
    vrf-import spoke;
    vrf-export null;
  }
  Hub-CE-to-Spokes {
    vrf-import null;
    vrf-export hub;
  }
}
```

On spoke PE Router E and Router F, configure the following policies to apply to the VRF tables:

- hub—Accepts routes received from the IBGP session between it and the hub PE routers that contain the community target hub, and rejects all other routes.
- spoke—Adds the community target spoke to all routes received from OSPF (that is, from the session between it and the hub CE router) rejects all other routes.
- redistribute-vpn—Redistributes OSPF routes to neighbors within the routing instance.

On spoke PE Router E and Router F, configure the following VPN import and export policies:

```
[edit]
policy-options {
    policy-statement hub {
        term a {
            from {
                protocol bgp;
                community hub;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    policy-statement spoke {
        term a {
            from protocol ospf;
            then {
                community add spoke;
                accept;
            }
        }
        term b {
            then reject;
        }
    }
    policy-statement redistribute-vpn {
        term a {
            from protocol bgp;
            then accept;
        }
        term b {
            then reject;
        }
    }
    community hub members target:65535:1;
    community spoke members target 65535:2;
}
```

To apply the VRF policies on the spoke routers, include the vrf-export and vrf-import statements when you configure the routing instances:

```
[edit]
routing-instance {
   Spoke-E-to-Hub {
      vrf-import hub;
      vrf-export spoke;
   }
}
[edit]
routing-instance {
   Spoke-F-to-Hub {
      vrf-import hub;
      vrf-export spoke;
   }
}
```

Hub-and-Spoke VPN Configuration Summarized by Router

Router D (Hub PE Router)

Routing Instance for Distributing Spoke Routes to Hub CE

```
routing-instance {
   Spokes-to-Hub-CE {
      instance-type vrf;
      interface ge-0/0/0.0;
      route-distinguisher 10.255.1.174:65535;
      vrf-import spoke;
      vrf-export null;
   }
}
```

Instance Routing Protocol

```
protocols {
   ospf {
      export redistribute-vpn;
      area 0.0.0.0 {
         interface ge-0/0/0;
      }
   }
}
```

Routing Instance for Distributing Hub CE Routes to Spokes

```
Hub-CE-to-Spokes {
   instance-type vrf;
   interface ge-0/0/1.0;
   route-distinguisher 10.255.1.174:65535;
   vrf-import null;
   vrf-export hub;
}
```

Routing Instance Routing Protocols

```
protocols {
   ospf {
      export redistribute-vpn;
      area 0.0.0.0 {
         interface ge-0/0/1.0;
      }
   }
}
```

Routing Options (Master Instance)

```
routing-options {
   autonomous-system 1 loops 1;
}
```

Protocols (Master Instance)

```
protocols {
}
```

Enable LDP

```
ldp {
   interface so-1/0/0.0;
   interface t3-1/1/0.0;
}
```

Configure IBGP

```
bgp {
   group Hub-to-Spokes {
      type internal;
      local-address 10.255.14.174;
      family inet-vpn {
         unicast;
      }
      neighbor 10.255.14.180;
      neighbor 10.255.14.182;
   }
}
```

Configure VPN Policy

```
policy-options {
   policy-statement spoke {
      term a {
         from {
            protocol bgp;
            community spoke;
         }
         then accept;
      }
      term b {
         then reject;
      }
   }
   policy-statement hub {
      term a {
         from protocol ospf;
         then {
            community add hub;
            accept;
         }
      }
      term b {
         then reject;
      }
   }
   policy-statement null {
      then reject;
   }
   policy-statement redistribute-vpn {
      term a {
```

```
        from protocol bgp;
        then accept;
      }
      term b {
        then reject;
      }
    }
    community hub members target:65535:1;
    community spoke members target:65535:2;
}
```

Router E (Spoke PE Router)

Routing Instance

```
routing-instance {
    Spoke-E-to-Hub {
        instance-type vrf;
        interface fe-0/1/0.0;
        route-distinguisher 10.255.14.80:65535;
        vrf-import hub;
        vrf-export spoke;
    }
}
```

Instance Routing Protocol

```
protocols {
    ospf {
        export redistribute-vpn;
        area 0.0.0.0 {
            interface fe-0/1/0.0;
        }
    }
}
```

Routing Options (Master Instance)

```
routing-options {
    autonomous-system 1 loops 1;
}
```

Protocols (Master Instance)

```
protocols {
}
```

Enable LDP

```
ldp {
    interface fe-0/1/2.0;
}
```

Configure IBGP

```
bgp {
    group Spoke-E-to-Hub {
        type internal;
        local-address 10.255.14.180;
        neighbor 10.255.14.174 {
            family inet-vpn {
                unicast;
            }
        }
    }
```

```
}
```

Configure VPN Policy

```
policy-options {
   policy-statement hub {
      term a {
         from {
            protocol bgp;
            community hub;
         }
         then accept;
      }
      term b {
         then reject;
      }
   }
   policy-statement spoke {
      term a {
         from protocol ospf;
         then {
            community add spoke;
            accept;
         }
      }
      term b {
         then reject;
      }
   }
   policy-statement redistribute-vpn {
      term a {
         from protocol bgp;
         then accept;
      }
      term b {
         then reject;
      }
   }
   community hub members target:65535:1;
   community spoke members target:65535:2;
}
```

Router F (Spoke PE Router)

Routing Instance

```
routing-instance {
   Spoke-F-to-Hub {
      instance-type vrf;
      interface fe-1/0/1.0;
      route-distinguisher 10.255.14.182:65535;
      vrf-import hub;
      vrf-export spoke;
   }
}
```

Instance Routing Protocol

```
protocols {
   ospf {
```

```
    export redistribute-vpn;
    area 0.0.0.0 {
      interface fe-1/0/1.0;
    }
  }
}
```

Routing Options (Master Instance)

```
routing-options {
  autonomous-system 1 loops 1;
}
```

Protocols (Master Instance)

```
protocols {
}
```

Enable LDP

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

Configure IBGP

```
bgp {
  group Spoke-F-to-Hub {
    type internal;
    local-address 10.255.14.182;
    neighbor 10.255.14.174 {
      family inet-vpn {
        unicast;
      }
    }
  }
}
```

Configure VPN Policy

```
policy-options {
  policy-statement hub {
    term a {
      from {
        protocol bgp;
        community hub;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement spoke {
    term a {
      from protocol ospf;
      then {
        community add spoke;
        accept;
      }
    }
    term b {
      then reject;
    }
```

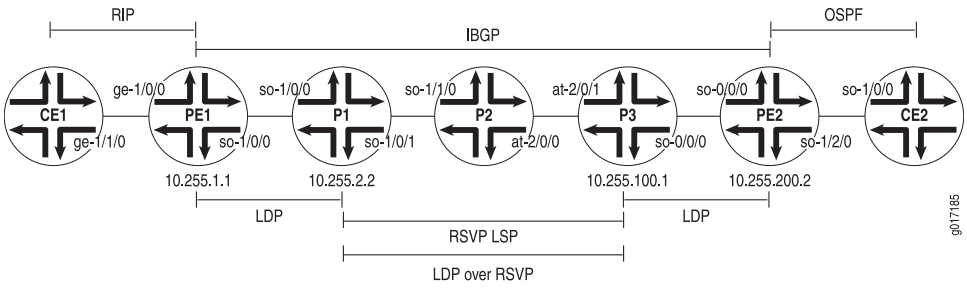
```
    }
    policy-statement redistribute-vpn {
        term a {
            from {
                protocol bgp;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    community hub members target:65535:1;
    community spoke members target:65535:2;
}
```

Configuring an LDP-over-RSVP VPN Topology

This example shows how to set up a VPN topology in which LDP packets are tunneled over an RSVP LSP. This configuration consists of the following components (see Figure 25 on page 244):

- One VPN (VPN-A)
- Two PE routers
- LDP as the signaling protocol between the PE routers and their adjacent P routers
- An RSVP LSP between two of the P routers over which LDP is tunneled

Figure 25: Example of an LDP-over-RSVP VPN Topology



The following steps describe how this topology is established and how packets are sent from CE Router CE2 to CE Router CE1:

1. The P routers P1 and P3 establish RSVP LSPs between each other and install their loopback addresses in their inet.3 routing tables.
2. PE Router PE1 establishes an LDP session with Router P1 over interface so-1/0/0.0.
3. Router P1 establishes an LDP session with Router P3's loopback address, which is reachable using the RSVP LSP.

4. Router P1 sends its label bindings, which include a label to reach Router PE1, to Router P3. These label bindings allow Router P3 to direct LDP packets to Router PE1.
5. Router P3 establishes an LDP session with Router PE2 over interface so0-0/0/0.0 and establishes an LDP session with Router P1's loopback address.
6. Router P3 sends its label bindings, which include a label to reach Router PE2, to Router P1. These label bindings allow Router P1 to direct LDP packets to Router PE2's loopback address.
7. Routers PE1 and PE2 establish IBGP sessions with each other.
8. When Router PE1 announces to Router PE2 routes that it learned from Router CE1, it includes its VPN label. (The PE router creates the VPN label and binds it to the interface between the PE and CE routers.) Similarly, when Router PE2 announces routes that it learned from Router CE2, it sends its VPN label to Router PE1.

When Router PE2 wants to forward a packet to Router CE1, it pushes two labels onto the packet's label stack: first the VPN label that is bound to the interface between Router PE1 and Router CE1, then the LDP label used to reach Router PE1. Then it forwards the packets to Router P3 over interface so-0/0/1.0.

1. When Router P3 receives the packets from Router PE2, it swaps the LDP label that is on top of the stack (according to its LDP database) and also pushes an RSVP label onto the top of the stack so that the packet can now be switched by the RSVP LSP. At this point, there are three labels on the stack: the inner (bottom) label is the VPN label, the middle is the LDP label, and the outer (top) is the RSVP label.
2. Router P2 receives the packet and switches it to Router P1 by swapping the RSVP label. In this topology, because Router P2 is the penultimate-hop router in the LSP, it pops the RSVP label and forwards the packet over interface so-1/1/0.0 to Router P1. At this point, there are two labels on the stack: The inner label is the VPN label, and the outer one is the LDP label.
3. When Router P1 receives the packet, it pops the outer label (the LDP label) and forwards the packet to Router PE1 using interface so-1/0/0.0. In this topology, Router PE1 is the egress LDP router, so Router P1 pops the LDP label instead of swapping it with another label. At this point, there is only one label on the stack, the VPN label.
4. When Router PE1 receives the packet, it pops the VPN label and forwards the packet as an IPv4 packet to Router CE1 over interface ge-1/1/0.0.

A similar set of operations occurs for packets sent from Router CE1 that are destined for Router CE2.

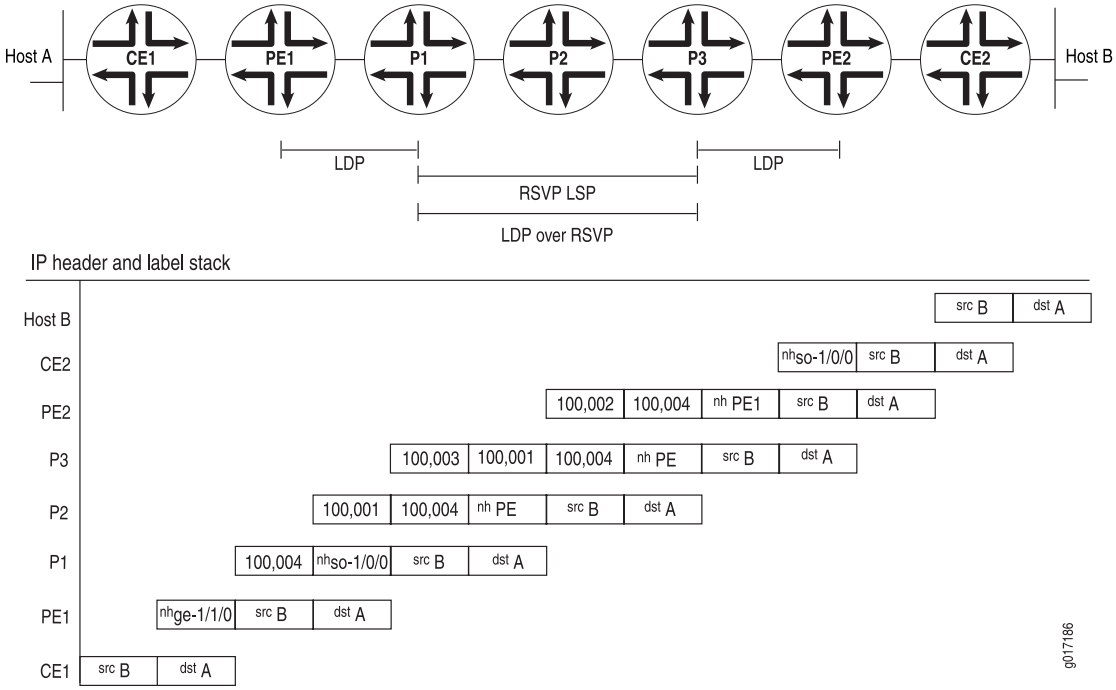
The following list explains how, for packets being sent from Router CE2 to Router CE1, the LDP, RSVP, and VPN labels are announced by the various routers. These steps include examples of label values (illustrated in Figure 26 on page 246).

- LDP labels
  - Router PE1 announces LDP label 3 for itself to Router P1.
  - Router P1 announces LDP label 100,001 for Router PE1 to Router P3.

- Router P3 announces LDP label 100,002 for Router PE1 to Router PE2.

- RSVP labels
  - Router P1 announces RSVP label 3 to Router P2.
  - Router P2 announces RSVP label 100,003 to Router P3.
- VPN label
  - Router PE1 announces VPN label 100,004 to Router PE2 for the route from Router CE1 to Router CE2.

Figure 26: Label Pushing and Popping



For a packet sent from Host B in Figure 26 on page 246 to Host A, the packet headers and labels change as the packet travels to its destination:

1. The packet that originates from Host B has a source address of B and a destination address of A in its header.
2. Router CE2 adds to the packet a next-hop of interface so-1/0/0.
3. Router PE2 swaps out the next-hop of interface so-1/0/0 and replaces it with a next-hop of PE1. It also adds two labels for reaching Router PE1, first the VPN label (100,004), then the LDP label (100,002). The VPN label is thus the inner (bottom) label on the stack, and the LDP label is the outer label.
4. Router P3 swaps out the LDP label added by Router PE2 (100,002) and replaces it with its LDP label for reaching Router PE1 (100,001). It also adds the RSVP label for reaching Router P2 (100,003).

5. Router P2 removes the RSVP label (100,003) because it is the penultimate hop in the MPLS LSP.
6. Router P1 removes the LDP label (100,001) because it is the penultimate LDP router. It also swaps out the next-hop of PE1 and replaces it with the next-hop interface, so-1/0/0.
7. Router PE1 removes the VPN label (100,004). It also swaps out the next-hop interface of so-1/0/0 and replaces it with its next-hop interface, ge-1/1/0.
8. Router CE1 removes the next-hop interface of ge-1/1/0, and the packet header now contains just a source address of B and a destination address of A.

The final section in this example, "LDP-over-MPLS VPN Configuration Summarized by Router" on page 254, consolidates the statements needed to configure VPN functionality on each of the service P routers shown in Figure 25 on page 244.

NOTE: In this example, a private AS number is used for the route distinguisher and the route target. This number is used for illustration only. When you are configuring VPNs, you should use an assigned AS number.

The following sections explain how to configure the VPN functionality on the PE and P routers. The CE routers do not have any information about the VPN, so you configure them normally.

Enabling an IGP on the PE and P Routers

To allow the PE and P routers to exchange routing information among themselves, you must configure an IGP on all these routers or you must configure static routes. You configure the IGP on the master instance of the routing protocol process (rpd) (that is, at the [edit protocols] hierarchy level), not within the VPN routing instance (that is, not at the [edit routing-instances] hierarchy level).

You configure the IGP in the standard way. This configuration example does not include this portion of the configuration.

Enabling LDP on the PE and P Routers

In this configuration example, the LDP is the signaling protocol between the PE routers. For the VPN to function, you must configure LDP on the two PE routers and

on the P routers that are connected to the PE routers. You need to configure LDP only on the interfaces in the core of the service provider's network; that is, between the PE and P routers and between the P routers. You do not need to configure LDP on the interface between the PE and CE routers.

In this configuration example, you configure LDP on the P routers' loopback interfaces because these are the interfaces on which the MPLS LSP is configured.

On the PE routers, you must also configure family inet when you configure the logical interface.

On Router PE1, configure LDP:

```
[edit protocols]
ldp {
   interface so-1/0/0.0;
}
[edit interfaces]
so-1/0/0 {
   unit 0 {
      family mpls;
   }
}
```

On Router PE2, configure LDP:

```
[edit protocols]
ldp {
   interface so-0/0/0.0;
}
[edit interfaces]
so-0/0/1 {
   unit 0 {
      family mpls;
   }
}
```

On Router P1, configure LDP:

```
[edit protocols]
ldp {
   interface so-1/0/0.0;
   interface lo0;
}
```

On Router P3, configure LDP:

```
[edit protocols]
ldp {
   interface lo0;
   interface so-0/0/0.0;
}
```

On Router P2, although you do not need to configure LDP, you can optionally configure it to provide a fallback LDP path in case the RSVP LSP becomes nonoperational:

```
[edit protocols]
ldp {
   interface so-1/1/0.0;
   interface at-2/0/0.0;
}
```

Enabling RSVP and MPLS on the P Router

On the P Router P2 you must configure RSVP and MPLS because this router exists on the MPLS LSP path between the P Routers P1 and P3:

```
[edit]
protocols {
   rsvp {
      interface so-1/1/0.0;
      interface at-2/0/0.0;
   }
   mpls {
      interface so-1/1/0.0;
      interface at-2/0/0.0;
   }
}
```

Configuring the MPLS LSP Tunnel Between the P Routers

In this configuration example, LDP is tunneled over an RSVP LSP. Therefore, in addition to configuring RSVP, you must enable traffic engineering support in an IGP, and you must create an MPLS LSP to tunnel the LDP traffic.

On Router P1, enable RSVP and configure one end of the MPLS LSP tunnel. In this example, traffic engineering support is enabled for OSPF, and you configure MPLS on the interfaces to the LSP and to Router PE1. In the to statement, you specify the loopback address of Router P3.

```
[edit]
protocols {
   rsvp {
      interface so-1/0/1.0;
   }
   mpls {
      label-switched-path P1-to-P3 {
         to 10.255.100.1;
         ldp-tunneling;
      }
      interface so-1/0/0.0;
      interface so-1/0/1.0;
   }
   ospf {
      traffic-engineering;
      area 0.0.0.0 {
```

```
            interface so-1/0/0.0;
            interface so-1/0/1.0;
        }
    }
}
```

On Router P3, enable RSVP and configure the other end of the MPLS LSP tunnel. Again, traffic engineering support is enabled for OSPF, and you configure MPLS on the interfaces to the LSP and to Router PE2. In the to statement, you specify the loopback address of Router P1.

```
[edit]
protocols {
    rsvp {
        interface at-2/0/1.0;
    }
    mpls {
        label-switched-path P3-to-P1 {
            to 10.255.2.2;
            ldp-tunneling;
        }
        interface at-2/0/1.0;
        interface so-0/0/0.0;
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface at-2/0/1.0;
            interface so-0/0/0.0;
        }
    }
}
```

Configuring IBGP on the PE Routers

On the PE routers, configure an IBGP session with the following properties:

- VPN family—To indicate that the IBGP session is for the VPN, include the family inet-vpn statement.
- Loopback address—Include the local-address statement, specifying the local PE router's loopback address. The IBGP session for VPNs runs through the loopback address. You must also configure the lo0 interface at the [edit interfaces] hierarchy level. The example does not include this part of the router's configuration.
- Neighbor address—Include the neighbor statement, specifying the IP address of the neighboring PE router, which is its loopback (lo0) address.

On Router PE1, configure IBGP:

```
[edit]
protocols {
    bgp {
        group PE1-to-PE2 {
            type internal;
```

```
            local-address 10.255.1.1;
            family inet-vpn {
               unicast;
            }
            neighbor 10.255.200.2;
         }
      }
   }
```

On Router PE2, configure IBGP:

```
[edit]
protocols {
   bgp {
      group PE2-to-PE1 {
         type internal;
         local-address 10.255.200.2;
         family inet-vpn {
            unicast;
         }
         neighbor 10.255.1.1;
      }
   }
}
```

Configuring Routing Instances for VPNs on the PE Routers

Both PE routers service VPN-A, so you must configure one routing instance on each router for the VPN in which you define the following:

- Route distinguisher, which must be unique for each routing instance on the PE router. It is used to distinguish the addresses in one VPN from those in another VPN.
- Instance type of vrf, which creates the VRF table on the PE router.
- Interfaces connected to the CE routers.
- VRF import and export policies, which must be the same on each PE router that services the same VPN. Unless the import policy contains only a then reject statement, it must include reference to a community. Otherwise, when you try to commit the configuration, the commit fails.

NOTE: In this example, a private AS number is used for the route distinguisher. This number is used for illustration only. When you are configuring VPNs, you should use an assigned AS number.

- Routing between the PE and CE routers, which is required for the PE router to distribute VPN-related routes to and from connected CE routers. You can configure a routing protocol—BGP, OSPF, or RIP—or you can configure static routing.

On Router PE1, configure the following routing instance for VPN-A. In this example, Router PE1 uses RIP to distribute routes to and from the CE router to which it is connected.

```
[edit]
routing-instance {
   VPN-A {
      instance-type vrf;
      interface ge-1/0/0.0;
      route-distinguisher 65535:0;
      vrf-import VPN-A-import;
      vrf-export VPN-A-export;
      protocols {
         rip {
            group PE1-to-CE1 {
               neighbor ge-1/0/0.0;
            }
         }
      }
   }
}
```

On Router PE2, configure the following routing instance for VPN-A. In this example, Router PE2 uses OSPF to distribute routes to and from the CE router to which it is connected.

```
[edit]
routing-instance {
   VPN-A {
      instance-type vrf;
      interface so-1/2/0.0;
      route-distinguisher 65535:1;
      vrf-import VPN-A-import;
      vrf-export VPN-A-export;
      protocols {
         ospf {
            area 0.0.0.0 {
               interface so-1/2/0.0;
            }
         }
      }
   }
}
```

Configuring VPN Policy on the PE Routers

You must configure VPN import and export policies on each of the PE routers so that they install the appropriate routes in their VRF tables, which they use to forward packets within a VPN. For VPN-A, the VRF table is VPN-A.inet.0.

In the VPN policy, you also configure VPN target communities.

NOTE: In this example, a private AS number is used for the route target. This number is used for illustration only. When you are configuring VPNs, you should use an assigned AS number.

On Router PE1, configure the following VPN import and export policies:

NOTE: The policy qualifiers shown in this example are only those needed for the VPN to function. You can configure additional qualifiers, as needed, to any policies that you configure.

```
[edit]
policy-options {
   policy-statement VPN-A-import {
      term a {
         from {
            protocol bgp;
            community VPN-A;
         }
         then accept;
      }
      term b {
         then reject;
      }
   }
   policy-statement VPN-A-export {
      term a {
         from protocol rip;
         then {
            community add VPN-A;
            accept;
         }
      }
      term b {
         then reject;
      }
   }
   community VPN-A members target:65535:00;
}
```

On Router PE2, configure the following VPN import and export policies:

```
[edit]
policy-options {
   policy-statement VPN-A-import {
      term a {
         from {
            protocol bgp;
            community VPN-A;
         }
         then accept;
      }
```

```
        term b {
            then reject;
        }
    }
    policy-statement VPN-A-export {
        term a {
            from protocol ospf;
            then {
                community add VPN-A;
                accept;
            }
        }
        term b {
            then reject;
        }
    }
    community VPN-A members target:65535:00;
}
```

To apply the VPN policies on the routers, include the vrf-export and vrf-import statements when you configure the routing instance on the PE routers. The VRF import and export policies handle the route distribution across the IBGP session running between the PE routers.

LDP-over-MPLS VPN Configuration Summarized by Router

Router PE1

Routing Instance for VPN-A

```
routing-instance {
    VPN-A {
        instance-type vrf;
        interface ge-1/0/0.0;
        route-distinguisher 65535:0;
        vrf-import VPN-A-import;
        vrf-export VPN-A-export;
    }
}
```

Instance Routing Protocol

```
protocols {
    rip {
        group PE1-to-CE1 {
            neighbor ge-1/0/0.0;
        }
    }
}
```

Interfaces

```
interfaces {
    so-1/0/0 {
        unit 0 {
            family mpls;
        }
    }
    ge-1/0/0 {
        unit 0;
```

```
    }
}
```

Master Protocol Instance

```
protocols {
}
```

Enable LDP

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

Enable MPLS

```
mpls {
    interface so-1/0/0.0;
    interface ge-1/0/0.0;
}
```

Configure IBGP

```
bgp {
    group PE1-to-PE2 {
        type internal;
        local-address 10.255.1.1;
        family inet-vpn {
            unicast;
        }
        neighbor 10.255.100.1;
    }
}
```

Configure VPN Policy

```
policy-options {
    policy-statement VPN-A-import {
        term a {
            from {
                protocol bgp;
                community VPN-A;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    policy-statement VPN-A-export {
        term a {
            from protocol rip;
            then {
                community add VPN-A;
                accept;
            }
        }
        term b {
            then reject;
        }
    }
    community VPN-A members target:65535:00;
}
```

Router P1

Master Protocol Instance

```
protocols {
}
```

Enable RSVP

```
rsvp {
    interface so-1/0/1.0;
}
```

Enable LDP

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

Enable MPLS

```
mpls {
    label-switched-path P1-to-P3 {
        to 10.255.100.1;
        ldp-tunneling;
    }
    interface so-1/0/0.0;
    interface so-1/0/1.0;
}
```

Configure OSPF for Traffic Engineering Support

```
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface so-1/0/0.0;
        interface so-1/0/1.0;
    }
}
```

Router P2

Master Protocol Instance

```
protocols {
}
```

Enable RSVP

```
rsvp {
    interface so-1/1/0.0;
    interface at-2/0/0.0;
}
```

Enable MPLS

```
mpls {
    interface so-1/1/0.0;
    interface at-2/0/0.0;
}
```

Router P3

Master Protocol Instance

```
protocols {
}
```

Enable RSVP

```
rsvp {
```

```
    interface at-2/0/1.0;
}
```

Enable LDP

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

Enable MPLS

```
mpls {
    label-switched-path P3-to-P1 {
        to 10.255.2.2;
        ldp-tunneling;
    }
    interface at-2/0/1.0;
    interface so-0/0/0.0;
}
```

Configure OSPF for Traffic Engineering Support

```
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface at-2/0/1.0;
        interface at-2/0/1.0;
    }
}
```

Router PE2

Routing Instance for VPN-A

```
routing-instance {
    VPN-A {
        instance-type vrf;
        interface so-1/2/0.0;
        route-distinguisher 65535:1;
        vrf-import VPN-A-import;
        vrf-export VPN-A-export;
    }
}
```

Instance Routing Protocol

```
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-1/2/0.0;
        }
    }
}
```

Interfaces

```
interfaces {
    so-0/0/0 {
        unit 0 {
            family mpls;
        }
    }
    so-1/2/0 {
        unit 0;
```

```
    }
}
```

Master Protocol Instance

```
protocols {
}
```

Enable LDP

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

Enable MPLS

```
mpls {
    interface so-0/0/0.0;
    interface so-1/2/0.0;
}
```

Configure IBGP

```
bgp {
    group PE2-to-PE1 {
        type internal;
        local-address 10.255.200.2;
        family inet-vpn {
            unicast;
        }
        neighbor 10.255.1.1;
    }
}
```

Configure VPN Policy

```
policy-options {
    policy-statement VPN-A-import {
        term a {
            from {
                protocol bgp;
                community VPN-A;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    policy-statement VPN-A-export {
        term a {
            from protocol ospf;
            then {
                community add VPN-A;
                accept;
            }
        }
        term b {
            then reject;
        }
    }
    community VPN-A members target:65535:01;
}
```

Configuring an Application-Based Layer 3 VPN Topology

This example illustrates an application-based mechanism for forwarding traffic into a Layer 3 VPN. Typically, one or more interfaces are associated with, or bound to, a VPN by including them in the configuration of the VPN routing instance. By binding the interface to the VPN, the VPN's VRF table is used to make forwarding decisions for any incoming traffic on that interface. Binding the interface also includes the interface local routes in the VRF table, which provides next-hop resolution for VRF routes.

In this example, a firewall filter is used to define which incoming traffic on an interface is forwarded by means of the standard routing table, inet.0, and which incoming traffic is forwarded by means of the VRF table. You can expand this example such that incoming traffic on an interface can be redirected to one or more VPNs. For example, you can define a configuration to support a VPN that forwards traffic based on source address, that forwards Hypertext Transfer Protocol (HTTP) traffic, or that forwards only streaming media.

For this configuration to work, the following must be true:

- The interfaces that use filter-based forwarding must not be bound to the VPN.
- Static routing must be used as the means of routing.
- You must define an interface routing table group that is shared among inet.0 and the VRF tables to provide local routes to the VRF table.

This example consists of two client hosts (Client D and Client E) that are in two different VPNs and that want to send traffic both within the VPN and to the Internet. The paths are defined as follows:

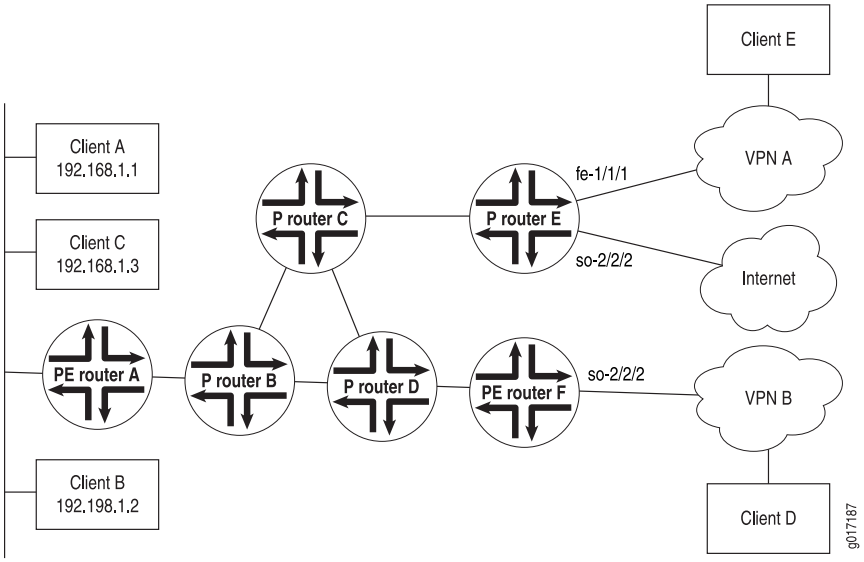
- Client A sends traffic to Client E over VPN A with a return path that also uses VPN A (using the VPN's VRF table).
- Client B sends traffic to Client D over VPN B with a return path that uses standard destination-based routing (using the inet.0 routing table).
- Clients B and C send traffic to the Internet using standard routing (using the inet.0 routing table), with a return path that also uses standard routing.

This example illustrates that there are a large variety of options in configuring an application-based Layer 3 VPN topology. This flexibility has application in many network implementations that require specific traffic to be forwarded in a constrained routing environment.

This configuration example shows only the portions of the configuration for the filter-based forwarding, routing instances, and policy. It does not illustrate how to configure a Layer 3 VPN.

Figure 27 on page 260 illustrates the network topology used in this example.

Figure 27: Application-Based Layer 3 VPN Example Configuration



Configuration on Router A

On Router A, you configure the interface to Clients A, B, and C. The configuration evaluates incoming traffic to determine whether it is to be forwarded by means of VPN or standard destination-based routing.

First, you apply an inbound filter and configure the interface:

```
[edit]
interfaces {
    fe-1/1/0 {
        unit 0 {
            family inet {
                filter {
                    input fbf-vrf;
                }
                address 192.168.1.1/24;
            }
        }
    }
}
```

Because the interfaces that use filter-based forwarding must not be bound to a VPN, you must configure an alternate method to provide next-hop routes to the VRF table. You do this by defining an interface routing table group and sharing this group among all the routing tables:

```
[edit]
routing-options {
    interface-routes {
        rib-group inet if-rib;
    }
    rib-groups {
```

```
        if-rib {
            import-rib [ inet.0 vpn-A.inet.0 vpn-B.inet.0 ];
        }
    }
}
```

You apply the following filter to incoming traffic on interface fe-1/1/0.0. The first term matches traffic from Client A and forwards it to the routing instance for VPN A. The second term matches traffic from Client B that is destined for Client D and forwards it to the routing instance for VPN B. The third term matches all other traffic, which is forwarded normally by means of destination-based forwarding according to the routes in inet.0.

```
[edit firewall family family-name]
filter fbf-vrf {
    term vpnA {
        from {
            source-address {
                192.168.1.1/32;
            }
        }
        then {
            routing-instance vpn-A;
        }
    }
    term vpnB {
        from {
            source-address {
                192.168.1.2/32;
            }
            destination-address {
                192.168.3.0/24;
            }
        }
        then routing-instance vpn-B;
    }
}
term internet {
    then accept;
}
```

You then configure the routing instances for VPN A and VPN B. Notice that these statements include all the required statements to define a Layer 3 VPN except for the interface statement.

```
[edit]
routing-instances {
    vpn-A {
        instance-type vrf;
        route-distinguisher 172.21.10.63:100;
        vrf-import vpn-A-import;
        vrf-export vpn-A-export;
    }
    vpn-B {
        instance-type vrf;
```

```
        route-distinguisher 172.21.10.63:200;
        vrf-import vpn-B-import;
        vrf-export vpn-B-export;
    }
}
```

Configuration on Router E

On Router E, configure a default route to reach the Internet. You should inject this route into the local IBGP mesh to provide an exit point from the network.

```
[edit]
routing-options {
    static {
        route 0.0.0.0/0 next-hop so-2/2/2.0 discard
    }
}
```

Configure the interface to Client E so that all incoming traffic on interface fe-1/1/1.0 that matches the VPN policy is forwarded over VPN A:

```
[edit]
routing-instances {
    vpn-A {
        interface fe-1/1/1.0
        instance-type vrf;
        route-distinguisher 172.21.10.62:100;
        vrf-import vpn-A-import;
        vrf-export vpn-A-export;
        routing-options {
            static {
                route 192.168.2.0/24 next-hop fe-1/1/1.0;
            }
        }
    }
}
```

Configuration on Router F

Again, because the interfaces that use filter-based forwarding must not be bound to a VPN, you configure an alternate method to provide next-hop routes to the VRF table by defining an interface routing table group and sharing this group among all the routing tables. To provide a route back to the clients for normal inet.0 routing, you define a static route to include in inet.0 and redistribute the static route into BGP:

```
[edit]
routing-options {
    interface-routes {
        rib-group inet if-rib;
    }
    rib-groups {
        if-rib {
            import-rib [ inet.0 vpn-B.inet.0 ];
        }
```

```
    }
}
```

To direct traffic from VPN B to Client D, you configure the routing instance for VPN B on Router F. All incoming traffic from Client D on interface so-3/3/3.0 is forwarded normally by means of the destination address based on the routes in inet.0.

```
[edit]
routing-instances {
    vpn-B {
        instance-type vrf;
        route-distinguisher 172.21.10.64:200;
        vrf-import vpn-B-import;
        vrf-export vpn-B-export;
        routing-options {
            static {
                route 192.168.3.0/24 next-hop so-3/3/3.0;
            }
        }
    }
}
```

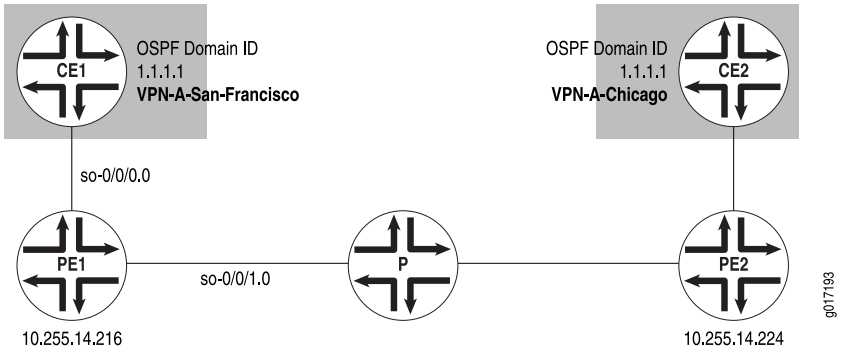
Configuring an OSPF Domain ID for a Layer 3 VPN

This example illustrates how to configure an OSPF domain ID for a VPN by using OSPF as the routing protocol between the PE and CE routers. Routes from an OSPF domain need an OSPF domain ID when they are distributed in BGP as VPN-IPv4 routes in VPNs with multiple OSPF domains. In a VPN connecting multiple OSPF domains, the routes from one domain might overlap with the routes of another.

For more information on OSPF domain IDs and Layer 3 VPNs, see “Configuring an OSPF Domain ID” on page 152.

Figure 28 on page 263 shows this example’s configuration topology. Only the configuration for Router PE1 is provided. The configuration for Router PE2 can be similar to the configuration for Router PE1. There are no special configuration requirements for the CE routers.

Figure 28: Example of a Configuration Using an OSPF Domain ID



For configuration information, see the following sections:

Configuring Interfaces on Router PE1

You need to configure two interfaces for Router PE1—the so-0/0/0 interface for traffic to Router CE1 (San Francisco) and the so-0/0/1 interface for traffic to a P router in the service provider's network.

Configure the interfaces for Router PE1:

```
[edit]
interfaces {
    so-0/0/0 {
        unit 0 {
            family inet {
                address 10.19.1.2/30;
            }
        }
    }
    so-0/0/1 {
        unit 0 {
            family inet {
                address 10.19.2.1/30;
            }
            family mpls;
        }
    }
}
```

Configuring Routing Options on Router PE1

At the [edit routing-options] hierarchy level, you need to configure the router-id and autonomous-system statements. The router-id statement identifies Router PE1.

Configure the routing options for Router PE1:

```
[edit]
routing-options {
    router-id 10.255.14.216;
    autonomous-system 69;
}
```

Configuring Protocols on Router PE1

On Router PE1, you need to configure MPLS, BGP, OSPF, and LDP at the [edit protocols] hierarchy level:

```
[edit]
protocols {
    mpls {
        interface so-0/0/1.0;
    }
    bgp {
        group San-Francisco-Chicago {
            type internal;
            preference 10;
            local-address 10.255.14.216;
            family inet-vpn {
                unicast;
            }
            neighbor 10.255.14.224;
        }
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface so-0/0/1.0;
        }
    }
    ldp {
        interface so-0/0/1.0;
    }
}
```

Configuring Policy Options on Router PE1

On Router PE1, you need to configure policies at the [edit policy-options] hierarchy level. These policies ensure that the CE routers in the Layer 3 VPN exchange routing information. In this example, Router CE1 in San Francisco exchanges routing information with Router CE2 in Chicago.

Configure the policy options on the PE1 router:

```
[edit]
policy-options {
    policy-statement vpn-import-VPN-A {
        term term1 {
            from {
                protocol bgp;
                community import-target-VPN-A;
            }
            then accept;
        }
        term term2 {
            then reject;
        }
    }
```

```
    policy-statement vpn-export-VPN-A {
        term term1 {
            from protocol ospf;
            then {
                community add export-target-VPN-A;
                accept;
            }
        }
        term term2 {
            then reject;
        }
    }
    community export-target-VPN-A members [target:10.255.14.216:11
    domain-id:1.1.1.1:0];
    community import-target-VPN-A members target:10.255.14.224:31;
}
```

Configuring the Routing Instance on Router PE1

You need to configure a Layer 3 VPN routing instance on Router PE1. To indicate that the routing instance is for a Layer 3 VPN, add the instance-type vrf statement at the [edit routing-instance *routing-instance-name*] hierarchy level.

The domain-id statement is configured at the [edit routing-instances routing-options protocols ospf] hierarchy level. As shown in Figure 28 on page 263, the routing instance on Router PE2 must share the same domain ID as the corresponding routing instance on Router PE1 so that routes from Router CE1 to Router CE2 and vice versa are distributed as Type 3 LSAs. If you configure different OSPF domain IDs in the routing instances for Router PE1 and Router PE2, the routes from each CE router will be distributed as Type 5 LSAs.

Configure the routing instance on Router PE1:

```
[edit]
routing-instances {
    VPN-A-San-Francisco-Chicago {
        instance-type vrf;
        interface so-0/0/0.0;
        route-distinguisher 10.255.14.216:11;
        vrf-import vpn-import-VPN-A;
        vrf-export vpn-export-VPN-A;
        routing-options {
            router-id 10.255.14.216;
            autonomous-system 69;
        }
        protocols {
            ospf {
                domain-id 1.1.1.1;
                export vpn-import-VPN-A;
                area 0.0.0.0 {
                    interface so-0/0/0.0;
                }
            }
        }
    }
```

```
}
```

Configuration Summary for Router PE1

Configure Interfaces

```
interfaces {
   so-0/0/0 {
      unit 0 {
         family inet {
            address 10.19.1.2/30;
         }
      }
   }
   so-0/0/1 {
      unit 0 {
         family inet {
            address 10.19.2.1/30;
         }
         family mpls;
      }
   }
}
```

Configure Routing Options

```
routing-options {
   router-id 10.255.14.216;
   autonomous-system 69;
}
```

Configure Protocols

```
protocols {
   mpls {
      interface so-0/0/0.0;
   }
   bgp {
      group San-Francisco-Chicago {
         type internal;
         preference 10;
         local-address 10.255.14.216;
         family inet-vpn {
            unicast;
         }
         neighbor 10.255.14.224;
      }
   }
   ospf {
      traffic-engineering;
      area 0.0.0.0 {
         interface so-0/0/1.0;
      }
   }
   ldp {
      interface so-0/0/1.0;
   }
}
```

Configure VPN Policy

```
policy-options {
   policy-statement vpn-import-VPN-A {
```

```
        term term1 {
            from {
                protocol bgp;
                community import-target-VPN-A;
            }
            then accept;
        }
        term term2 {
            then reject;
        }
    }
    policy-statement vpn-export-VPN-A {
        term term1 {
            from protocol ospf;
            then {
                community add export-target-VPN-A;
                accept;
            }
        }
        term term2 {
            then reject;
        }
    }
    community export-target-VPN-B members [
        target:10.255.14.216:11domain-id:1.1.1.1:0 ];
    community import-target-VPN-B members target:10.255.14.224:31;
}
```

Routing Instance for Layer 3 VPN

```
routing-instances {
    VPN-A-San-Francisco-Chicago {
        instance-type vrf;
        interface so-0/0/0.0;
        route-distinguisher 10.255.14.216:11;
        vrf-import vpn-import-VPN-A;
        vrf-export vpn-export-VPN-A;
        routing-options {
            router-id 10.255.14.216;
            autonomous-system 69;
        }
        protocols {
            ospf {
                domain-id 1.1.1.1;
                export vpn-import-VPN-A;
                area 0.0.0.0 {
                    interface so-0/0/0.0;
                }
            }
        }
    }
}
```

Configuring Overlapping VPNs Using Routing Table Groups

In Layer 3 VPNs, a CE router is often a member of more than one VPN. This example illustrates how to configure PE routers that support CE routers that support multiple VPNs. Support for this type of configuration uses a JUNOS software feature called routing table groups (sometimes also called routing information base [RIB] groups), which allows a route to be installed into several routing tables. A routing table group is a list of routing tables into which the protocol should install its routes.

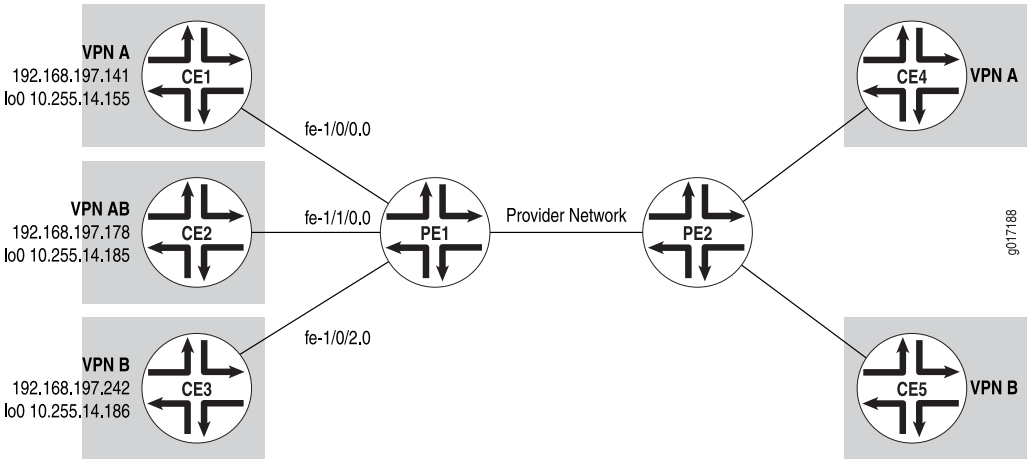
You define routing table groups at the [edit routing-options] hierarchy level for the default instance. You cannot configure routing table groups at the [edit routing-instances routing-options] hierarchy level; doing so results in a commit error.

After you define a routing table group, it can be used by multiple protocols. You can also apply routing table groups to static routing. The configuration examples in this section include both types of configurations.

Figure 29 on page 269 illustrates the topology for the configuration example in this section. The configurations in this section illustrate local connectivity between CE routers connected to the same PE router. If Router PE1 were connected only to Router CE2 (VPN AB), there would be no need for any extra configuration. The configuration statements in the sections that follow enable VPN AB Router CE2 to communicate with VPN A Router CE1 and VPN B Router CE3, which are directly connected to Router PE1. VPN routes that originate from the remote PE routers (the PE2 router in this case) are placed in a global Layer 3 VPN routing table (bgp.l3vpn.inet.0), and routes with appropriate route targets are imported into the routing tables as dictated by the VRF import policy configuration. The goal is to be able to choose routes from individual VPN routing tables that are locally populated.

Router PE1 is where all the filtering and configuration modification takes place. Therefore only VPN configurations for PE1 are shown. The CE routers do not have any information about the VPN, so you can configure them normally.

Figure 29: Example of an Overlapping VPN Topology



The following sections explain several ways to configure overlapping VPNs. For all the examples that follow, you need to configure routing table groups as described in “Configuring Routing Table Groups” on page 270.

The following sections illustrate different scenarios for configuring overlapping VPNs, depending on the routing protocol used between the PE and CE routers. For all of these examples, you need to configure routing table groups.

Configuring Routing Table Groups

In this example, routing table groups are common in the four configuration scenarios. The routing table groups are used to install routes (including interface, static, OSPF, and BGP routes) into several routing tables for the default and other instances. In the routing table group definition, the first routing table is called the primary routing table. (Normally, the primary routing table is the table into which the route would be installed if you did not configure routing table groups. The other routing tables are called secondary routing tables.)

The routing table groups in this configuration install routes as follows:

- vpna-vpnab installs routes into routing tables VPN-A.inet.0 and VPN-AB.inet.0.
- vpnb-vpnab installs routes into routing tables VPN-B.inet.0 and VPN-AB.inet.0.
- vpnab-vpna_and_vpnb installs routes into routing tables VPN-AB.inet.0, VPN-A.inet.0, and VPN-B.inet.0.

Configure the routing table groups:

```
[edit]
routing-options {
    rib-groups {
        vpna-vpnab {
            import-rib [ VPN-A.inet.0 VPN-AB.inet.0 ];
        }
        vpnb-vpnab {
            import-rib [ VPN-B.inet.0 VPN-AB.inet.0 ];
        }
        vpnab-vpna_and_vpnb {
            import-rib [ VPN-AB.inet.0 VPN-A.inet.0 VPN-B.inet.0 ];
        }
    }
}
```

Configuring Static Routes Between the PE and CE Routers

To configure static routing between the PE1 router and the CE1, CE2, and CE3 routers, you must configure routing instances for VPN A, VPN B, and VPN AB (you configure static routing under each instance):

Configuring the Routing Instance for VPN A

On Router PE1, configure VPN A:

```
[edit]
routing-instances {
   VPN-A {
      instance-type vrf;
      interface fe-1/0/0.0;
      route-distinguisher 10.255.14.175:3;
      vrf-import vpna-import;
      vrf-export vpna-export;
      routing-options {
         interface-routes {
            rib-group inet vpna-vpnab;
         }
         static {
            route 10.255.14.155/32 next-hop 192.168.197.141;
            route 10.255.14.185/32 next-hop 192.168.197.178;
         }
      }
   }
}
```

The interface-routes statement installs VPN A's interface routes into the routing tables defined in the routing table group vpna-vpnab.

The static statement configures the static routes that are installed in the VPN-A.inet.0 routing table. The first static route is for Router CE1 (VPN A) and the second is for Router CE2 (in VPN AB).

Next-hop 192.168.197.178 is not in VPN A. Route 10.255.14.185/32 cannot be installed in VPN-A.inet.0 unless interface routes from routing instance VPN AB are installed in this routing table. Including the interface-routes statements in the VPN AB configuration provides this next hop. Similarly, including the interface-routes statement in the VPN AB configuration installs 192.168.197.141 into VPN-AB.inet.0.

Configuring the Routing Instance for VPN AB

On Router PE1, configure VPN AB:

```
[edit]
```

```
routing instances {
   VPN-AB {
      instance-type vrf;
      interface fe-1/1/0.0;
      route-distinguisher 10.255.14.175:9;
      vrf-import vpnab-import;
      vrf-export vpnab-export;
      routing-options {
         interface-routes {
            rib-group vpnab-vpna_and_vpnb;
         }
         static {
            route 10.255.14.185/32 next-hop 192.168.197.178;
            route 10.255.14.155/32 next-hop 192.168.197.141;
            route 10.255.14.186/32 next-hop 192.168.197.242;
         }
      }
   }
}
```

In this configuration, the following static routes are installed in the VPN-AB.inet.0 routing table:

- 10.255.14.185/32 is for Router CE2 (in VPN AB)
- 10.255.14.155/32 is for Router CE1 (in VPN A)
- 10.255.14.186/32 is for Router CE3 (in VPN B)

Next-hops 192.168.197.141 and 192.168.197.242 do not belong to VPN AB. Routes 10.255.14.155/32 and 10.255.14.186/32 cannot be installed in VPN-AB.inet.0 unless interface routes from VPN A and VPN B are installed in this routing table. The interface route configurations in VPN A and VPN B routing instances provide these next hops.

Configuring the Routing Instance for VPN B

On Router PE1, configure VPN B:

```
[edit]
routing instances {
   VPN-B {
      instance-type vrf;
      interface fe-1/0/2.0;
      route-distinguisher 10.255.14.175:10;
      vrf-import vpnb-import;
      vrf-export vpnb-export;
      routing-options {
         interface-routes {
            rib-group inet vpnb-vpnab;
         }
         static {
            route 10.255.14.186/32 next-hop 192.168.197.242;
            route 10.255.14.185/32 next-hop 192.168.197.178;
         }
      }
   }
```

```
}
```

When you configure the routing instance for VPN B, these static routes are placed in VPNB.inet.0:

- 10.255.14.186/32 is for Router CE3 (in VPN B)
- 10.255.14.185/32 is for Router CE2 (in VPN AB)

Next-hop 192.168.197.178 does not belong to VPN B. Route 10.255.14.185/32 cannot be installed in VPN-B.inet.0 unless interface routes from VPN AB are installed in this routing table. The interface route configuration in VPN AB provides this next hop.

Configuring VPN Policy

The vrf-import and vrf-export policy statements that you configure for overlapping VPNs are the same as policy statements for regular VPNs, except that you include the from interface statement in each VRF export policy. This statement forces each VPN to announce only those routes that originated from that VPN. For example, VPN A has routes that originated in VPN A and VPN AB. If you do not include the from interface statement, VPN A announces its own routes as well as VPN AB's routes, so the remote PE router receives multiple announcements for the same routes. Including the from interface statement restricts each VPN to announcing only the routes it originated and allows you to filter out the routes imported from other routing tables for local connectivity.

In this configuration example, the vpnab-import policy accepts routes from VPN A, VPN B, and VPN AB. The vpna-export policy exports only routes that originate in VPN A. Similarly, the vpnb-export and vpnab-export policies export only routes that originate within the respective VPNs.

On Router PE1, configure the following VPN import and export policies:

```
[edit]
policy-options {
    policy-statement vpna-import {
        term a {
            from {
                protocol bgp;
                community VPNA-comm;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    policy-statement vpnb-import {
        term a {
            from {
                protocol bgp;
                community VPNB-comm;
            }
```

```
        then accept;
    }
    term b {
        then reject;
    }
}
policy-statement vpnab-import {
    term a {
        from {
            protocol bgp;
            community [ VPNA-comm VPNB-comm ];
        }
        then accept;
    }
    term b {
        then reject;
    }
}
policy-statement vpna-export {
    term a {
        from {
            protocol static;
            interface fe-1/0/0.0;
        }
        then {
            community add VPNA-comm;
            accept;
        }
    }
    term b {
        then reject;
    }
}
policy-statement vpnb-export {
    term a {
        from {
            protocol static;
            interface fe-1/0/2.0;
        }
        then {
            community add VPNB-comm;
            accept;
        }
    }
    term b {
        then reject;
    }
}
policy-statement vpnab-export {
    term a {
        from {
            protocol static;
            interface fe-1/1/0.0;
        }
        then {
            community add VPNB-comm;
```

```
            community add VPNA-comm;
            accept;
        }
    }
    term b {
        then reject;
    }
}
community VPNA-comm members target:69:1;
community VPNB-comm members target:69:2;
}
```

On Router PE1, apply the VPN import and export policies:

```
[edit]
routing-instances {
    VPN-A {
        instance-type vrf;
        interface fe-1/0/0.0;
        route-distinguisher 10.255.14.175:3;
        vrf-import vpna-import;
        vrf-export vpna-export;
        routing-options {
            static {
                rib-group vpna-vpnab;
                route 10.255.14.155/32 next-hop 192.168.197.141;
                route 10.255.14.185/32 next-hop 192.168.197.178;
            }
        }
    }
    VPN-AB {
        instance-type vrf;
        interface fe-1/1/0.0;
        route-distinguisher 10.255.14.175:9;
        vrf-import vpnab-import;
        vrf-export vpnab-export;
        routing-options {
            static {
                rib-group vpnab-vpna_and_vpnb;
                route 10.255.14.185/32 next-hop 192.168.197.178;
            }
        }
    }
    VPN-B {
        instance-type vrf;
        interface fe-1/0/2.0;
        route-distinguisher 10.255.14.175:10;
        vrf-import vpnb-import;
        vrf-export vpnb-export;
        routing-options {
            static {
                rib-group vpnb-vpnab;
                route 10.255.14.186/32 next-hop 192.168.197.242;
            }
        }
    }
```

```
}
```

For VPN A, include the routing-options statement at the [edit routing-instances *routing-instance-name*] hierarchy level to install the static routes directly into the routing tables defined in the routing table group vpna-vpnab. For VPN AB, the configuration installs the static route directly into the routing tables defined in the routing table group vpnab-vpna and vpnab-vpnb. For VPN B the configuration installs the static route directly into the routing tables defined in the routing table group vpnb-vpnab.

Configuring BGP Between the PE and CE Routers

In this configuration example, the vpna-site1 BGP group for VPN A installs the routes learned from the BGP session into the routing tables defined in the vpna-vpnab routing table group. For VPN AB, the vpnab-site1 group installs the routes learned from the BGP session into the routing tables defined in the vpnab-vpna_and_vpnb routing table group. For VPN B, the vpnb-site1 group installs the routes learned from the BGP session into the routing tables defined in the vpnb-vpnab routing table group. Interface routes are not needed for this configuration.

The VRF import and export policies are similar to those defined in "Configuring Static Routes Between the PE and CE Routers" on page 271, except the export protocol is BGP instead of a static route. On all vrf-export policies, you use the from protocol bgp statement.

On Router PE1, configure BGP between the PE and CE routers:

```
[edit]
routing-instances {
    VPN-A {
        instance-type vrf;
        interface fe-1/0/0.0;
        route-distinguisher 10.255.14.175:3;
        vrf-import vpna-import;
        vrf-export vpna-export;
        protocols {
            bgp {
                group vpna-site1 {
                    family inet {
                        unicast {
                            rib-group vpna-vpnab;
                        }
                    }
                    peer-as 1;
                    neighbor 192.168.197.141;
                }
            }
        }
    }
    VPN-AB {
        instance-type vrf;
        interface fe-1/1/0.0;
        route-distinguisher 10.255.14.175:9;
        vrf-import vpnab-import;
        vrf-export vpnab-export;
```

```
        protocols {
            bgp {
                group vpnab-site1 {
                    family inet {
                        unicast {
                            rib-group vpnab-vpna_and_vpnb;
                        }
                    }
                    peer-as 9;
                    neighbor 192.168.197.178;
                }
            }
        }
    }
    VPN-B {
        instance-type vrf;
        interface fe-1/0/2.0;
        route-distinguisher 10.255.14.175:10;
        vrf-import vpnb-import;
        vrf-export vpnb-export;
        protocols {
            bgp {
                group vpnb-site1 {
                    family inet {
                        unicast {
                            rib-group vpnb-vpnab;
                        }
                    }
                    neighbor 192.168.197.242 {
                        peer-as 10;
                    }
                }
            }
        }
    }
}
```

Configuring OSPF Between the PE and CE Routers

In this configuration example, routes learned from the OSPF session for VPN A are installed into the routing tables defined in the vpna-vpnab routing table group. For VPN AB, routes learned from the OSPF session are installed into the routing tables defined in the vpnab-vpna_and_vpnb routing table group. For VPN B, routes learned from the OSPF session are installed into the routing tables defined in the vpnb-vpnab routing table group.

The VRF import and export policies are similar to those defined in "Configuring Static Routes Between the PE and CE Routers" on page 271 and "Configuring BGP Between the PE and CE Routers" on page 276, except the export protocol is OSPF instead of BGP or a static route. Therefore, on all vrf-export policies, you use the from protocol ospf statement instead of the from protocol <static | bgp> statement.

On Router PE1, configure OSPF between the PE and CE routers:

```
[edit]
```

```
routing-instances {
   VPN-A {
      instance-type vrf;
      interface fe-1/0/0.0;
      route-distinguisher 10.255.14.175:3;
      vrf-import vpna-import;
      vrf-export vpna-export;
      protocols {
         ospf {
            rib-group vpna-vpnab;
            export vpna-import;
            area 0.0.0.0 {
               interface fe-1/0/0.0;
            }
         }
      }
   }
   VPN-AB {
      instance-type vrf;
      interface fe-1/1/0.0;
      route-distinguisher 10.255.14.175:9;
      vrf-import vpnab-import;
      vrf-export vpnab-export;
      protocols {
         ospf {
            rib-group vpnab-vpna_and_vpnb;
            export vpnab-import;
            area 0.0.0.0 {
               interface fe-1/1/0.0;
            }
         }
      }
   }
   VPN-B {
      instance-type vrf;
      interface fe-1/0/2.0;
      route-distinguisher 10.255.14.175:10;
      vrf-import vpnb-import;
      vrf-export vpnb-export;
      protocols {
         ospf {
            rib-group vpnb-vpnab;
            export vpnb-import;
            area 0.0.0.0 {
               interface fe-1/0/2.0;
            }
         }
      }
   }
}
```

Configuring Static, BGP, and OSPF Routes Between PE and CE Routers

This section shows how to configure the routes between the PE and CE routers by using a combination of static routes, BGP, and OSPF:

- The connection between Router PE1 and Router CE1 uses static routing.
- The connection between Router PE1 and Router CE2 uses BGP.
- The connection between Router PE1 and Router CE3 uses OSPF.

Here, the configuration for VPN AB also includes a static route to CE1.

On Router PE1, configure a combination of static routing, BGP, and OSPF between the PE and CE routers:

```
[edit]
routing-instances {
    VPN-A {
        instance-type vrf;
        interface fe-1/0/0.0;
        route-distinguisher 10.255.14.175:3;
        vrf-import vpna-import;
        vrf-export vpna-export;
        routing-options {
            static {
                rib-group vpna-vpnab;
                route 10.255.14.155/32 next-hop 192.168.197.141;
            }
        }
    }
    VPN-AB {
        instance-type vrf;
        interface fe-1/1/0.0;
        route-distinguisher 10.255.14.175:9;
        vrf-import vpnab-import;
        vrf-export vpnab-export;
        protocols {
            bgp {
                group vpnab-site1 {
                    family inet {
                        unicast {
                            rib-group vpnab-vpna_and_vpnb;
                        }
                    }
                    export to-vpnab-site1;
                    peer-as 9;
                    neighbor 192.168.197.178;
                }
            }
        }
    }
    VPN-B {
        instance-type vrf;
        interface fe-1/0/2.0;
        route-distinguisher 10.255.14.175:10;
        vrf-import vpnb-import;
        vrf-export vpnb-export;
        protocols {
            ospf {
                rib-group vpnb-vpnab;
```

```
            export vpnb-import;
            area 0.0.0.1 {
               interface t3-0/3/3.0;
            }
         }
      }
   }
}
policy-options {
   policy-statement to-vpnab-site1 {
      term a {
         from protocol static;
         then accept;
      }
      term b {
         from protocol bgp;
         then accept;
      }
      term c {
         then reject;
      }
   }
}
```

Configuring Overlapping VPNs Using Automatic Route Export

A problem with multiple routing instances is how to export routes between routing instances. You can accomplish this in JUNOS software by configuring routing table groups for each routing instance that needs to export routes to other routing tables. For information on how to configure overlapping VPNs by using routing table groups, see “Configuring Overlapping VPNs Using Routing Table Groups” on page 269.

However, using routing table groups has limitations:

- Routing table group configuration is complex. You must define a unique routing table group for each routing instance that will export routes.
- You must also configure a unique routing table group for each protocol that will export routes.

To limit and sometimes eliminate the need to configure routing table groups in multiple routing instance topologies, you can use the functionality provided by the auto-export statement.

The auto-export statement is particularly useful for configuring overlapping VPNs—VPN configurations where more than one VRF routing instance lists the same community route target in its vrf-import policy. The auto-export statement finds out which routing tables to export routes from and import routes to by examining the existing policy configuration.

The auto-export statement automatically exports routes between the routing instances referencing a given route target community. When the auto-export statement is configured, a VRF target tree is constructed based on the vrf-import and vrf-export policies configured on the system. If a routing instance references a route target in

its vrf-import policy, the route target is added to the import list for the target. If it references a specific route target in its vrf-export policy, the route target is added to the export list for that target. Route targets where there is a single importer that matches a single exporter or with no importers or exporters are ignored.

Changes to routing tables that export route targets are tracked. When a route change occurs, the routing instance's vpn-export policy is applied to the route. If it is allowed, the route is imported to all the import tables (subject to the vrf-import policy) of the route targets set by the export policy.

The sections that follow describe how to configure overlapping VPNs by using the auto-export statement for inter-instance export in addition to routing table groups:

- Configuring Overlapping VPNs with BGP and Automatic Route Export on page 281
- Configuring Overlapping VPNs and Additional Tables on page 282
- Configuring Automatic Route Export for All VRF Instances on page 283

Configuring Overlapping VPNs with BGP and Automatic Route Export

The following example provides the configuration for an overlapping VPN where BGP is used between the PE and CE routers.

Configure routing instance VPN-A:

```
[edit]
routing-instances {
   VPN-A {
      instance-type vrf;
      interface fe-1/0/0.0;
      route-distinguisher 10.255.14.175:3;
      vrf-import vpna-import;
      vrf-export vpna-export;
      routing-options {
         auto-export;
      }
      protocols {
         bgp {
            group vpna-site1 {
               peer-as 1;
               neighbor 192.168.197.141;
            }
         }
      }
   }
}
```

Configure routing instance VPN-AB:

```
[edit]
routing-instances {
   VPN-AB {
      instance-type vrf;
      interface fe-1/1/0.0;
      route-distinguisher 10.255.14.175:9;
```

```
        vrf-import vpnab-import;
        vrf-export vpnab-export;
        routing-options {
            auto-export;
        }
        protocols {
            bgp {
                group vpnab-site1 {
                    peer-as 9;
                    neighbor 192.168.197.178;
                }
            }
        }
    }
}
```

For this configuration, the auto-export statement replaces the functionality that was provided by a routing table group configuration. However, sometimes additional configuration is required.

Since the vrf-import policy and the vrf-export policy from which the auto-export statement deduces the import and export matrix are configured on a per-instance basis, you must be able to enable or disable them for unicast and multicast, in case multicast network layer reachability information (NLRI) is configured.

Configuring Overlapping VPNs and Additional Tables

You might need to use the auto-export statement between overlapping VPNs but require that a subset of the routes learned from a VRF table be installed into the inet.0 table or in routing-instance.inet.2.

To support this type of scenario, where not all of the information needed is present in the vrf-import and vrf-export policies, you configure an additional list of routing tables by using an additional routing table group.

To add routes from VPN-A and VPN-AB to inet.0 in the example described in “Configuring Overlapping VPNs with BGP and Automatic Route Export” on page 281, you need to include the following additional configuration statements:

Configure the routing options:

```
[edit]
routing-options {
    rib-groups {
        inet-access {
            import-rib inet.0;
        }
    }
}
```

Configure routing instance VPN-A:

```
[edit]
routing-instances {
```

```
    VPN-A {
        routing-options {
            auto-export {
                family inet {
                    unicast {
                        rib-group inet-access;
                    }
                }
            }
        }
    }
}
```

Configure routing instance VPN-AB:

```
[edit]
routing-instances {
    VPN-AB {
        routing-options {
            auto-export {
                family inet {
                    unicast {
                        rib-group inet-access;
                    }
                }
            }
        }
    }
}
```

Routing table groups are used in this configuration differently from how they are generally used in JUNOS software. Routing table groups normally require that the exporting routing table be referenced as the primary import routing table in the routing table group. For this configuration, the restriction does not apply. The routing table group functions as an additional list of tables to which to export routes.

Configuring Automatic Route Export for All VRF Instances

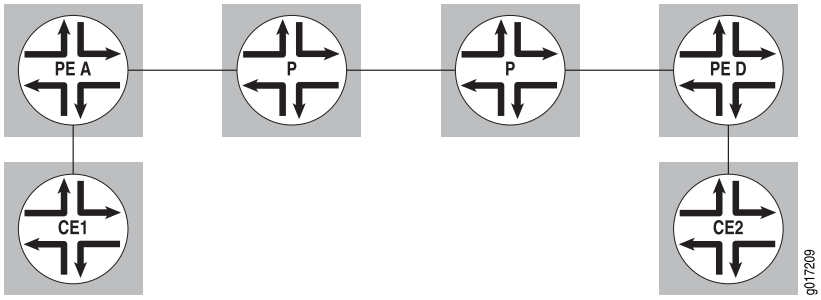
The following configuration allows you to configure the auto-export statement for all of the routing instances in a configuration group:

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

Configuring a GRE Tunnel Interface Between PE Routers

This example shows how to configure a generic routing encapsulation (GRE) tunnel interface between PE routers to provide VPN connectivity. You can use this configuration to tunnel VPN traffic across a non-MPLS core network. The network topology used in this example is shown in Figure 30 on page 284. The P routers shown in this illustration do not run MPLS.

Figure 30: PE Routers A and D Connected by a GRE Tunnel Interface



For configuration information, see the following sections:

Configuring the Routing Instance on Router A

Configure a routing instance on Router A:

```
[edit routing-instances]
gre-config {
    instance-type vrf;
    interface fe-1/0/0.0;
    route-distinguisher 10.255.14.176:69;
    vrf-import import-config;
    vrf-export export-config;
    protocols {
        ospf {
            export import-config;
            area 0.0.0.0 {
                interface all;
```

```
        }
      }
    }
}
```

Configuring the Routing Instance on Router D

Configure a routing instance on Router D:

```
[edit routing-instances]
gre-config {
    instance-type vrf;
    interface fe-1/0/1.0;
    route-distinguisher 10.255.14.178:69;
    vrf-import import-config;
    vrf-export export-config;
    protocols {
        ospf {
            export import-config;
            area 0.0.0.0 {
                interface all;
            }
        }
    }
}
```

Configuring MPLS, BGP, and OSPF on Router A

Although you do not need to configure MPLS on the P routers in this example, it is needed on the PE routers for the interface between the PE and CE routers and on the GRE interface (gr-1/1/0.0) linking the PE routers (Router A and Router D). Configure MPLS, BGP, and OSPF on Router A:

```
[edit protocols]
mpls {
    interface all;
}
bgp {
    group pe-to-pe {
        type internal;
        neighbor 10.255.14.178 {
            family inet-vpn {
                unicast;
            }
        }
    }
}
ospf {
    area 0.0.0.0 {
        interface all;
        interface gr-1/1/0.0 {
            disable;
        }
    }
```

```
}
```

Configuring MPLS, BGP, and OSPF on Router D

Although you do not need to configure MPLS on the P routers in this example, it is needed on the PE routers for the interface between the PE and CE routers and on the GRE interface (gr-1/1/0.0) linking the PE routers (Router D and Router A). Configure MPLS, BGP, and OSPF on Router D:

```
[edit protocols]
mpls {
    interface all;
}
bgp {
    group pe-to-pe {
        type internal;
        neighbor 10.255.14.176 {
            family inet-vpn {
                unicast;
            }
        }
    }
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface all;
        interface fxp0.0 {
            disable;
        }
        interface gr-1/1/0.0 {
            disable;
        }
    }
}
```

Configuring the Tunnel Interface on Router A

Configure the tunnel interface on Router A (the tunnel is unnumbered):

```
[edit interfaces interface-name]
unit 0 {
    tunnel {
        source 10.255.14.176;
        destination 10.255.14.178;
    }
    family inet;
    family mpls;
}
```

Configuring the Tunnel Interface on Router D

Configure the tunnel interface on Router D (the tunnel is unnumbered):

```
[edit interfaces interface-name]
unit 0 {
   tunnel {
      source 10.255.14.178;
      destination 10.255.14.176;
   }
   family inet;
   family mpls;
}
```

Configuring the Routing Options on Router A

As part of the routing options configuration for Router A, you need to configure routing table groups to enable VPN route resolution in the inet.3 routing table.

Configure the routing options on Router A:

```
[edit routing-options]
interface-routes {
   rib-group inet if-rib;
}
rib inet.3 {
   static {
      route 10.255.14.178/32 next-hop gr-1/1/0.0;
   }
}
rib-groups {
   if-rib {
      import-rib [ inet.0 inet.3 ];
   }
}
```

Configuring the Routing Options on Router D

As part of the routing options configuration for Router D, you need to configure routing table groups to enable VPN route resolution in the inet.3 routing table.

Configure the routing options on Router D:

```
[edit routing-options]
interface-routes {
   rib-group inet if-rib;
}
rib inet.3 {
   static {
      route 10.255.14.176/32 next-hop gr-1/1/0.0;
   }
}
rib-groups {
   if-rib {
      import-rib [ inet.0 inet.3 ];
   }
}
```

Configuration Summary for Router A

Configure the Routing Instance

```
gre-config {
    instance-type vrf;
    interface fe-1/0/0.0;
    route-distinguisher 10.255.14.176:69;
    vrf-import import-config;
    vrf-export export-config;
    protocols {
        ospf {
            export import-config;
            area 0.0.0.0 {
                interface all;
            }
        }
    }
}
```

Configure MPLS

```
mpls {
    interface all;
}
```

Configure BGP

```
bgp {
    traceoptions {
        file bgp.trace world-readable;
        flag update detail;
    }
    group pe-to-pe {
        type internal;
        neighbor 10.255.14.178 {
            family inet-vpn {
                unicast;
            }
        }
    }
}
```

Configure OSPF

```
ospf {
    area 0.0.0.0 {
        interface all;
        interface gr-1/1/0.0 {
            disable;
        }
    }
}
```

Configure the Tunnel Interface

```
interface-name {
    unit 0 {
        tunnel {
            source 10.255.14.176;
            destination 10.255.14.178;
        }
        family inet;
        family mpls;
    }
```

```
}
```

Configure Routing Options

```
interface-routes {
   rib-group inet if-rib;
}
rib inet.3 {
   static {
      route 10.255.14.178/32 next-hop gr-1/1/0.0;
   }
}
rib-groups {
   if-rib {
      import-rib [ inet.0 inet.3 ];
   }
}
```

Configuration Summary for Router D

Configure the Routing Instance

```
gre-config {
   instance-type vrf;
   interface fe-1/0/1.0;
   route-distinguisher 10.255.14.178:69;
   vrf-import import-config;
   vrf-export export-config;
   protocols {
      ospf {
         export import-config;
         area 0.0.0.0 {
            interface all;
         }
      }
   }
}
```

Configure MPLS

```
mpls {
   interface all;
}
```

Configure BGP

```
bgp {
   group pe-to-pe {
      type internal;
      neighbor 10.255.14.176 {
         family inet-vpn {
            unicast;
         }
      }
   }
}
```

Configure OSPF

```
ospf {
   traffic-engineering;
   area 0.0.0.0 {
      interface all;
```

```
        interface fxp0.0 {
            disable;
        }
        interface gr-1/1/0.0 {
            disable;
        }
    }
}
```

Configure the Tunnel Interface

```
interface-name {
    unit 0 {
        tunnel {
            source 10.255.14.178;
            destination 10.255.14.176;
        }
        family inet;
        family mpls;
    }
}
```

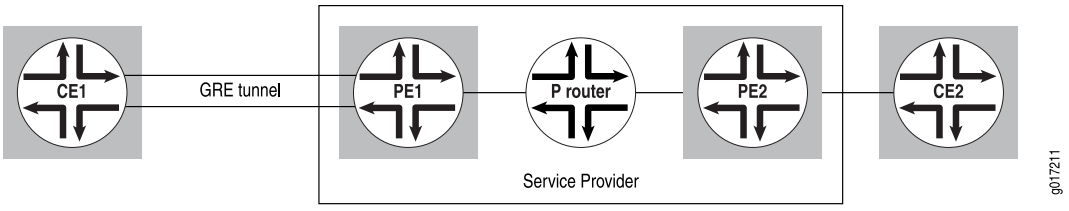
Configure the Routing Options

```
interface-routes {
    rib-group inet if-rib;
}
rib inet.3 {
    static {
        route 10.255.14.176/32 next-hop gr-1/1/0.0;
    }
}
rib-groups {
    if-rib {
        import-rib [ inet.0 inet.3 ];
    }
}
```

Configuring a GRE Tunnel Interface Between a PE and CE Router

This example shows how to configure a GRE tunnel interface between a PE router and a CE router. You can use this configuration to tunnel VPN traffic across a non-MPLS core network. The network topology used in this example is shown in Figure 31 on page 290.

Figure 31: GRE Tunnel Between the CE Router and the PE Router



For this example, complete the procedures described in the following sections:

- Configuring the Routing Instance Without the Encapsulating Interface on page 291
- Configuring the Routing Instance with the Encapsulating Interface on page 292
- Configuring the GRE Tunnel Interface on Router CE1 on page 293

Configuring the Routing Instance Without the Encapsulating Interface

You can configure the routing instance either with or without the encapsulating interface. The following sections explain how to configure the routing instance without it:

- Configuring the Routing Instance on Router PE1 on page 291
- Configuring the GRE Tunnel Interface on Router PE1 on page 291
- Configuring the Encapsulation Interface on Router PE1 on page 292

Configuring the Routing Instance on Router PE1

Configure the routing instance on Router PE1:

```
[edit routing-instances]
vpna {
    instance-type vrf;
    interface gr-1/2/0.0;
    route-distinguisher 10.255.14.174:1;
    vrf-import vpna-import;
    vrf-export vpna-export;
    protocols {
        bgp {
            group vpna {
                type external;
                peer-as 100;
                as-override;
                neighbor 10.49.2.1;
            }
        }
    }
}
```

Configuring the GRE Tunnel Interface on Router PE1

Configure the GRE tunnel interface on Router PE1:

```
[edit interfaces gr-1/2/0]
unit 0 {
    tunnel {
        source 192.168.197.249;
        destination 192.168.197.250;
    }
    family inet {
        address 10.49.2.2/30;
    }
}
```

In this example, interface t3-0/1/3 acts as the encapsulating interface for the GRE tunnel.

Configuring the Encapsulation Interface on Router PE1

Configure the encapsulation interface on Router PE1:

```
[edit interfaces t3-0/1/3]
unit 0 {
   family inet {
      address 192.168.197.249/30;
   }
}
```

Configuring the Routing Instance with the Encapsulating Interface

If the tunnel-encapsulating interface, t3-0/1/3, is also configured under the routing instance, then you need to specify the name of that routing instance under the interface definition. The system uses this routing instance to search for the tunnel destination address.

To configure the routing instance with the encapsulating interface, you perform the steps in the following sections:

- Configuring the Routing Instance on Router PE1 on page 292
- Configuring the GRE Tunnel Interface on Router PE1 on page 293
- Configuring the Encapsulation Interface on Router PE1 on page 293

Configuring the Routing Instance on Router PE1

If you configure the tunnel-encapsulating interface under the routing instance, then configure the routing instance on Router PE1:

```
[edit routing-instances]
vpna {
   instance-type vrf;
   interface gr-1/2/0.0;
   interface t3-0/1/3.0;
   route-distinguisher 10.255.14.174:1;
   vrf-import vpna-import;
   vrf-export vpna-export;
   protocols {
      bgp {
         group vpna {
            type external;
            peer-as 100;
            as-override;
            neighbor 10.49.2.1;
         }
      }
   }
}
```

Configuring the GRE Tunnel Interface on Router PE1

Configure the GRE tunnel interface on Router PE1:

```
[edit interfaces gr-1/2/0]
unit 0 {
    tunnel {
        source 192.168.197.249;
        destination 192.168.197.250;
        routing-instance {
            destination vpna;
        }
    }
    family inet {
        address 10.49.2.2/30;
    }
}
```

Configuring the Encapsulation Interface on Router PE1

Configure the encapsulation interface on Router PE1:

```
[edit interfaces t3-0/1/3]
unit 0 {
    family inet {
        address 192.168.197.249/30;
    }
}
```

Configuring the GRE Tunnel Interface on Router CE1

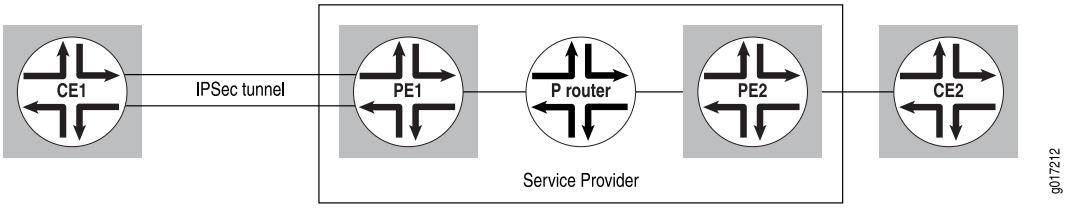
Configure the GRE tunnel interface on Router CE1:

```
[edit interfaces gr-1/2/0]
unit 0 {
    tunnel {
        source 192.168.197.250;
        destination 192.168.197.249;
    }
    family inet {
        address 10.49.2.1/30;
    }
}
```

Configuring an ES Tunnel Interface Between a PE and CE Router

This example shows how to configure an ES tunnel interface between a PE router and a CE router in a Layer 3 VPN. The network topology used in this example is shown in Figure 32 on page 294.

Figure 32: ES Tunnel Interface (IPSec Tunnel)



To configure this example, you perform the steps in the following sections:

- Configuring IPSec on Router PE1 on page 294
- Configuring the Routing Instance Without the Encapsulating Interface on page 295
- Configuring the Routing Instance with the Encapsulating Interface on page 296
- Configuring the ES Tunnel Interface on Router CE1 on page 297
- Configuring IPSec on Router CE1 on page 297

Configuring IPSec on Router PE1

Configure IP Security (IPSec) on Router PE1:

```
[edit security]
ipsec {
    security-association sa-esp-manual {
        mode tunnel;
        manual {
            direction bidirectional {
                protocol esp;
                spi 16000;
                authentication {
                    algorithm hmac-md5-96;
                    key ascii-text
                        "$9$ABULt1heK87dsWLDk.P3nrevM7V24ZHkPaZ/tp0cSvWLNwgZUH";
                }
                encryption {
                    algorithm des-cbc;
                    key ascii-text "$9$/H8Q90IyrvL7VKMZjHqQzcyleLN";
                }
            }
        }
    }
}
```

Configuring the Routing Instance Without the Encapsulating Interface

You can configure the routing instance on Router PE1 with or without the encapsulating interface (t3-0/1/3 in this example). The following sections explain how to configure the routing instance without it:

- Configuring the Routing Instance on Router PE1 on page 295
- Configuring the ES Tunnel Interface on Router PE1 on page 295
- Configuring the Encapsulating Interface for the ES Tunnel on page 295

Configuring the Routing Instance on Router PE1

Configure the routing instance on Router PE1:

```
[edit routing-instances]
vpna {
    instance-type vrf;
    interface es-1/2/0.0;
    route-distinguisher 10.255.14.174:1;
    vrf-import vpna-import;
    vrf-export vpna-export;
    protocols {
        bgp {
            group vpna {
                type external;
                peer-as 100;
                as-override;
                neighbor 10.49.2.1;
            }
        }
    }
}
```

Configuring the ES Tunnel Interface on Router PE1

Configure the ES tunnel interface on Router PE1:

```
[edit interfaces es-1/2/0]
unit 0 {
    tunnel {
        source 192.168.197.249;
        destination 192.168.197.250;
    }
    family inet {
        address 10.49.2.2/30;
        ipsec-sa sa-esp-manual;
    }
}
```

Configuring the Encapsulating Interface for the ES Tunnel

For this example, interface t3-0/1/3 is the encapsulating interface for the ES tunnel. Configure interface t3-0/1/3:

```
[edit interfaces t3-0/1/3]
unit 0 {
   family inet {
      address 192.168.197.249/30;
   }
}
```

Configuring the Routing Instance with the Encapsulating Interface

If the tunnel-encapsulating interface, t3-0/1/3, is also configured under the routing instance, you need to specify the routing instance name under the interface definition. The system uses this routing instance to search for the tunnel destination address for the IPSec tunnel using manual security association.

The following sections explain how to configure the routing instance with the encapsulating interface:

- Configuring the Routing Instance on Router PE1 on page 296
- Configuring the ES Tunnel Interface on Router PE1 on page 296
- Configuring the Encapsulating Interface on Router PE1 on page 297

Configuring the Routing Instance on Router PE1

Configure the routing instance on Router PE1 (including the tunnel encapsulating interface):

```
[edit routing-instances]
vpna {
   instance-type vrf;
   interface es-1/2/0.0;
   interface t3-0/1/3.0;
   route-distinguisher 10.255.14.174:1;
   vrf-import vpna-import;
   vrf-export vpna-export;
   protocols {
      bgp {
         group vpna {
            type external;
            peer-as 100;
            as-override;
            neighbor 10.49.2.1;
         }
      }
   }
}
```

Configuring the ES Tunnel Interface on Router PE1

Configure the ES tunnel interface on Router PE1:

```
[edit interfaces es-1/2/0]
unit 0 {
   tunnel {
```

```
        source 192.168.197.249;
        destination 192.168.197.250;
        routing-instance {
            destination vpna;
        }
    }
    family inet {
        address 10.49.2.2/30;
        ipsec-sa sa-esp-manual;
    }
}
```

Configuring the Encapsulating Interface on Router PE1

Configure the encapsulating interface on Router PE1:

```
[edit interfaces t3-0/1/3]
unit 0 {
    family inet {
        address 192.168.197.249/30;
    }
}
```

Configuring the ES Tunnel Interface on Router CE1

Configure the ES tunnel interface on Router CE1:

```
[edit interfaces es-1/2/0]
unit 0 {
    tunnel {
        source 192.168.197.250;
        destination 192.168.197.249;
    }
    family inet {
        address 10.49.2.1/30;
        ipsec-sa sa-esp-manual;
    }
}
```

Configuring IPSec on Router CE1

Configure IPSec on Router CE1:

```
[edit security]
ipsec {
    security-association sa-esp-manual {
        mode tunnel;
        manual {
            direction bidirectional {
                protocol esp;
                spi 16000;
                authentication {
                    algorithm hmac-md5-96;
```

```
                key ascii-text
                    "$9$ABULt1heK87dsWLDk.P3nrevM7V24ZHkPaZ/tp0cSvWLNwgZUH";
            }
            encryption {
                algorithm des-cbc;
                key ascii-text "$9$/H8Q90IyrvL7VKMZjHqQzcyIeLN";
            }
        }
    }
  }
}
```

Chapter 13

Layer 3 VPN Internet Access Examples

JUNOS software supports Internet access from a Layer 3 virtual private network (VPN). This chapter provides examples that demonstrate how to configure a provider edge (PE) router to provide Internet access to customer edge (CE) routers in a VPN. The method you use depends on the needs and specifications of the individual network. To provide Internet access through a Layer 3 VPN, you need to configure policies on the PE router. You also need to configure the next-table statement at the [edit routing-instances *routing-instance-name* routing-options static route] hierarchy level. When configured, this statement can point a default route from the VPN table (routing instance) to the main routing table (default instance) inet.0. The main routing table stores all Internet routes and is where final route resolution occurs.

There are several ways to configure a PE router to provide CE routers access to the Internet. These types of access are described in the following sections:

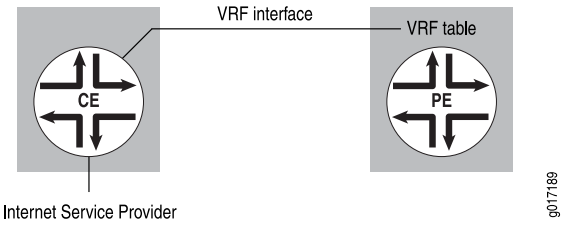
Non-VRF Internet Access

The following sections describe ways to provide Internet access to a CE router in a Layer 3 VPN without using the VPN routing and forwarding (VRF) interface. Because these methods effectively bypass the Layer 3 VPN, they are not discussed in detail.

CE Router Accesses Internet Independently of the PE Router

In this configuration, the PE router does not provide the Internet access. The CE router sends Internet traffic either to another service provider, or to the same service provider but a different router. The PE router handles Layer 3 VPN traffic only (see Figure 33 on page 300).

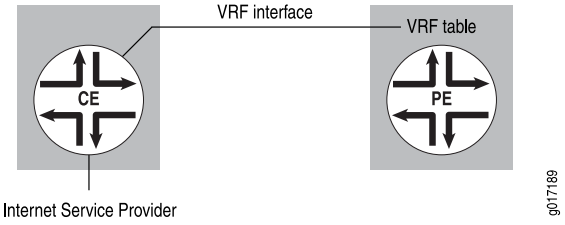
Figure 33: PE Router Does Not Provide Internet Access



PE Router Provides Layer 2 Internet Service

In this configuration, the PE router acts as a Layer 2 device, providing a Layer 2 connection (such as circuit cross-connect [CCC]) to another router that has a full set of Internet routes. The CE router can use just one physical interface and two logical interfaces to the PE router, or it can use multiple physical interfaces to the PE router (see Figure 34 on page 300).

Figure 34: PE Router Connects to a Router Connected to the Internet



Distributed Internet Access

In this scenario, the PE routers provide Internet access to the CE routers. In the examples that follow, it is assumed that the Internet routes (or defaults) are present in the inet.0 table of the PE routers that provide Internet access to selected CE routers.

When accessing the Internet from a VPN, Network Address Translation (NAT) must be performed between the VPN's private addresses and the public addresses used on the Internet unless the VPN is using the public address space. This section includes several examples of how to provide Internet access for VPNs, most of which require that the CE routers perform the address translation. The "Routing Internet Traffic Through a Separate NAT Device" on page 316 example, however, requires that the service provider supply the NAT functionality using a NAT device connected to the PE router.

In all of the examples, the VPN's public IP address pool (whose entries correspond to the translated private addresses) must be added to the inet.0 table and propagated to the Internet routers to receive reverse traffic from public destinations.

This section includes the following examples:

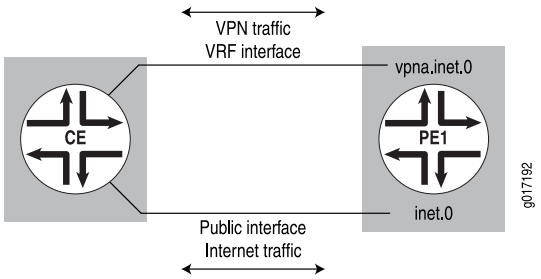
- Routing VPN and Internet Traffic Through Different Interfaces on page 301
- Routing VPN and Outgoing Internet Traffic Through the Same Interface and Routing Return Internet Traffic Through a Different Interface on page 307

- Routing VPN and Internet Traffic Through the Same Interface Bidirectionally (VPN Has Public Addresses) on page 309
- Routing VPN and Internet Traffic Through the Same Interface Bidirectionally (VPN Has Private Addresses) on page 312
- Routing Internet Traffic Through a Separate NAT Device on page 316

Routing VPN and Internet Traffic Through Different Interfaces

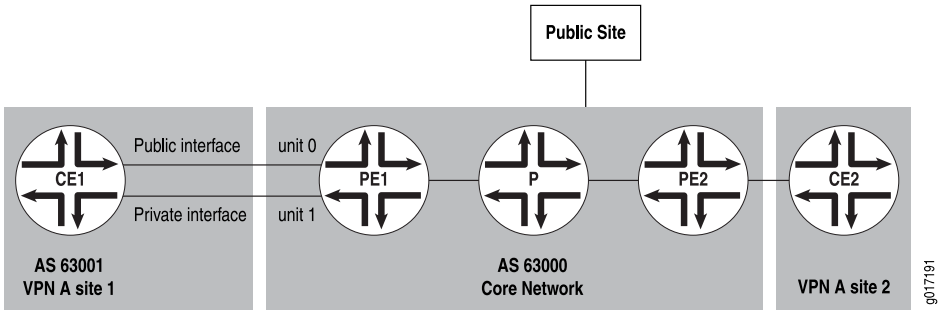
In this example, VPN and Internet traffic are routed through different interfaces. The CE router sends the VPN traffic through the VPN interface and sends the Internet traffic through a separate interface that is part of the main routing table on Router PE1 (the CE router can use either one physical interface with two logical units or two physical interfaces). NAT also occurs on the CE router (see Figure 35 on page 301).

Figure 35: Routing VPN and Internet Traffic Through Different Interfaces



The PE router is configured to install and advertise the public IP address pool for the VPN to other core routers (for return traffic). The VPN traffic is routed normally. Figure 36 on page 301 illustrates the PE router's VPN configuration.

Figure 36: Example of Internet Traffic Routed Through Separate Interfaces



The configuration in this example has the following features:

- Router PE1 uses two logical interfaces to connect to Router CE1 using Frame Relay encapsulation.
- The routing protocol between Router PE1 and Router CE1 is the external BGP (EBGP).

- Router CE1's public IP address pool is 10.12.1.1 through 10.12.1.254 (10.12.1.0/24).
- The next-hop-self setting is derived from the fix-nh policy statement on Router PE1. PE routers are forced to use next-hop-self so that next-hop resolution is done only for the PE router's loopback address for non-VPN routes (by default, VPN–Internet Protocol version 4 [IPv4] routes are sent by means of next-hop-self).

You can configure Router CE1 with a static default route pointing to its public interface for everything else.

The following sections show how to route VPN and Internet traffic through different interfaces:

- Configuring Interfaces on Router PE1 on page 302
- Configuring Routing Options on Router PE1 on page 303
- Configuring BGP, IS-IS, and LDP Protocols on Router PE1 on page 303
- Configuring a Routing Instance on Router PE1 on page 304
- Configuring Policy Options on Router PE1 on page 304
- Traffic Routed by Different Interfaces: Configuration Summarized by Router on page 305

Configuring Interfaces on Router PE1

Configure an interface to handle VPN traffic and an interface to handle Internet traffic:

```
[edit]
interfaces {
   t3-0/2/0 {
      dce;
      encapsulation frame-relay;
      unit 0 {
         description "to CE1 VPN interface";
         dlci 10;
         family inet {
            address 192.168.197.13/30;
         }
      }
      unit 1 {
         description "to CE1 public interface";
         dlci 20;
         family inet {
            address 192.168.198.201/30;
         }
      }
   }
}
```

Configuring Routing Options on Router PE1

Configure a static route on Router PE1 to install a route to the CE router's public IP address pool in inet.0:

```
[edit]
routing-options {
   static {
      route 10.12.1.0/24 next-hop 192.168.198.202;
   }
}
```

Configuring BGP, IS-IS, and LDP Protocols on Router PE1

Configure BGP on Router PE1 to allow non-VPN and VPN peering and to advertise the VPN's public IP address pool:

```
[edit]
protocols {
   bgp {
      group pe-pe {
         type internal;
         local-address 10.255.14.171;
         family inet {
            any;
         }
         family inet-vpn {
            any;
         }
         export [fix-nh redist-static];
         neighbor 10.255.14.177;
         neighbor 10.255.14.179;
      }
   }
}
```

Configure Intermediate System-to-Intermediate System (IS-IS) on Router PE1 to allow access to internal routes:

```
[edit protocols]
isis {
   level 1 disable;
   interface so-0/0/0.0;
   interface lo0.0;
}
```

Configure Label Distribution Protocol (LDP) on Router PE1 to tunnel VPN routes:

```
[edit protocols]
ldp {
   interface so-0/0/0.0;
}
```

Configuring a Routing Instance on Router PE1

Configure a routing instance on Router PE1:

```
[edit]
routing-instances {
   vpna {
      instance-type vrf;
      interface t3-0/2/0.0;
      route-distinguisher 10.255.14.171:100;
      vrf-import vpna-import;
      vrf-export vpna-export;
      protocols {
         bgp {
            group to-CE1 {
               peer-as 63001;
               neighbor 192.168.197.14;
            }
         }
      }
   }
}
```

Configuring Policy Options on Router PE1

You need to configure policy options on Router PE1. The fix-nh policy statement sets next-hop-self for all non-VPN routes:

```
[edit]
policy-options {
   policy-statement fix-nh {
      then {
         next-hop self;
      }
   }
}
```

The redist-static policy statement advertises the VPN's public IP address pool:

```
[edit policy-options]
policy-statement redist-static {
   term a {
      from {
         protocol static;
         route-filter 10.12.1.0/24 exact;
      }
      then accept;
   }
   term b {
      then reject;
   }
}
```

Configure import and export policies for vpna:

```
[edit policy-options]
policy-statement vpna-import {
   term a {
      from {
         protocol bgp;
         community vpna-comm;
      }
      then accept;
   }
   term b {
      then reject;
   }
}
policy-statement vpna-export {
   term a {
      from protocol bgp;
      then {
         community add vpna-comm;
         accept;
      }
   }
   term b {
      then reject;
   }
}
community vpna-comm members target:63000:100;
```

Traffic Routed by Different Interfaces: Configuration Summarized by Router

Router PE1

Interfaces

```
interfaces {
   t3-0/2/0 {
      dce;
      encapsulation frame-relay;
      unit 0 {
         description "to CE1 VPN interface";
         dlci 10;
         family inet {
            address 192.168.197.13/30;
         }
      }
      unit 1 {
         description "to CE1 public interface";
         dlci 20;
         family inet {
            address 192.168.198.201/30;
         }
      }
   }
}
```

Routing Options

```
routing-options {
   static {
      route 10.12.1.0/24 next-hop 192.168.198.202;
```

```
    }
}
```

BGP Protocol

```
protocols {
    bgp {
        group pe-pe {
            type internal;
            local-address 10.255.14.171;
            family inet {
                any;
            }
            family inet-vpn {
                any;
            }
            export [ fix-nh redist-static];
            neighbor 10.255.14.177;
            neighbor 10.255.14.179;
        }
    }
}
```

IS-IS Protocol

```
isis {
    level 1 disable;
    interface so-0/0/0.0;
    interface lo0.0;
}
```

LDP Protocol

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

Routing Instance

```
routing-instances {
    vpna {
        instance-type vrf;
        interface t3-0/2/0.0;
        route-distinguisher 10.255.14.171:100;
        vrf-import vpna-import;
        vrf-export vpna-export;
        protocols {
            bgp {
                group to-CE1 {
                    peer-as 63001;
                    neighbor 192.168.197.14;
                }
            }
        }
    }
}
```

Policy Options/Policy Statements

```
policy-options {
    policy-statement fix-nh {
        then {
            next-hop self;
        }
```

```
    }
    policy-statement redist-static {
        term a {
            from {
                protocol static;
                route-filter 10.12.1.0/24 exact;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
}
```

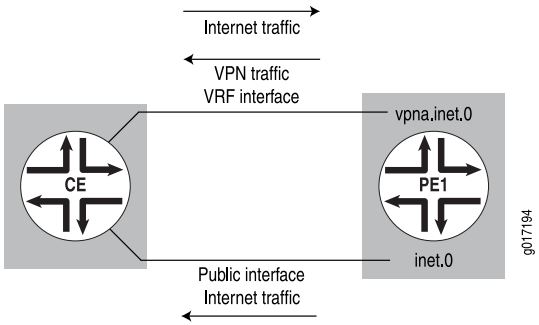
Import and Export Policies

```
policy-statement vpna-import {
    term a {
        from {
            protocol bgp;
            community vpna-comm;
        }
        then accept;
    }
    term b {
        then reject;
    }
}
policy-statement vpna-export {
    term a {
        from protocol bgp;
        then {
            community add vpna-comm;
            accept;
        }
    }
    term b {
        then reject;
    }
}
community vpna-comm members target:63000:100;
```

Routing VPN and Outgoing Internet Traffic Through the Same Interface and Routing Return Internet Traffic Through a Different Interface

In this example, the CE router sends VPN and Internet traffic through the same interface but receives return Internet traffic through a different interface. The PE router has a default route in the VRF table pointing to the main routing table inet.0. It routes the VPN public IP address pool (return Internet traffic) through a different interface in inet.0 (see Figure 37 on page 308). The CE router still performs NAT functions.

Figure 37: VPN and Outgoing Internet Traffic Routed Through the Same Interface and Return Internet Traffic Routed Through a Different Interface



Configuration for Router PE1

This example has the same configuration as Router PE1 in "Routing VPN and Internet Traffic Through Different Interfaces" on page 301. It uses the topology shown in Figure 36 on page 301. The default route to the VPN routing table is configured differently. At the [edit routing-instances *routing-instance-name* routing-options] hierarchy level, you configure a default static route that is installed in vpna.inet.0 and points to inet.0 for resolution:

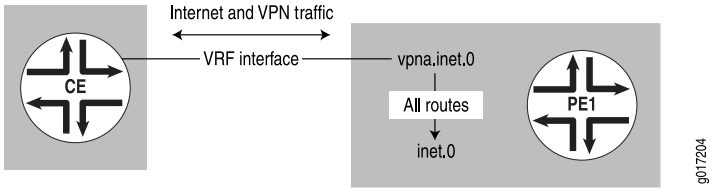
```
[edit]
routing-instances {
   vpna {
      instance-type vrf;
      interface t3-0/2/0.0;
      route-distinguisher 10.255.14.171:100;
      vrf-import vpna-import;
      vrf-export vpna-export;
      routing-options {
         static {
            route 0.0.0.0/0 next-table inet.0;
         }
      }
      protocols {
         bgp {
            group to-CE1 {
               peer-as 63001;
               neighbor 192.168.197.14;
            }
         }
      }
   }
}
```

You also need to change the configuration of Router CE1 (from the configuration that works with the configuration for Router PE1 described in "Routing VPN and Internet Traffic Through Different Interfaces" on page 301) to account for the differences in the configuration of the PE routers.

Routing VPN and Internet Traffic Through the Same Interface Bidirectionally (VPN Has Public Addresses)

This section shows how to configure a single logical interface to handle VPN and Internet traffic traveling both to and from the Internet and the CE router. This interface can handle both VPN and Internet traffic as long as there are no private addresses in the VPN. The VPN routes received from the CE router are added to the main routing table inet.0 by means of routing table groups. This allows the PE router to attract the return traffic from the Internet (see Figure 38 on page 309).

Figure 38: Interface Configured to Carry Both Internet and VPN Traffic



In this example, the CE router does not need to perform NAT, because all the VPN routes are public. The CE router has a single interface to the PE router, to which it advertises VPN routes. The PE router has a default route in the VRF table pointing to the main routing table inet.0. The PE router also imports VPN routes received from the CE router into inet.0 by means of routing table groups.

The following configuration for Router PE1 uses the same topology as in "Routing VPN and Internet Traffic Through Different Interfaces" on page 301. This configuration uses a single logical interface (instead of two) between Router PE1 and Router CE1.

The following sections show how to route VPN and Internet traffic through the same interface bidirectionally (VPN has public addresses):

- Configuring Routing Options on Router PE1 on page 309
- Configuring Routing Protocols on Router PE1 on page 310
- Configuring the Routing Instance on Router PE1 on page 310
- Traffic Routed Through the Same Interface Bidirectionally: Configuration Summarized by Router on page 311

Configuring Routing Options on Router PE1

Configure a routing table group definition for installing VPN routes in routing table groups vpna.inet.0 and inet.0:

```
[edit]
routing-options {
    rib-groups {
        vpna-to-inet0 {
            import-rib [ vpna.inet.0 inet.0 ];
        }
    }
}
```

Configuring Routing Protocols on Router PE1

Configure the Multiprotocol Label Switching (MPLS), BGP, IS-IS, and LDP protocols on Router PE1. This configuration does not include the policy redist-static statement at the [edit protocols bgp group pe-pe] hierarchy level. The VPN routes are sent directly to IBGP.

Configure BGP on Router PE1 to allow non-VPN and VPN peering, and to advertise the VPN's public IP address pool:

```
[edit]
protocols {
    mpls {
        interface t3-0/2/0.0;
    }
    bgp {
        group pe-pe {
            type internal;
            local-address 10.255.14.171;
            family inet {
                any;
            }
            family inet-vpn {
                any;
            }
            export fix-nh;
            neighbor 10.255.14.177;
            neighbor 10.255.14.173;
        }
    }
    isis {
        level 1 disable;
        interface so-0/0/0.0;
        interface lo0.0;
    }
    ldp {
        interface so-0/0/0.0;
    }
}
```

Configuring the Routing Instance on Router PE1

This section describes how to configure the routing instance on Router PE1. The static route defined in the routing-options statement directs Internet traffic from the CE router to the inet.0 routing table. The routing table group defined by the rib-group vpna-to-inet0 statement adds the VPN routes to inet.0.

Configure the routing instance on Router PE1:

```
[edit]
routing-instances {
    vpna {
        instance-type vrf;
        interface t3-0/2/0.0;
        route-distinguisher 10.255.14.171:100;
```

```
        vrf-import vpna-import;
        vrf-export vpna-export;
        routing-options {
            static {
                route 0.0.0.0/0 next-table inet.0;
            }
        }
        protocols {
            bgp {
                group to-CE1 {
                    family inet {
                        unicast {
                            rib-group vpna-to-inet0;
                        }
                    }
                    peer-as 63001;
                    neighbor 192.168.197.14;
                }
            }
        }
    }
}
```

You must configure Router CE1 to forward all traffic to Router PE1 using a default route. Alternatively, the default route can be advertised from Router PE1 to Router CE1 with EBGP.

Traffic Routed Through the Same Interface Bidirectionally: Configuration Summarized by Router

Router PE1

This example uses the same configuration as in "Routing VPN and Internet Traffic Through Different Interfaces" on page 301. This configuration uses a single logical interface (instead of two) between Router PE1 and Router CE1.

Routing Options

```
routing-options {
    rib-groups {
        vpna-to-inet0 {
            import-rib [ vpna.inet.0 inet.0 ];
        }
    }
}
```

Routing Protocols

```
protocols {
    mpls {
        interface t3-0/2/0.0;
    }
    bgp {
        group pe-pe {
            type internal;
            local-address 10.255.14.171;
            family inet {
                any;
            }
```

```
            family inet-vpn {
               any;
            }
            export fix-nh;
            neighbor 10.255.14.177;
            neighbor 10.255.14.173;
         }
      }
      isis {
         level 1 disable;
         interface so-0/0/0.0;
         interface lo0.0;
      }
      ldp {
         interface so-0/0/0.0;
      }
   }
```

Routing Instance

```
routing-instances {
   vpna {
      instance-type vrf;
      interface t3-0/2/0.0;
      route-distinguisher 10.255.14.171:100;
      vrf-import vpna-import;
      vrf-export vpna-export;
      routing-options {
         static {
            route 0.0.0.0/0 next-table inet.0;
         }
      }
      protocols {
         bgp {
            group to-CE1 {
               family inet {
                  unicast {
                     rib-group vpna-to-inet0;
                  }
               }
               peer-as 63001;
               neighbor 192.168.197.14;
            }
         }
      }
   }
}
```

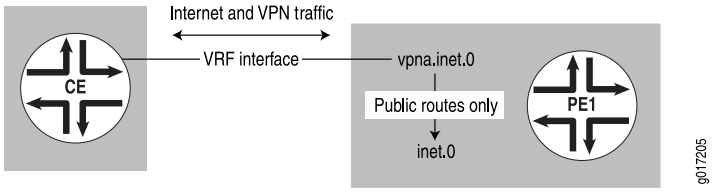
Routing VPN and Internet Traffic Through the Same Interface Bidirectionally (VPN Has Private Addresses)

The example in this section shows how to route VPN and Internet traffic through the same interface in both directions (from the CE router to the Internet and from the Internet to the CE router). The VPN in this example has private addresses. If you can configure EBGP on the CE router, you can configure a PE router using the configuration

outlined in “Routing VPN and Internet Traffic Through the Same Interface Bidirectionally (VPN Has Public Addresses)” on page 309, even if the VPN has private addresses.

In the example described in this section, the CE router uses separate communities to advertise its VPN routes and public routes. The PE router selectively imports only the public routes into the inet.0 routing table. This configuration ensures that return traffic from the Internet uses the same interface between the PE and CE routers as that used by VPN traffic going out to public Internet addresses (see Figure 39 on page 313).

Figure 39: VPN and Internet Traffic Routed Through the Same Interface



In this example, the CE router has one interface and a BGP session with the PE router, and it tags VPN routes and Internet routes with different communities. The PE router has one interface, selectively imports routes for the VPN’s public IP address pool into inet.0, and has a default route in the VRF routing table pointing to inet.0.

The following sections show how to route VPN and Internet traffic through the same interface bidirectionally (VPN has private addresses):

- Configuring Routing Options for Router PE1 on page 313
- Configuring a Routing Instance for Router PE1 on page 314
- Configuring Policy Options for Router PE1 on page 314
- Traffic Routed by the Same Interface Bidirectionally (VPN Has Private Addresses): Configuration Summarized by Router on page 315

Configuring Routing Options for Router PE1

On Router PE1, configure a routing table group to install VPN routes in the vpna.inet.0 and inet.0 routing tables:

```
[edit]
routing-options {
    rib-groups {
        vpna-to-inet0 {
            import-rib [ vpna.inet.0 inet.0 ];
        }
    }
}
```

Configuring a Routing Instance for Router PE1

On Router PE1, configure a routing instance. As part of the configuration for the routing instance, configure a static route that is installed in vpna.inet.0 and is pointed at inet.0 for resolution.

```
[edit]
routing-instances {
    vpna {
        instance-type vrf;
        interface t3-0/2/0.0;
        route-distinguisher 10.255.14.171:100;
        vrf-import vpna-import;
        vrf-export vpna-export;
        routing-options {
            static {
                route 0.0.0.0/0 next-table inet.0;
            }
        }
    }
}
```

At the [edit routing-instances vpna protocols bgp] hierarchy level, configure a policy (import-public-addr-to-inet0) to import public routes into inet.0 and a routing table group (vpna-to-inet0) to allow BGP to install routes into multiple routing tables (vpna.inet.0 and inet.0):

```
[edit routing-instances vpna]
protocols {
    bgp {
        group to-CE1 {
            import import-public-addr-to-inet0;
            family inet {
                unicast {
                    rib-group vpna-to-inet0;
                }
            }
            peer-as 63001;
            neighbor 192.168.197.14;
        }
    }
}
```

Configuring Policy Options for Router PE1

Configure the policy options for Router PE1 to accept all routes initially (term a) and then to install routes with a public-comm community into routing table inet.0 (term b):

```
[edit]
policy-options {
    policy-statement import-public-addr-to-inet0 {
        term a {
            from {
                protocol bgp;
```

```
            rib vpna.inet.0;
            community [ public-comm private-comm ];
          }
          then accept;
        }
        term b {
          from {
            protocol bgp;
            community public-comm;
          }
          to rib inet.0;
          then accept;
        }
        term c {
          then reject;
        }
      }
      community private-comm members target:1:333;
      community public-comm members target:1:111;
      community vpna-comm members target:63000:100;
    }
```

Traffic Routed by the Same Interface Bidirectionally (VPN Has Private Addresses): Configuration Summarized by Router

Router PE1

Routing Options

```
routing-options {
  rib-groups {
    vpna-to-inet0 {
      import-rib [ vpna.inet.0 inet.0 ];
    }
  }
}
```

Routing Instances

```
routing-instances {
  vpna {
    instance-type vrf;
    interface t3-0/2/0.0;
    route-distinguisher 10.255.14.171:100;
    vrf-import vpna-import;
    vrf-export vpna-export;
    routing-options {
      static {
        route 0.0.0.0/0 next-table inet.0;
      }
    }
  }
}
```

Routing Instances Protocols BGP

```
protocols {
  bgp {
    group to-CE1 {
      import import-public-addr-to-inet0;
```

```
            family inet {
               unicast {
                  rib-group vpna-to-inet0;
               }
            }
            peer-as 63001;
            neighbor 192.168.197.14;
         }
      }
   }
```

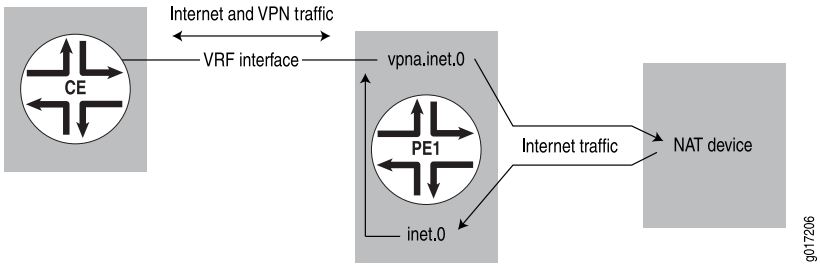
Policy Options

```
policy-options {
   policy-statement import-public-addr-to-inet0 {
      term a {
         from {
            protocol bgp;
            rib vpna.inet.0;
            community [ public-comm private-comm ];
         }
         then accept;
      }
      term b {
         from {
            protocol bgp;
            community public-comm;
         }
         to rib inet.0;
         then accept;
      }
      term c {
         then reject;
      }
   }
   community private-comm members target:1:333;
   community public-comm members target:1:111;
   community vpna-comm members target:63000:100;
}
```

Routing Internet Traffic Through a Separate NAT Device

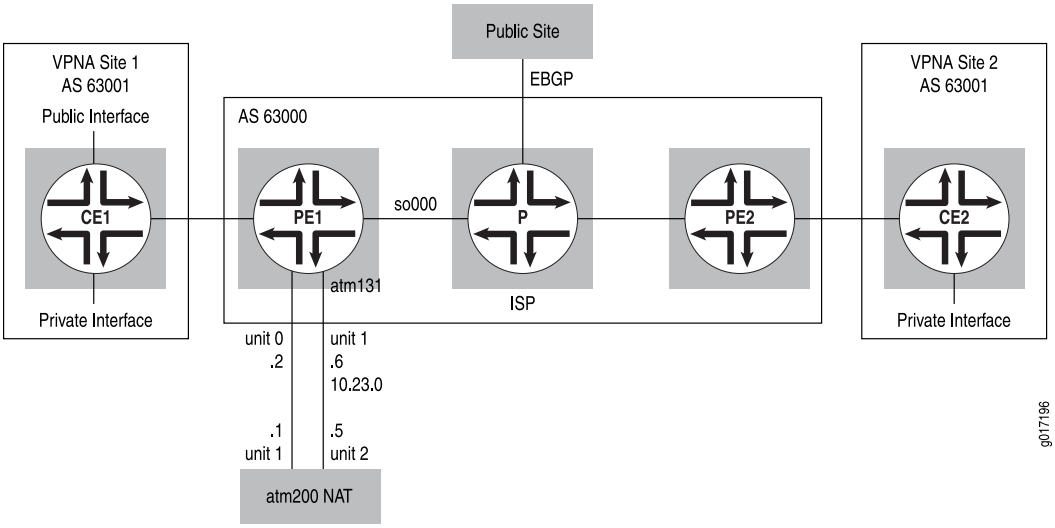
In this example, the CE router does not perform NAT. It sends both VPN and Internet traffic over the same interface to the PE router. The PE router is connected to a NAT device by means of two interfaces. One interface is configured in the PE router's VRF table and points to a VPN interface on the NAT device, which can route Internet traffic for the VPN. The other interface is in a default instance; for example, part of public routing table inet.0. There can be a single physical connection between the PE router and the NAT device and multiple logical connections—one for each VRF table and another interface—as part of the global routing table (see Figure 40 on page 317).

Figure 40: Internet Traffic Routed Through a Separate NAT Device



This example's topology expands upon that illustrated in Figure 36 on page 301. The CE router sends both VPN and Internet traffic to Router PE1. VPN traffic is routed based on the VPN routes received by Router PE1. Traffic for everything else is sent to the NAT device using Router PE1's private interface to the NAT device, which then translates the private addresses and sends the traffic back to Router PE1 using that router's public interface (see Figure 41 on page 317).

Figure 41: Internet Traffic Routed Through a NAT Example Topology



The following sections show how to route Internet traffic through a separate NAT device:

- Configuring Interfaces on Router PE1 on page 318
- Configuring Routing Options for Router PE1 on page 318
- Configuring Routing Protocols on Router PE1 on page 319
- Configuring a Routing Instance for Router PE1 on page 319
- Traffic Routed by Separate NAT Device: Configuration Summarized by Router on page 321

Configuring Interfaces on Router PE1

Configure an interface for VPN traffic to and from Router CE1, an interface for VPN traffic to and from the NAT device, and an interface for Internet traffic to and from the NAT device:

```
[edit]
interfaces {
    t3-0/2/0 {
        dce;
        encapsulation frame-relay;
        unit 0 {
            description "to CE1 VPN interface";
            dlci 10;
            family inet {
                address 192.168.197.13/30;
            }
        }
    }
    at-1/3/1 {
        atm-options {
            vpi 1 maximum-vcs 255;
        }
        unit 0 {
            description "to NAT VPN interface";
            vci 1.100;
            family inet {
                address 10.23.0.2/32 {
                    destination 10.23.0.1;
                }
            }
        }
        unit 1 {
            description "to NAT public interface";
            vci 1.101;
            family inet {
                address 10.23.0.6/32 {
                    destination 10.23.0.5;
                }
            }
        }
    }
}
```

Configuring Routing Options for Router PE1

Configure a static route on Router PE1 to direct Internet traffic to the CE router through the NAT device. Router PE1 distributes this route to the Internet.

```
[edit]
routing-options {
    static {
        route 10.12.1.0/24 next-hop 10.23.0.5;
    }
}
```

Configuring Routing Protocols on Router PE1

Configure MPLS, BGP, IS-IS, and LDP on Router PE1. For the MPLS configuration, include the NAT device's VPN interface in the VRF table. As part of the BGP configuration, include a policy to advertise the public IP address pool:

```
[edit]
protocols {
    mpls {
        interface t3-0/2/0.0;
        interface at-1/3/1.0;
    }
    bgp {
        group pe-pe {
            type internal;
            local-address 10.255.14.171;
            family inet {
                any;
            }
            family inet-vpn {
                any;
            }
            export [ fix-nh redist-static ];
            neighbor 10.255.14.177;
            neighbor 10.255.14.173;
        }
    }
    isis {
        level 1 disable;
        interface so-0/0/0.0;
        interface lo0.0;
    }
    ldp {
        interface so-0/0/0.0;
    }
}
```

Configuring a Routing Instance for Router PE1

Configure a routing instance on Router PE1. As part of the routing instance configuration, under routing-options, configure a static default route in vpna.inet.0 pointing to the NAT device's VPN interface (this directs all non-VPN traffic to the NAT device):

```
[edit]
routing-instances {
    vpna {
        instance-type vrf;
        interface t3-0/2/0.0;
        interface at-1/3/1.0;
        route-distinguisher 10.255.14.171:100;
        vrf-import vpna-import;
        vrf-export vpna-export;
        routing-options {
            static {
```

```
            route 0.0.0.0/0 next-hop 10.23.0.1;
          }
        }
        protocols {
          bgp {
            group to-CE1 {
              peer-as 63001;
              neighbor 192.168.197.14;
            }
          }
        }
      }
    }
    policy-options {
      policy-statement fix-nh {
        then {
          next-hop self;
        }
      }
      policy-statement redist-static {
        term a {
          from {
            protocol static;
            route-filter 10.12.1.0/24 exact;
          }
          then accept;
        }
        term b {
          from protocol bgp;
          then accept;
        }
        term c {
          then accept;
        }
      }
      policy-statement vpna-import {
        term a {
          from {
            protocol bgp;
            community vpna-comm;
          }
          then accept;
        }
        term b {
          then reject;
        }
      }
      policy-statement vpna-export {
        term a {
          from protocol bgp;
          then {
            community add vpna-comm;
            accept;
          }
        }
        term b {
```

```
        then reject;
      }
    }
    community vpna-comm members target:63000:100;
  }
```

Traffic Routed by Separate NAT Device: Configuration Summarized by Router

Router PE1

Interfaces

```
interfaces {
  t3-0/2/0 {
    dce;
    encapsulation frame-relay;
    unit 0 {
      description "to CE1 VPN interface";
      dlci 10;
      family inet {
        address 192.168.197.13/30;
      }
    }
  }
  at-1/3/1 {
    atm-options {
      vpi 1 maximum-vcs 255;
    }
    unit 0 {
      description "to NAT VPN interface";
      vci 1.100;
      family inet {
        address 10.23.0.2/32 {
          destination 10.23.0.1;
        }
      }
    }
    unit 1 {
      description "to NAT public interface";
      vci 1.101;
      family inet {
        address 10.23.0.6/32 {
          destination 10.23.0.5;
        }
      }
    }
  }
}
```

Routing Options

```
routing-options {
  static {
    route 10.12.1.0/24 next-hop 10.23.0.5;
  }
}
```

Routing Protocols

```
protocols {
    mpls {
        interface t3-0/2/0.0;
        interface at-1/3/1.0;
    }
    bgp {
        group pe-pe {
            type internal;
            local-address 10.255.14.171;
            family inet {
                any;
            }
            family inet-vpn {
                any;
            }
            export [ fix-nh redist-static ];
            neighbor 10.255.14.177;
            neighbor 10.255.14.173;
        }
    }
    isis {
        level 1 disable;
        interface so-0/0/0.0;
        interface lo0.0;
    }
    ldp {
        interface so-0/0/0.0;
    }
}
```

Routing Instance

```
routing-instances {
    vpna {
        instance-type vrf;
        interface t3-0/2/0.0;
        interface at-1/3/1.0;
        route-distinguisher 10.255.14.171:100;
        vrf-import vpna-import;
        vrf-export vpna-export;
        routing-options {
            static {
                route 0.0.0.0/0 next-hop 10.23.0.1;
            }
        }
        protocols {
            bgp {
                group to-CE1 {
                    peer-as 63001;
                    neighbor 192.168.197.14;
                }
            }
        }
    }
}
```

Policy Options

```
policy-options {
```

```
    policy-statement fix-nh {
        then {
            next-hop self;
        }
    }
    policy-statement redist-static {
        term a {
            from {
                protocol static;
                route-filter 10.12.1.0/24 exact;
            }
            then accept;
        }
        term b {
            from protocol bgp;
            then accept;
        }
        term c {
            then accept;
        }
    }
    policy-statement vpna-import {
        term a {
            from {
                protocol bgp;
                community vpna-comm;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    policy-statement vpna-export {
        term a {
            from protocol bgp;
            then {
                community add vpna-comm;
                accept;
            }
        }
        term b {
            then reject;
        }
    }
    community vpna-comm members target:63000:100;
}
```

Centralized Internet Access

This section describes several ways to configure a CE router to act as a central site for Internet access. Internet traffic from other sites (CE routers) is routed to the hub CE router (which also performs NAT) using that router's VPN interface. The hub CE router then forwards the traffic to a PE router connected to the Internet through

another interface identified in the inet.0 table. The hub CE router can advertise a default route to the spoke CE routers. The disadvantage of this type of configuration is that all traffic has to go through the central CE router before going to the Internet, causing network delays if this router receives too much traffic. However, in a corporate network, traffic might have to be routed to a central site because most corporate networks separate the VPN from the Internet by means of a single firewall.

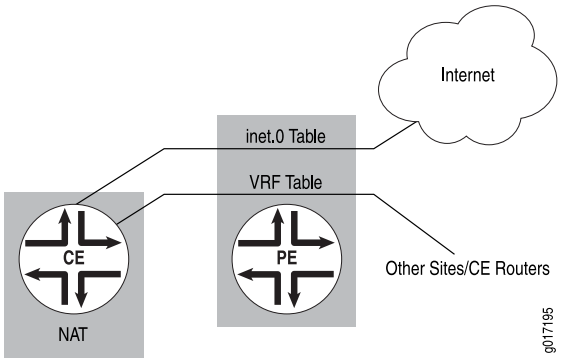
This section includes the following examples:

- Routing Internet Traffic Through a Hub CE Router on page 324
- Routing Internet Traffic Through Multiple CE Routers on page 328

Routing Internet Traffic Through a Hub CE Router

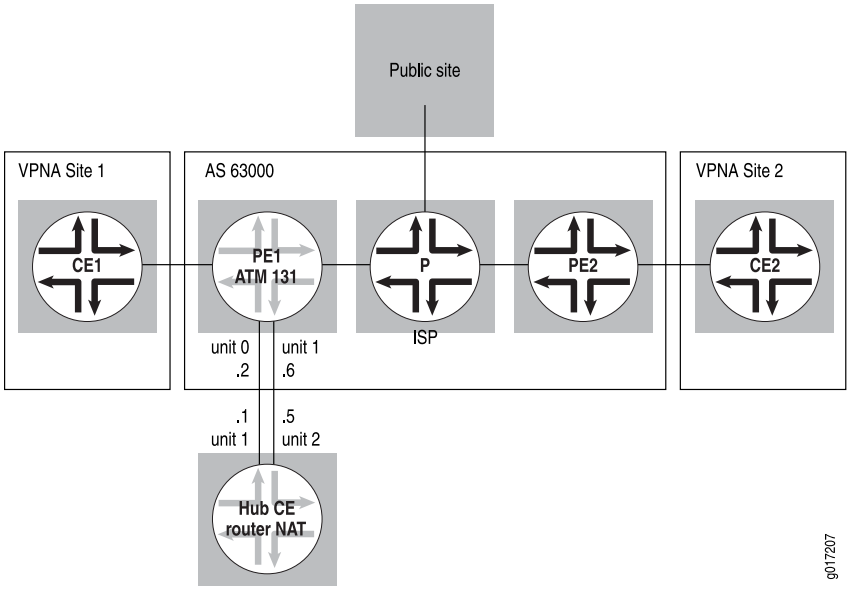
In this example, Internet traffic is routed through a hub CE router. The hub CE router has two interfaces to the hub PE router: a VPN interface and a public interface. It performs NAT on traffic forwarded from the hub PE router through the VPN interface and forwards that traffic from its public interface back to the hub PE router. The hub PE router has a static default route in its VRF table pointing to the hub CE router's VPN interface. It announces this default route to the rest of the VPN, attracting all non-VPN traffic to the hub CE route. The hub PE router also installs and distributes the VPN's public IP address space (see Figure 42 on page 324).

Figure 42: Internet Access Through a Hub CE Router Performing NAT



The configuration for this example is almost identical to that described in "Routing Internet Traffic Through a Separate NAT Device" on page 316. The difference is that Router PE1 is configured to announce a static default route to the other CE routers (see Figure 43 on page 325).

Figure 43: Internet Access Provided Through a Hub CE Router



The following sections show how to configure centralized Internet access by routing Internet traffic through a hub CE router:

- Configuring a Routing Instance on Router PE1 on page 325
- Configuring Policy Options on Router PE1 on page 326
- Internet Traffic Routed by a Hub CE Router: Configuration Summarized by Router on page 327

Configuring a Routing Instance on Router PE1

Configure a routing instance for Router PE1. As part of this configuration, under routing-options, configure a default static route (route 0.0.0.0/0) to be installed in vpna.inet.0, and point the route to the hub CE router's VPN interface (10.23.0.1). Also, configure BGP under the routing instance to export the default route to the local CE router:

```
[edit]
routing-instances {
   vpna {
      instance-type vrf;
      interface t3-0/2/0.0;
      interface at-1/3/1.0;
      route-distinguisher 10.255.14.171:100;
      vrf-import vpna-import;
      vrf-export vpna-export;
      routing-options {
         static {
            route 0.0.0.0/0 next-hop 10.23.0.1;
         }
      }
      protocols {
```

```
        bgp {
            group to-CE1 {
                export export-default;
                peer-as 63001;
                neighbor 192.168.197.14;
            }
        }
    }
}
```

Configuring Policy Options on Router PE1

Configure policy options on Router PE1. As part of this configuration, Router PE1 should export the static default route to all the remote PE routers in vpna (configured in the policy-statement vpna-export statement under term b):

```
[edit]
policy-options {
    policy-statement vpna-export {
        term a {
            from protocol bgp;
            then {
                community add vpna-comm;
                accept;
            }
        }
        term b {
            from {
                protocol static;
                route-filter 0.0.0.0/0 exact;
            }
            then {
                community add vpna-comm;
                accept;
            }
        }
        term c {
            then reject;
        }
    }
    policy-statement export-default {
        term a {
            from {
                protocol static;
                route-filter 0.0.0.0/0 exact;
            }
            then accept;
        }
        term b {
            from protocol bgp;
            then accept;
        }
        term c {
            then reject;
```

```
        }
    }
}
```

Internet Traffic Routed by a Hub CE Router: Configuration Summarized by Router

Router PE1

The configuration for Router PE1 is almost identical to that for the example in "Routing Internet Traffic Through a Separate NAT Device" on page 316. The difference is that Router PE1 is configured to announce a static default route to the other CE routers.

Routing Instance

```
routing-instances {
    vpna {
        instance-type vrf;
        interface t3-0/2/0.0;
        interface at-1/3/1.0;
        route-distinguisher 10.255.14.171:100;
        vrf-import vpna-import;
        vrf-export vpna-export;
        routing-options {
            static {
                route 0.0.0.0/0 next-hop 10.23.0.1;
            }
        }
        protocols {
            bgp {
                group to-CE1 {
                    export export-default;
                    peer-as 63001;
                    neighbor 192.168.197.14;
                }
            }
        }
    }
}
```

Policy Options

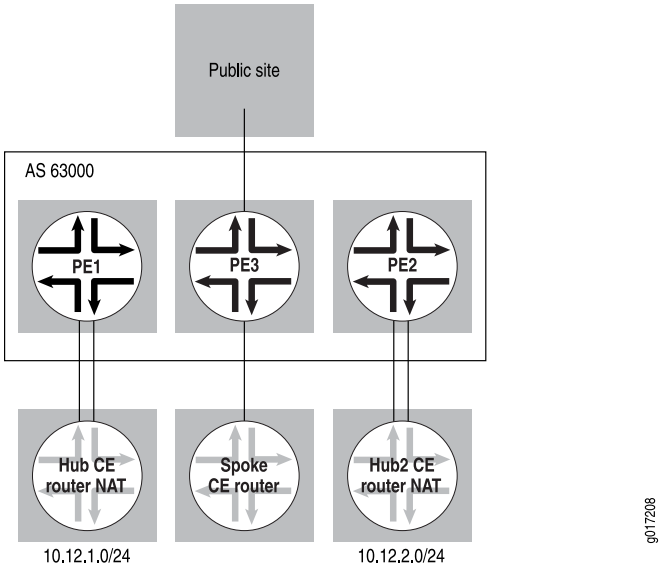
```
policy-options {
    policy-statement vpna-export {
        term a {
            from protocol bgp;
            then {
                community add vpna-comm;
                accept;
            }
        }
        term b {
            from {
                protocol static;
                route-filter 0.0.0.0/0 exact;
            }
            then {
                community add vpna-comm;
```

```
                accept;
            }
        }
        term c {
            then reject;
        }
    }
    policy-statement export-default {
        term a {
            from {
                protocol static;
                route-filter 0.0.0.0/0 exact;
            }
            then accept;
        }
        term b {
            from protocol bgp;
            then accept;
        }
        term c {
            then reject;
        }
    }
}
```

Routing Internet Traffic Through Multiple CE Routers

The example in this section is an extension of that described in "Routing Internet Traffic Through a Hub CE Router" on page 324. This example provides different exit points for different sites by means of multiple hub CE routers that perform similar functions. Each hub CE router tags the default route with a different route target and allows the spoke CE routers to select the hub site that should be used for Internet access (see Figure 44 on page 329).

Figure 44: Two Hub CE Routers Handling Internet Traffic and NAT



This example uses two hub CE routers that handle NAT and Internet traffic:

- Hub1 CE router tags 0/0 with community public-comm1 (target: 1:111)
- Hub2 CE router tags 0/0 with community public-comm2 (target: 1:112)

The spoke CE router in this example is configured to have a bias toward Hub2 for Internet access.

The following sections describe how configure two hub CE routers to handle internet traffic and NAT:

- Configuring a Routing Instance on Router PE1 on page 329
- Configuring Policy Options on Router PE1 on page 330
- Configuring a Routing Instance on Router PE3 on page 331
- Configuring Policy Options on Router PE3 on page 331
- Routing Internet Traffic Through Multiple CE Routers: Configuration Summarized by Router on page 332

Configuring a Routing Instance on Router PE1

Configure a routing instance on Router PE1:

```
[edit]
routing-instances {
   vpna {
      instance-type vrf;
      interface t3-0/2/0.0;
      interface at-1/3/1.0;
      route-distinguisher 10.255.14.171:100;
      vrf-import vpna-import;
```

```
        vrf-export vpna-export;
        routing-options {
            static {
                route 0.0.0.0/0 next-hop 10.23.0.1;
            }
        }
        protocols {
            bgp {
                group to-CE1 {
                    export export-default;
                    peer-as 63001;
                    neighbor 192.168.197.14;
                }
            }
        }
    }
}
```

Configuring Policy Options on Router PE1

The policy options for Router PE1 are the same as in "Routing Internet Traffic Through a Hub CE Router" on page 324, but the configuration in this example includes an additional community, public-comm1, in the export statement:

```
[edit]
policy-options {
    policy-statement vpna-import {
        term a {
            from {
                protocol bgp;
                community vpna-comm;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    policy-statement vpna-export {
        term a {
            from {
                protocol static;
                route-filter 0.0.0.0/0 exact;
            }
            then {
                community add public-comm1;
                community add vpna-comm;
                accept;
            }
        }
        term b {
            from protocol bgp;
            then {
                community add vpna-comm;
                accept;
```

```
            }
        }
        term c {
            then reject;
        }
    }
    community public-comm1 members target:1:111;
    community public-comm2 members target:1:112;
    community vpna-comm members target:63000:100;
}
```

The configuration of Router PE2 is identical to that of Router PE1 except that Router PE2 exports the default route through community public-comm2.

Configuring a Routing Instance on Router PE3

Configure routing instance vpna on Router PE3:

```
[edit]
routing-instances {
    vpna {
        instance-type vrf;
        interface t1-0/2/0.0;
        route-distinguisher 10.255.14.173:100;
        vrf-import vpna-import;
        vrf-export vpna-export;
        protocols {
            rip {
                group to-vpn12 {
                    export export-CE;
                    neighbor t1-0/2/0.0;
                }
            }
        }
    }
}
```

Configuring Policy Options on Router PE3

Configure the vrf-import policy for Router PE3 to select the Internet exit point based on the additional communities specified in "Configuring Policy Options on Router PE1" on page 330:

```
[edit]
policy-options {
    policy-statement vpna-export {
        term a {
            from protocol rip;
            then {
                community add vpna-comm;
                accept;
            }
        }
        term b {
            then reject;
```

```
        }
    }
    policy-statement vpna-import {
        term a {
            from {
                protocol bgp;
                community public-comm1;
                route-filter 0.0.0.0/0 exact;
            }
            then reject;
        }
        term b {
            from {
                protocol bgp;
                community vpna-comm;
            }
            then accept;
        }
        term c {
            then reject;
        }
    }
    policy-statement export-CE {
        from protocol bgp;
        then accept;
    }
    community vpna-comm members target:69:100;
    community public-comm1 members target:1:111;
    community public-comm2 members target:1:112;
}
```

Routing Internet Traffic Through Multiple CE Routers: Configuration Summarized by Router

Router PE1

This configuration is an extension of the example in "Routing Internet Traffic Through a Hub CE Router" on page 324. It provides different exit points for various sites by using multiple hub CE routers that perform similar functions.

Routing Instances

```
routing-instances {
    vpna {
        instance-type vrf;
        interface t3-0/2/0.0;
        interface at-1/3/1.0;
        route-distinguisher 10.255.14.171:100;
        vrf-import vpna-import;
        vrf-export vpna-export;
        routing-options {
            static {
                route 0.0.0.0/0 next-hop 10.23.0.1;
            }
        }
        protocols {
            bgp {
```

```
            group to-CE1 {
                export export-default;
                peer-as 63001;
                neighbor 192.168.197.14;
            }
        }
    }
}
```

Policy Options

```
policy-options {
    policy-statement vpna-import {
        term a {
            from {
                protocol bgp;
                community vpna-comm;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    policy-statement vpna-export {
        term a {
            from {
                protocol static;
                route-filter 0.0.0.0/0 exact;
            }
            then {
                community add public-comm1;
                community add vpna-comm;
                accept;
            }
        }
        term b {
            from protocol bgp;
            then {
                community add vpna-comm;
                accept;
            }
        }
        term c {
            then reject;
        }
    }
    community public-comm1 members target:1:111;
    community public-comm2 members target:1:112;
    community vpna-comm members target:63000:100;
}
```

Router PE2

The configuration of Router PE2 is identical to that of Router PE1, except that Router PE2 exports the default route through community public-comm2 (see "Policy Options" on page 333).

Router PE3

Routing Instances

```
routing-instances {
   vpna {
      instance-type vrf;
      interface t1-0/2/0.0;
      route-distinguisher 10.255.14.173:100;
      vrf-import vpna-import;
      vrf-export vpna-export;
      protocols {
         rip {
            group to-vpn12 {
               export export-CE;
               neighbor t1-0/2/0.0;
            }
         }
      }
   }
}
```

Policy Options

```
policy-options {
   policy-statement vpna-export {
      term a {
         from protocol rip;
         then {
            community add vpna-comm;
            accept;
         }
      }
      term b {
         then reject;
      }
   }
   policy-statement vpna-import {
      term a {
         from {
            protocol bgp;
            community public-comm1;
            route-filter 0.0.0.0/0 exact;
         }
         then reject;
      }
      term b {
         from {
            protocol bgp;
            community vpna-comm;
         }
         then accept;
      }
```

```
        term c {
            then reject;
        }
    }
    policy-statement export-CE {
        from protocol bgp;
        then accept;
    }
    community vpna-comm members target:69:100;
    community public-comm1 members target:1:111;
    community public-comm2 members target:1:112;
}
```

Chapter 14

Summary of Layer 3 VPN Configuration Statements

The following section explains the major routing-instances configuration statements that apply specifically to Layer 3 virtual private networks (VPNs).

classifiers

Syntax

```
classifiers {
   exp (classifier-name | default);
}
```

Hierarchy Level [edit class-of-service routing-instances *routing-instance-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description For routing instances with VRF table labels enabled, apply a custom MPLS EXP classifier to the routing instance. You can apply the default MPLS EXP classifier or one that is previously defined.

Default If you do not include this statement, the default MPLS EXP classifier is applied to the routing instance.

Options *classifier-name*—Name of the behavior aggregate MPLS EXP classifier.

Usage Guidelines See "Applying MPLS EXP Classifiers to Routing Instances" on page 167 and the *JUNOS Network Interfaces Configuration Guide*.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

domain-id

Syntax domain-id *domain-id*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols (ospf | ospf3)],
[edit routing-instances *routing-instance-name* protocols (ospf | ospf3)]

Release Information Statement introduced before JUNOS Release 7.4.

Description Specify a domain ID for a route. The domain ID identifies the OSPFv2 domain from which the route originated.

Default If the router ID is not configured in the routing instance, the router ID is derived from an interface address belonging to the routing instance.

Options *domain-id*—IP address.

Usage Guidelines See "Configuring an OSPF Domain ID" on page 152.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

domain-vpn-tag

Syntax domain-vpn-tag *number*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols (ospf | ospf3)],
[edit routing-instances *routing-instance-name* protocols (ospf | ospf3)]

Release Information Statement introduced before JUNOS Release 7.4.

Description Set a virtual private network (VPN) tag for OSPFv2 external routes generated by the provider edge (PE) router.

Options *number*—VPN tag.

Usage Guidelines See "Configuring an OSPF Domain ID" on page 152.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

dynamic-tunnels

Syntax

```
dynamic-tunnels tunnel-name {
    destination-networks prefix;
    source-address address;
    tunnel-type gre;
}
```

Hierarchy Level [edit logical-systems *logical-system-name* routing-options],
[edit routing-options]

Release Information Statement introduced before JUNOS Release 7.4.

Description Enable dynamic tunnel creation.

Options

destination-networks *prefix*—Specifies the IP version 4 (IPv4) prefix range for the destination network by including the destination-networks statement. Only tunnels within the specified IPv4 prefix range are allowed to be initiated.

source-address *address*—Specifies the source address for the generic routing encapsulation (GRE) tunnels. The source address specifies the address used as the source for the local tunnel endpoint. This could be any local address on the router (typically the router ID or the loopback address).

tunnel-name—Specifies the name of the dynamic tunnel.

tunnel-type gre—Specifies that a GRE tunnel is to be dynamically created.

Usage Guidelines See "Configuring GRE Tunnels Dynamically" on page 174 and the *JUNOS Routing Protocols Configuration Guide*.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

independent-domain

Syntax independent-domain;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options autonomous-system <loops *number*>],
[edit logical-systems *logical-system-name* routing-options autonomous-system <loops *number*>],
[edit routing-instances *routing-instance-name* routing-options autonomous-system <loops *number*>],
[edit routing-options autonomous-system <loops *number*>]

Release Information Statement introduced before JUNOS Release 7.4.

Description Improve the transparency of Layer 3 VPN services for customer networks by preventing the internal BGP (IBGP) routes that originate within an autonomous system (AS) in the customer network from being sent to a service provider's AS. Similarly, IBGP routes that originate within an AS in the service provider's network are prevented from being sent to a customer AS.

Usage Guidelines See "Configuring Layer 3 VPNs to Carry IBGP Traffic" on page 161 and the *JUNOS Routing Protocols Configuration Guide*.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

inet6-vpn

Syntax

```
inet6-vpn (any | multicast | unicast) {
    aggregate-label;
    prefix-limit maximum;
    rib-group rib-group-name;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols bgp family],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* family],
[edit protocols bgp family],
[edit protocols bgp group *group-name* family]

Release Information

Statement introduced before JUNOS Release 7.4.

Description

Enable IP version 6 (IPv6) on the provider edge (PE) router for the Layer 3 VPN.

Options

any—Configure the family type to be both multicast and unicast.

multicast—Configure the family type to be multicast. This means that the BGP peers are being used only to carry the unicast routes that are being used by multicast for resolving the multicast routes.

prefix-limit *maximum*—Maximum prefix limit.
Range: 1 through 4,294,967,295
Default: 1

rib-group *rib-group-name*—The name of the routing table group.

unicast—Configure the family type to be unicast. This means that the BGP peers only carry the unicast routes that are being used for unicast forwarding purposes.

Usage Guidelines

See "Configuring IPv6 Between the PE and CE Routers" on page 157 and the *JUNOS Routing Protocols Configuration Guide*.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

maximum-paths

Syntax maximum-paths *path-limit* <log-interval *interval* | log-only | threshold *percentage*>;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options],
[edit logical-systems *logical-system-name* routing-options],
[edit routing-instances *routing-instance-name* routing-options],
[edit routing-options]

Release Information Statement introduced in JUNOS Release 8.0.

Description Specify a maximum limit on the number of paths that can be installed into the routing tables. Using a path limit, you can curtail the number of paths received from a CE router in a VPN. Path limits apply only to dynamic routing protocols and are not applicable to static or interface routes.

Options *path-limit*—Specify the maximum number of paths.
Range: 1 through 4,294,967,295 paths

log-interval—Minimum interval between log messages.
Range: 5 through 86,400 seconds

log-only—Generate warning messages only. No limit is placed on the number of paths stored in the routing tables.

threshold—Percentage of the path limit at which to begin sending warning log messages.

Usage Guidelines See "Limiting the Paths and Prefixes Accepted from a CE Router" on page 156.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

maximum-prefixes

Syntax maximum-prefixes *prefix-limit* <log-interval *interval* | log-only | threshold *percentage*>;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options],
[edit logical-systems *logical-system-name* routing-options],
[edit routing-instances *routing-instance-name* routing-options],
[edit routing-options]

Release Information Statement introduced in JUNOS Release 8.0.

Description Specify a maximum limit on the number of prefixes that can be installed into the routing tables. Using a prefix limit, you can curtail the number of prefixes received from a CE router in a VPN. Prefix limits apply only to dynamic routing protocols and are not applicable to static or interface routes.

Options *prefix-limit*—Specify the maximum number of prefixes.
Range: 1 through 4,294,967,295 prefixes

log-interval—Minimum interval between log messages.
Range: 5 through 86,400 seconds

log-only—Generate warning messages only. No limit is placed on the number of prefixes stored in the routing tables.

threshold—Percentage of the prefix limit at which to begin sending warning log messages.

Usage Guidelines See "Limiting the Paths and Prefixes Accepted from a CE Router" on page 156.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

metric

Syntax metric *number*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols ospf area *area-id* sham-link-remote],
[edit routing-instances *routing-instance-name* protocols ospf area *area-id* sham-link-remote]

Release Information Statement introduced before JUNOS Release 7.4.

Description Specify the cost of using the Open Shortest Path First (OSPF) sham link.

Default *number*—1

Options *number*—1 through 65,535

Usage Guidelines See "Configuring OSPF Sham Links" on page 150.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

multihop

Syntax multihop *ttl-value*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure an external BGP (EBGP) multihop session between the PE and customer edge (CE) routers of a Layer 3 VPN. This allows you to have one or more routers between the PE and CE routers.

Options *ttl-value*—Specify the time-to-live (TTL) value for the multihop session to prevent routing loops.

Usage Guidelines See "Configuring EBGP or IBGP Multihop Between PE and CE Routers" on page 160.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

multipath

Syntax

```
multipath {
    vpn-unequal-cost equal-external-internal;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options rib *routing-table-name*],
[edit routing-instances *routing-instance-name* routing-options],
[edit routing-instances *routing-instance-name* routing-options rib *routing-table-name*]

Release Information

Statement introduced before JUNOS Release 7.4. The equal-external-internal option was added for JUNOS Release 8.4.

Description

Enable protocol-independent load balancing for Layer 3 VPNs. This allows the forwarding next hops for both the active route and alternative paths to be used for load balancing.

The options are explained separately.

Usage Guidelines

See "Protocol-Independent Load Balancing for Layer 3 VPNs" on page 181.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

routing-instances

Syntax

```
routing-instances routing-instance-name {
    classifiers {
        exp (classifier-name | default);
    }
}
```

Hierarchy Level [edit class-of-service]

Release Information Statement introduced before JUNOS Release 7.4.

Description For routing instances with the vrf-table-label statement configured, apply a custom MPLS EXP classifier to the routing instance. You can apply the default MPLS EXP classifier or one that is previously defined.

Options *routing-instance-name*—Name of the routing instance.

The other statements are explained separately.

Usage Guidelines See "Applying MPLS EXP Classifiers to Routing Instances" on page 167 and the *JUNOS Network Interfaces Configuration Guide*.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

sham-link

Syntax

```
sham-link {
    local address;
}
```

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols ospf],
[edit routing-instances *routing-instance-name* protocols ospf]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure a sham link for the Layer 3 VPN routing instance.

Options local *address*—The address for the local endpoint of the sham link.

Usage Guidelines See "Configuring OSPF Sham Links" on page 150 and the *JUNOS Routing Protocols Configuration Guide*.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

sham-link-remote

Syntax sham-link-remote *address* <metric *number*>;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols ospf area *area-id*],
[edit routing-instances *routing-instance-name* protocols ospf area *area-id*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure the address for the remote end point of the sham link.

Options *address*—Address for the remote end point of the sham link.

The metric statement is explained separately.

Usage Guidelines See "Configuring OSPF Sham Links for Layer 3 VPNs" on page 149 and the *JUNOS Routing Protocols Configuration Guide*.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

vpn-group-address

Syntax vpn-group-address *address*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim],
[edit routing-instances *routing-instance-name* protocols pim]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure the group address for the Layer 3 VPN in the service provider's network.

Options *address*—Address for the Layer 3 VPN in the service provider's network.

Usage Guidelines See "Configuring Multicast over Layer 3 VPNs" on page 170 and the *JUNOS Multicast Protocols Configuration Guide*.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

vpn-unequal-cost

Syntax
```
vpn-unequal-cost {
    equal-external-internal;
}
```

Hierarchy Level
```
[edit logical-systems logical-system-name routing-instances routing-instance-name
    routing-options multipath],
[edit logical-systems logical-system-name routing-instances routing-instance-name
    routing-options rib routing-table-name multipath],
[edit routing-instances routing-instance-name routing-options multipath]
[edit routing-instances routing-instance-name routing-options rib routing-table-name
    multipath]
```

Release Information Statement introduced before JUNOS Release 7.4. The equal-external-internal option was added for JUNOS Release 8.4.

Description Apply protocol-independent load balancing to VPN routes that are equal until their interior gateway protocol (IGP) metrics with regard to route selection. If you do not configure the vpn-unequal-cost statement, protocol-independent load balancing is applied to VPN routes that are equal until their router identifiers with regard to route selection.

Options equal-external-internal—Specifies that both external and internal BGP paths can be selected for multipath.

Usage Guidelines See "Configuring Load Balancing for Layer 3 VPNs" on page 181.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

vrf-table-label

Syntax
```
vrf-table-label;
```

Hierarchy Level
```
[edit logical-systems logical-system-name routing-instances routing-instance-name],
[edit routing-instances routing-instance-name]
```

Release Information Statement introduced before JUNOS Release 7.4.

Description Map the inner label of a packet to a specific VPN routing and forwarding (VRF) table. This allows the examination of the encapsulated IP header.

Usage Guidelines See "Filtering Traffic Based on the IP Header" on page 162 and "Configuring EXP-Based Traffic Classification for VPLS" on page 404.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Part 4
Multicast VPNs

Chapter 15

Multicast VPNs Overview

This chapter provides an overview of BGP MPLS multicast virtual private networks (VPNs), also known as next-generation multicast VPNs. This chapter discusses the following topics:

BGP MPLS Multicast VPN Overview

BGP MPLS multicast VPNs employ the intra-autonomous system (AS) next-generation (NGEN) BGP control plane and Protocol Independent Multicast (PIM) sparse mode as the data plane.

The main characteristics of 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 provider edge (PE) router-to-PE router control plane for multicast VPN.
- They eliminate the requirement for the virtual router (VR) model (as specified in Internet draft draft-rosen-vpn-mcast, *Multicast in MPLS/BGP VPNs*) for multicast VPNs and the RFC 2547 model for unicast VPNs.
- They rely on RFC 2547-based unicast with extensions for intra-AS and inter-AS communication.

A multicast VPN is defined by two sets of sites, a sender site set and a receiver site set. These sites have the following properties:

- Hosts within the sender site set can originate multicast traffic for receivers in the receiver site set.
- Receivers outside the receiver site set should not be able to receive this traffic.
- Hosts within the receiver site set can receive multicast traffic originated by any host in the sender site set.
- Hosts within the receiver site set should not be able to receive multicast traffic originated by any host that is not in the sender site set.

A site can be in both the sender site set and the receiver site set, so hosts within such a site can both originate and receive multicast traffic. For example, the sender sites set could be the same as the receiver site set, in which case all sites could both originate and receive multicast traffic from one another.

Sites within a given multicast VPN might be within the same organization or in different organizations, which means that a multicast VPN can be either an intranet or an extranet. A given site can be in more than one multicast VPN, so multicast VPNs may overlap. Not all sites of a given multicast VPN have to be connected to the same service provider, meaning that a multicast VPN can span multiple service providers.

Another way to look at a multicast VPN is to say that a multicast VPN is defined by a set of administrative policies. These policies determine both the sender site set and the receiver site set. These policies are established by multicast VPN customers, but implemented by multicast VPN service providers using the existing BGP and MPLS VPN infrastructure.

Multicast VPN Terminology

The following terminology describes aspects of multicast VPNs:

- Inclusive tree—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 VPN is an aggregate inclusive tree. An inclusive tree contains as its members all the PE routers that attach to the receiver sites of any of the multicast VPNs using the tree.
- Selective tree—A single multicast distribution tree in the backbone that carries traffic belonging only to a specified set of one or more multicast groups, from one or more multicast VPNs. An aggregate selective tree carries traffic for multicast groups that belong to different multicast VPNs. By default, traffic from most multicast groups could be carried by an inclusive tree, whereas traffic from high-bandwidth groups should be carried by a selective tree.

Multicast VPN Standards

Multicast VPNs are defined in the following IETF Internet drafts:

- Internet draft draft-ietf-l3vpn-2547bis-mcast-bgp-03.txt, *BGP Encodings for Multicast in MPLS/BGP IP VPNs*
- Internet draft draft-ietf-l3vpn-2547bis-mcast-02.txt, *Multicast in MPLS/BGP IP VPNs*

Chapter 16

Multicast VPNs Configuration

This chapter describes how to configure multicast virtual private networks (VPNs).

You configure multicast VPNs at a number of different hierarchy levels within the JUNOS software. However, a majority of multicast VPN statements are configured within a routing instance as follows:

```
description text;
instance-type vrf;
interface interface-name;
route-distinguisher (as-number:number | ip-address:number);
vrf-export [policy-names];
vrf-import [policy-names];
vrf-target (community | export community-name | import community-name);
protocols {
    mvpn {
        receiver-site;
        sender-site;
        route-target {
            export-target {
                target target-community;
                unicast;
            }
            import-target {
                target {
                    target-value;
                    receiver target-value;
                    sender target-value;
                }
                unicast {
                    receiver;
                    sender;
                }
            }
        }
    }
}
provider-tunnel {
    pim-asm group-address address;
    rsvp-te {
        label-switched-path-template (default-template | lsp-template-name);
        static-lsp lsp-name;
    }
    selective {
```

```
        group address {
            source source-address {
                rsvp-te {
                    label-switched-path-template (default-template | lsp-template-name);
                    static-lsp lsp-name;
                }
                threshold-rate number;
            }
        }
        tunnel-limit number;
    }
}
```

You can include these statements at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*]

For more information on how to configure multicast VPNs, see the *JUNOS Multicast Protocols Configuration Guide* and the *JUNOS Feature Guide*. The *JUNOS Feature Guide* an example of how to configure multicast VPNs.

This chapter describes how to configure multicast VPNs, discussing the following topics:

Configuring the Multicast VPN Routing Instance

To configure multicast VPNs, include the mvpn statement:

```
mvpn {
    receiver-site;
    route-target {
        export-target {
            target target-community;
            unicast;
        }
        import-target {
            target {
                target-value;
                receiver target-value;
                sender target-value;
            }
            unicast {
                receiver;
                sender;
```

```
        }
      }
    }
    sender-site;
    traceoptions {
      file filename <files number> <size size> <world-readable | no-world-readable>;
      flag flag <flag-modifier> <disable>;
    }
  }
```

You can include this statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols]

By default a multicast VPN routing instance is associated with both the multicast sender and the receiver sites. If you configure the receiver-site option, the routing instance is associated with only multicast receiver sites. Configuring the sender-site option associates the routing instance with only multicast sender sites.

NOTE: When you configure the routing instance for the multicast VPN, you must configure MPLS LSPs (either RSVP-signaled or LDP-signaled) between the PE routers of the routing instance to ensure VPN unicast connectivity. P2MP LSPs are used for multicast data forwarding only.

Configuring a Route Target for the Multicast VPN Routing Instance

By default, the VPN routing and forwarding (VRF) import and export route targets (configured either using VRF import and export policies or using the vrf-target statement) are used for importing and exporting routes with the multicast VPN network layer reachability information (NLRI).

You can use the export-target and import-target statements to override the default VRF import and export route targets. Export and import targets can also be specified specifically for sender sites or receiver sites, or can be borrowed from a configured unicast route target. Note that a sender site export route target is always advertised when security association routes are exported.

NOTE: When you configure a multicast VPN routing instance, you should not configure a target value for a multicast VPN specific route target that is identical to a target value for a unicast route target configured in another routing instance.

Specifying route targets in the multicast VPN NLRI for sender and receiver sites is useful when there is a mix of sender only, receiver only, and sender and receiver sites. A sender site route target is used for exporting automatic discovery routes by a sender site and for importing automatic discovery routes by a receiver site. A receiver site route target is used for exporting routes by a receiver site and importing

routes by a sender site. A sender and receiver site exports and imports routes with both route targets.

A provider edge (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 a PE router is configured to be in both sender sites and receiver sites, these 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 set and the receiver site set.

To configure a route target for the multicast VPN routing instance, include the route-target statement:

```
route-target {
    export-target {
        target target-community;
        unicast;
    }
    import-target {
        target {
            target-value;
            receiver target-value;
            sender target-value;
        }
        unicast {
            receiver;
            sender;
    }
```

You can include this statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols mvpn]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvpn]

The following sections describes how to configure the export target and the import target for a multicast VPN:

- Configuring the Export Target for the Multicast VPN on page 357
- Configuring the Import Target for the Multicast VPN on page 357

Configuring the Export Target for the Multicast VPN

To configure an export target, include the export-target statement:

```
export-target {
    target target-community;
    unicast;
}
```

You can include this statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols mvpn route-target]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvpn route target]

Configure the target option to specify the export target community. Configure the unicast option to use the same target community that has been specified for unicast.

Configuring the Import Target for the Multicast VPN

To configure an import target, include the import-target statement:

```
import-target {
    target target-value {
        receiver target-value;
        sender target-value;
    }
    unicast {
        receiver;
        sender;
    }
}
```

You can include this statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols mvpn route-target]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvpn route-target]

The following sections describe how to configure the import target and unicast parameters:

- Configuring the Import Target Receiver and Sender on page 357
- Configuring the Import Target Unicast Parameters on page 358

Configuring the Import Target Receiver and Sender

To configure the import target community, include the target statement:

```
target target-value {
    receiver target-value;
    sender target-value;
}
```

You can include this statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols mvpn route-target import-target]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvpn route-target import-target]

You can specify the target community used when importing receiver site sets by including the receiver option. You can specify the target community used when importing sender site sets by including the sender option.

Configuring the Import Target Unicast Parameters

To configure a unicast target community as the import target, include the unicast statement:

```
unicast {
    receiver;
    sender;
}
```

You can include this statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols mvpn route-target import-target]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvpn route-target import-target]

You can specify the unicast target community used when importing receiver site routes by configuring the receiver option. You can specify the unicast target community used when importing sender site routes by configuring the sender option.

Configuring NLRI Parameters for Multicast VPN

To configure IPv4 multicast VPN NLRI parameters, include the inet-mvpn statement:

```
inet-mvpn;
```

To configure IPv6 multicast VPN NLRI parameters, include the inet6-mvpn statement:

```
inet6-mvpn;
```

You can include these statements at the following hierarchy levels:

- [edit protocols bgp family]
- [edit logical-systems *logical-system-name* protocols bgp family]

Configuring PIM Provider Tunnels for Multicast VPNs

To configure a Protocol Independent Multicast (PIM) sparse mode provider tunnel for a multicast VPN, include the pim-asm statement:

```
pim-asm {
    group-address address;
}
```

You can include this statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* provider-tunnel]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel]

To complete the PIM sparse mode provider tunnel configuration, you also need to specify the group address using the group-address option. The source address for a PIM sparse mode provider tunnel is set to the loopback address of the loopback interface in the inet.0 routing table.

Configuring Point-to-Multipoint LSPs for Multicast VPNs

The JUNOS software supports point-to-multipoint label-switched paths (LSPs) for multicast VPNs. Point-to-multipoint LSPs for multicast VPNs are supported for intra-autonomous system (AS) environments (within an AS), but are not supported for inter-AS environments (between ASs). A point-to-multipoint LSP is an RSVP-signaled LSP with a single source and multiple destinations. For more information about point-to-multipoint LSPs, see the *JUNOS MPLS Applications Configuration Guide*.

You can configure point-to-multipoint LSPs for multicast VPNs as follows:

- Static point-to-multipoint LSPs—Configure static point-to-multipoint LSPs using the standard MPLS LSP statements specified at the [edit protocols mpls] hierarchy level. You manually configure each of the leaf nodes for the point-to-multipoint LSP.
- Dynamic point-to-multipoint LSPs using the default template—Configuring dynamic point-to-multipoint LSPs using the default-template option causes the leaf nodes to be discovered automatically. The leaf nodes are discovered through BGP intra-AS automatic discovery. The default-template option allows you to minimize the amount of configuration needed; however, it does not allow you to configure any of the standard MPLS options.
- Dynamic point-to-multipoint LSPs using a user-configured template—Configuring dynamic point-to-multipoint LSPs using a user-configured template also causes the leaf nodes to be discovered automatically. By creating your own template for the point-to-multipoint LSPs, all of the standard MPLS features (such as bandwidth allocation and traffic engineering) can be configured.

Be aware of the following properties for the egress PE router in a point-to-multipoint LSP configured for a multicast VPN:

- Penultimate hop-popping is not used by point-to-multipoint LSPs for multicast VPNs. Only ultimate hop-popping is used.
- You must configure either the vrf-table-label statement or a virtual loopback tunnel interface on the egress PE router.
- If you configure the vrf-table-label statement on the egress PE router and the egress PE router is also a transit router for the point-to-multipoint LSP, the penultimate hop router sends two copies of each packet over the link to the egress PE router.
- If you configure the vrf-table-label statement on the egress PE router and the egress PE router is not a transit router for the point-to-multipoint LSP, the penultimate hop router can send just one copy of each packet over the link to the egress PE router.
- If you configure a virtual loopback tunnel interface on the egress PE router and the egress PE router is also a transit router for the point-to-multipoint LSP, the penultimate hop router sends just one copy of each packet over the link to the egress PE router. A virtual loopback tunnel interface can perform two lookups on an incoming packet, one for the multicast MPLS lookup and one for the IP lookup.

The following sections describe how to configure point-to-multipoint LSPs for multicast VPNs:

- Configuring Inclusive Point-to-Multipoint LSPs on page 360
- Configuring Selective Point-to-Multipoint LSPs on page 361

Configuring Inclusive Point-to-Multipoint LSPs

You can configure inclusive point-to-multipoint LSPs for multicast VPNs. Aggregation is not supported, so you need to configure an inclusive point-to-multipoint LSP for each sender PE router in each multicast VPN routing instance. The sender PE router is in the sender site set of the multicast VPN.

To configure a static inclusive point-to-multipoint LSP, include the static-lsp statement:

```
static-lsp lsp-name;
```

You can include the static-lsp statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* provider-tunnel rsvp-te]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel rsvp-te]

To configure a dynamic inclusive point-to-multipoint LSPs, include the label-switched-path-template statement:

```
label-switched-path-template (default-template | lsp-template-name);
```

You can include the static-lsp statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* provider-tunnel rsvp-te]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel rsvp-te]

You can configure either the default-template option or manually configure a point-to-multipoint LSP template and specify the template name. For information on how to configure a point-to-multipoint LSP template, see the *JUNOS MPLS Applications Configuration Guide*.

Configuring Selective Point-to-Multipoint LSPs

You can configure selective point-to-multipoint LSPs for multicast VPNs. Selective point-to-multipoint LSPs send traffic only to the receivers configured for the multicast VPNs, helping to minimize flooding in the service provider's network.

As with inclusive point-to-multipoint LSPs, you can configure both dynamic and static selective tunnels for the multicast VPN.

To configure selective point-to-multipoint provider tunnels, include the selective statement:

```
selective {
    group address {
        source source-address {
            rsvp-te {
                label-switched-path-template (default-template | lsp-template-name);
                static-lsp lsp-name;
            }
            threshold-rate number;
        }
    }
    tunnel-limit number;
}
```

You can include these statements at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* provider-tunnel]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel]

The following sections describe how to configure selective point-to-multipoint LSPs for multicast VPNs:

- Configuring the Threshold for Dynamic Selective Point-to-Multipoint LSPs on page 363
- Configuring the Tunnel Limit for Dynamic Selective Point-to-Multipoint LSPs on page 364

Configuring the Multicast Group Address

To configure a point-to-multipoint LSP for a multicast VPN, you need to specify a multicast group address using the group statement:

```
group address { ... }
```

You can include this statements at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* provider-tunnel selective]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective]

The address must be a valid multicast group address. Multicast uses the Class D IP address range (224.0.0.0 through 239.255.255.255). For more information about multicast addresses, see the *JUNOS Multicast Protocols Configuration Guide*.

Configuring the Multicast Source Address

To configure a point-to-multipoint LSP for a multicast VPN, you need to specify a multicast source address using the source statement:

```
source address { ... }
```

You can include this statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* provider-tunnel selective group *address*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective group *address*]

Configuring Static Selective Point-to-Multipoint LSPs

You can configure a static selective point-to-multipoint LSP for a multicast VPN. You need to configure a static LSP using the standard MPLS LSP statements at the [edit protocols mpls] hierarchy level. You then include the static LSP in your selective point-to-multipoint LSP configuration using the static-lsp statement. Once this functionality is enabled on the source PE router, the static point-to-multipoint LSP is created based on your configuration.

To configure a static selective point-to-multipoint LSP, include the rsvp-te and the static-lsp statements:

```
rsvp-te static-lsp lsp-name;
```

You can include these statements at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* provider-tunnel selective group *address* source *source-address*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective group *address* source *source-address*]

Configuring Dynamic Selective Point-to-Multipoint LSPs

You can configure a dynamic selective point-to-multipoint LSP for a multicast VPN. The leaf nodes for a dynamic point-to-multipoint LSP can be automatically discovered using leaf automatic discovery routes. Selective provider multicast service interface (S-PMSI) automatic discovery routes are also supported.

To configure a dynamic selective point-to-multipoint provider tunnel, include the rsvp-te and label-switched-path-template statements:

```
rsvp-te label-switched-path-template (default-template | lsp-template-name);
```

You can include these statements at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* provider-tunnel selective group *address* source *source-address*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective group *address* source *source-address*]

The label-switched-path-template statement includes the following options:

- default-template—Specify that point-to-multipoint LSPs are generated dynamically based on the the default template. No user configuration is required for the LSPs. However, the automatically generated LSPs include none of the common LSP features, such as bandwidth allocation and traffic engineering.
- *lsp-template-name*—Specify the name of an LSP template to be used for the point-to-multipoint LSP. You need to configure the LSP template to be used as a basis for the point-to-multipoint LSPs. You can configure any of the common LSP features for this template.

Configuring the Threshold for Dynamic Selective Point-to-Multipoint LSPs

To configure a selective point-to-multipoint LSP dynamically, you need to specify the data threshold (in kilobytes) required before a new tunnel is created using the threshold-rate statement:

```
threshold-rate number;
```

You can include this statements at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* provider-tunnel selective group *address* source *source-address*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective group *address* source *source-address*]

Configuring the Tunnel Limit for Dynamic Selective Point-to-Multipoint LSPs

To configure a limit on the number of tunnels that can be generated for a dynamic point-to-multipoint LSP, include the tunnel-limit statement:

```
tunnel-limit number;
```

You can include the tunnel-limit statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* provider-tunnel selective]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective]

Tracing Multicast VPN Traffic and Operations

To trace multicast VPN traffic, you can specify options with the traceoptions statement:

```
traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
}
```

You can include this statement at the following hierarchy levels:

- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvpn]
- [edit routing-instances *routing-instance-name* protocols mvpn]

The following trace flags display the operations associated with multicast VPNs:

- all—All multicast VPN tracing options
- error—Error conditions
- general—General events
- nlri—Multicast VPN advertisements received or sent by means of BGP
- normal—Normal events
- policy—Policy processing
- route—Routing information
- state—State transitions
- task—Routing protocol task processing
- timer—Routing protocol timer processing
- topology—Multicast VPN topology changes caused by reconfiguration or advertisements received from other PE routers using BGP

Chapter 17

Summary of Multicast VPN Configuration Statements

The following sections explain the configuration statements that apply specifically to multicast virtual private networks (VPNs). The statements are arranged alphabetically.

export-target

Syntax

```
export-target {
    target target-community;
    unicast;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvpn route-target],
[edit routing-instances *routing-instance-name* protocols mvpn route-target]

Release Information Statement introduced in JUNOS Release 8.4.

Description Enable you to override the Layer 3 VPN import and export route targets used for importing and exporting routes for the multicast VPN network layer reachability information (NLRI).

Options target *target-community*—Specify the export target community.

unicast—Use the same target community as specified for unicast.

Usage Guidelines See "Configuring the Export Target for the Multicast VPN" on page 357.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

group

Syntax

```
group address {
   source source-address {
      rsvp-te {
         label-switched-path-template (default-template | lsp-template-name);
         static-lsp lsp-name;
      }
      threshold-rate number;
   }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective],
[edit routing-instances *routing-instance-name* provider-tunnel selective]

Release Information

Statement introduced in JUNOS Release 8.5.

Description

Enable you to specify the IP address for the multicast group configured for point-to-multipoint label-switched paths (LSPs).

Options

address—Specify the IP address for the multicast group. This address must be a valid multicast group address.

The remaining statements are explained separately.

Usage Guidelines

See "Configuring the Multicast Group Address" on page 362.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

import-target

Syntax

```
import-target {
    target {
        target-value;
        receiver target-value;
        sender target-value;
    }
    unicast {
        receiver;
        sender;
    }
}
```

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvpn route-target],
[edit routing-instances *routing-instance-name* protocols mvpn route-target]

Release Information Statement introduced in JUNOS Release 8.4.

Description Enable you to override the Layer 3 VPN import and export route targets used for importing and exporting routes for the multicast VPN NLRI.

Options The remaining statements are explained separately.

Usage Guidelines See "Configuring the Import Target for the Multicast VPN" on page 357.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

inet-mvpn

Syntax

```
inet-mvpn;
```

Hierarchy Level [edit logical-systems *logical-system-name* protocols bgp family],
[edit protocols bgp family]

Release Information Statement introduced in JUNOS Release 8.4.

Description Enable the inet-mvpn address family in BGP.

Usage Guidelines See "Configuring NLRI Parameters for Multicast VPN" on page 358.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

inet6-mvpn

Syntax inet6-mvpn;

Hierarchy Level [edit logical-systems *logical-system-name* protocols bgp family],
[edit protocols bgp family]

Description Enable the inet6-mvpn address family in BGP.

Usage Guidelines See "Configuring NLRI Parameters for Multicast VPN" on page 358.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

label-switched-path-template

Syntax label-switched-path-template (default-template | *lsp-template-name*);

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel rsvp-te],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective group *address* source *source-address* rsvp-te],
[edit routing-instances *routing-instance-name* provider-tunnel rsvp-te]
[edit routing-instances *routing-instance-name* provider-tunnel selective group *address* source *source-address* rsvp-te]

Release Information Statement introduced in JUNOS Release 8.5.

Description Enable you to specify the LSP template used for the point-to-multipoint LSP. You can use either the default template or manually configure the properties of the LSP template.

Options default-template—Specify that the default template be used for the point-to-multipoint LSP.

lsp-template-name—Specify the name of an LSP template to be used for the point-to-multipoint LSP.

Usage Guidelines See "Configuring Inclusive Point-to-Multipoint LSPs" on page 360 and "Configuring Dynamic Selective Point-to-Multipoint LSPs" on page 363.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

mvpn

Syntax

```
mvpn {
    receiver-site;
    sender-site;
    route-target {
        export-target {
            target target-community;
            unicast;
        }
        import-target {
            target {
                target-value;
                receiver target-value;
                sender target-value;
            }
            unicast {
                receiver;
                sender;
            }
        }
    }
}
```

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols],
[edit routing-instances *routing-instance-name* protocols]

Release Information Statement introduced in JUNOS Release 8.4.

Description Enable next-generation multicast VPNs in a routing instance.

Options receiver-site—Allow sites with multicast receivers.

sender-site—Allow sites with multicast senders.

The remaining statements are explained separately.

Usage Guidelines See “Configuring the Multicast VPN Routing Instance” on page 354

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

pim-asm

Syntax

```
pim-asm {
    group-address address;
}
```

Hierarchy Level

```
[edit logical-systems logical-system-name routing-instances routing-instance-name
    provider-tunnel],
[edit routing-instances routing-instance-name provider-tunnel]
```

Release Information Statement introduced in JUNOS Release 8.3.

Description Specify a Protocol Independent Multicast (PIM) sparse mode provider tunnel for a multicast VPN.

Options group-address *address*—PIM sparse mode provider tunnel group address.

Usage Guidelines See “Configuring PIM Provider Tunnels for Multicast VPNs” on page 359.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

provider-tunnel

Syntax

```
provider-tunnel {
    pim-asm group-address address;
    rsvp-te {
        label-switched-path-template (default-template | lsp-template-name);
        static-lsp lsp-name;
    }
    selective {
        group address {
            source source-address {
                rsvp-te {
                    label-switched-path-template (default-template | lsp-template-name);
                    static-lsp lsp-name;
                }
                threshold-rate number;
            }
        }
        tunnel-limit number;
    }
}
```

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*], [edit routing-instances *routing-instance-name*]

Release Information Statement introduced in JUNOS Release 8.3. The selective statement and substatements were added in JUNOS Release 8.5.

Description Enables you to configure virtual private LAN service (VPLS) flooding of unknown unicast, broadcast, and multicast traffic using point-to-multipoint (P2MP) LSPs. Also enables you to configure P2MP LSPs for multicast VPNs.

Options The remaining statements are explained separately.

Usage Guidelines See "Flooding Unknown Traffic Using Point-to-Multipoint LSPs" on page 422 and "Configuring Inclusive Point-to-Multipoint LSPs" on page 360.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

route-target

Syntax

```
route-target {
    export-target {
        target target-community;
        unicast;
    }
    import-target {
        target {
            target-value;
            receiver target-value;
            sender target-value;
        }
        unicast {
            receiver;
            sender;
        }
    }
}
```

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvpn],
[edit routing-instances *routing-instance-name* protocols mvpn]

Release Information Statement introduced in JUNOS Release 8.4.

Description Enable you to override the Layer 3 VPN import and export route targets used for importing and exporting routes for the multicast VPN NLRI.

Default The multicast VPN routing instance uses the import and export route targets configured for the Layer 3 VPN.

Options The remaining statements are explained separately.

Usage Guidelines See "Configuring a Route Target for the Multicast VPN Routing Instance" on page 355.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

rsvp-te

Syntax

```
rsvp-te {
    label-switched-path-template (default-template | lsp-template-name);
    static-lsp lsp-name;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective group *address* source *source-address*],
[edit routing-instances *routing-instance-name* provider-tunnel],
[edit routing-instances *routing-instance-name* provider-tunnel selective group *address* source *source-address*]

Release Information

Statement introduced in JUNOS Release 8.5.

Description

Enables you to configure the properties of the RSVP traffic engineered point-to-multipoint LSP for multicast VPNs.

The remaining statements are explained separately.

Usage Guidelines

See "Configuring Inclusive Point-to-Multipoint LSPs" on page 360 and "Configuring Selective Point-to-Multipoint LSPs" on page 361.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

selective

Syntax

```
selective {
    group address {
        source source-address {
            rsvp-te {
                label-switched-path-template (default-template | lsp-template-name);
                static-lsp lsp-name;
            }
            threshold-rate number;
        }
    }
    tunnel-limit number;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel],
[edit routing-instances *routing-instance-name* provider-tunnel]

Release Information

Statement introduced in JUNOS Release 8.5.

Description

Enables you to configure selective point-to-multipoint LSPs for a multicast VPN. Selective point-to-multipoint LSPs send traffic only to the receivers configured for the multicast VPNs, helping to minimize flooding in the service provider's network.

The remaining statements are explained separately.

Usage Guidelines

See "Configuring Selective Point-to-Multipoint LSPs" on page 361.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

source

Syntax

```
source source-address {
    rsvp-te {
        label-switched-path-template (default-template | lsp-template-name);
        static-lsp lsp-name;
    }
    threshold-rate number;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective group *address*],
[edit routing-instances *routing-instance-name* provider-tunnel selective group *address*]

Release Information

Statement introduced in JUNOS Release 8.5.

Description

Enables you to specify the IP address for the multicast source. This statement is a part of the point-to-multipoint LSP configuration required for multicast VPNs.

Options

source-address—IP address for the multicast source.

The remaining statements are explained separately.

Usage Guidelines

See "Configuring the Multicast Source Address" on page 362.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

static-lsp

Syntax static-lsp *lsp-name*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel rsvp-te],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective group *address* source *source-address* rsvp-te],
[edit routing-instances *routing-instance-name* provider-tunnel rsvp-te],
[edit routing-instances *routing-instance-name* provider-tunnel selective group *address* source *source-address* rsvp-te]

Release Information Statement introduced in JUNOS Release 8.5.

Description Specify the name of the static point-to-multipoint LSP used for a multicast VPN. Use this statement to specify the static LSP for both inclusive and selective point-to-multipoint LSPs.

Usage Guidelines See "Configuring Inclusive Point-to-Multipoint LSPs" on page 360 and "Configuring Selective Point-to-Multipoint LSPs" on page 361.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

target

Syntax
```
target {
    target-value;
    receiver target-value;
    sender target-value;
}
```

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvpn route-target import-target],
[edit routing-instances *routing-instance-name* protocols mvpn route-target import-target]

Release Information Statement introduced in JUNOS Release 8.4.

Description Enable you to specify the target value when importing sender and receiver site routes.

Options receiver *target-value*—Specify the target community used when importing receiver site routes.

sender *target-value*—Specify the target community used when importing sender site routes.

Usage Guidelines See "Configuring the Import Target Receiver and Sender" on page 357.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

threshold-rate

Syntax threshold-rate *number*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective group *address* source *source-address*],
[edit routing-instances *routing-instance-name* provider-tunnel selective group *address* source *source-address*]

Release Information Statement introduced in JUNOS Release 8.5.

Description Specifies the data threshold required before a new tunnel is created for a dynamic selective point-to-multipoint LSP. This statement is part of the configuration for point-to-multipoint LSPs for multicast VPNs.

Options *number*—Specifies the data threshold required before a new tunnel is created.
Range: 0 through 1,000,000 KB

Usage Guidelines See "Configuring the Threshold for Dynamic Selective Point-to-Multipoint LSPs" on page 363.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

traceoptions

Syntax

```
traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvpn],
[edit routing-instances *routing-instance-name* protocols mvpn]

Release Information

Statement introduced in JUNOS Release 8.4.

Description

Trace traffic flowing through a multicast VPN.

Options

disable—(Optional) Disable the tracing operation. You can use this option to disable a single operation when you have defined a broad group of tracing operations, such as all.

file *filename*—Name of the file to receive the output of the tracing operation. Enclose the name in quotation marks (" ").

files *number*—(Optional) Maximum number of trace files. When a trace file named *trace-file* reaches this size, it is renamed *trace-file*.0. When *trace-file* again reaches its maximum size, *trace-file*.0 is renamed *trace-file*.1 and *trace-file* is renamed *trace-file*.0. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum number of files, you also must specify a maximum file size with the size option.
Range: 2 through 1000 files
Default: 2 files

flag *flag*—Tracing operation to perform. To specify more than one tracing operation, include multiple flag statements. You can specify any of the following flags:

- all—All multicast VPN tracing options
- error—Error conditions
- general—General events
- nlri—Multicast VPN advertisements received or sent by means of the BGP
- normal—Normal events
- policy—Policy processing
- route—Routing information
- state—State transitions
- task—Routing protocol task processing
- timer—Routing protocol timer processing

- topology—Multicast VPN topology changes caused by reconfiguration or advertisements received from other provider edge (PE) routers using BGP

flag-modifier—(Optional) Modifier for the tracing flag. You can specify the following modifiers:

- detail—Provide detailed trace information
- disable—Disable the tracing flag
- receive—Trace received packets
- send—Trace sent packets

no-world-readable—Do not allow any user to read the log file.

size *size*—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named *trace-file* reaches this size, it is renamed *trace-file*.0. When *trace-file* again reaches its maximum size, *trace-file*.0 is renamed *trace-file*.1 and *trace-file* is renamed *trace-file*.0. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum file size, you also must specify a maximum number of trace files with the files option.
Syntax: *x*k to specify kilobytes, *x*m to specify megabytes, or *x*g to specify gigabytes
Range: 10 KB through the maximum file size supported on your system
Default: 1 MB

world-readable—Allow any user to read the log file.

Usage Guidelines See "Tracing Multicast VPN Traffic and Operations" on page 364.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

tunnel-limit

Syntax tunnel-limit *number*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective],
[edit routing-instances *routing-instance-name* provider-tunnel selective]

Release Information Statement introduced in JUNOS Release 8.5.

Description Enables you to specify a limit on the number of tunnels that can be created for a point-to-multipoint LSP.

Options *number*—Specify the tunnel limit.
Range: 0 through 1024

Usage Guidelines See "Configuring the Tunnel Limit for Dynamic Selective Point-to-Multipoint LSPs" on page 364.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

unicast

Syntax
```
unicast {
    receiver;
    sender;
}
```

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvpn route-target import-target],
[edit routing-instances *routing-instance-name* protocols mvpn route-target import-target]

Release Information Statement introduced in JUNOS Release 8.4.

Description Specify the same target community configured for unicast.

Options receiver—Specify the unicast target community used when importing receiver site routes.

sender—Specify the unicast target community used when importing sender site routes.

Usage Guidelines See "Configuring the Import Target Unicast Parameters" on page 358.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Part 5
VPLS

Chapter 18

VPLS Overview

This chapter provides an overview of virtual private LAN service (VPLS) as it is implemented in the JUNOS software.

For information about virtual private networks (VPNs) and the differences between Layer 2 VPNs, Layer 3 VPNs, and VPLS, see “VPN Overview” on page 3.

This chapter discusses the following topics that provide background information about VPLS:

VPLS Overview

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

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

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

The paths carrying VPLS traffic between each PE router participating in a routing instance are called pseudowires. The pseudowires are signaled using either BGP or LDP.

VPLS Standards

VPLS is described in the following internet draft and RFC:

- Internet draft draft-ietf-l2vpn-vpls-bgp-08.txt, *Virtual Private LAN Service (VPLS) Using BGP for Auto-discovery and Signaling* (expires December 2006).
- RFC 4762 (FEC 128, control bit 0, and Ethernet pseudowire type hexadecimal 0x0005), *Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling*.

You can access Internet RFCs and drafts on the IETF Web site at http://www.ietf.org.

Supported Platforms and PICs

VPLS is supported on the following M-series platforms:

- M5
- M7i
- M10
- M10i
- M20
- M40
- M40e
- M120
- M320

VPLS is also supported on all T-series platforms and the MX-series platforms.

VPLS is supported on the following PICs:

- All ATM2 IQ PICs
- 4-port Fast Ethernet PIC with 10/100 Base-TX interfaces PIC
- 1-port, 2-port, and 10-port Gigabit Ethernet PICs
- 1-port, 2-port, and 4-port Gigabit Ethernet PICs with SFP
- 1-port 10 Gigabit Ethernet PIC
- 1-port and 2-port Gigabit Ethernet Intelligent Queuing (IQ) PICs
- 4-port and 8-port Gigabit Ethernet IQ2 PICs with SFP
- 1-port 10 Gigabit Ethernet IQ2 PIC with XFP
- 4-port, quad-wide Gigabit Ethernet PIC

VPLS Routing and Virtual Ports

Because a VPLS carries Ethernet traffic across a service provider network, it must mimic an Ethernet network in some ways. When a PE router configured with a VPLS routing instance receives a packet from a CE device, it first determines whether it has the destination of the VPLS packet in the appropriate routing table. If it does, it forwards the packet to the appropriate PE router or CE device. If it does not, it broadcasts the packet to all other PE routers and CE devices that are members of that VPLS routing instance. In both cases, the CE device receiving the packet must be different from the one sending the packet.

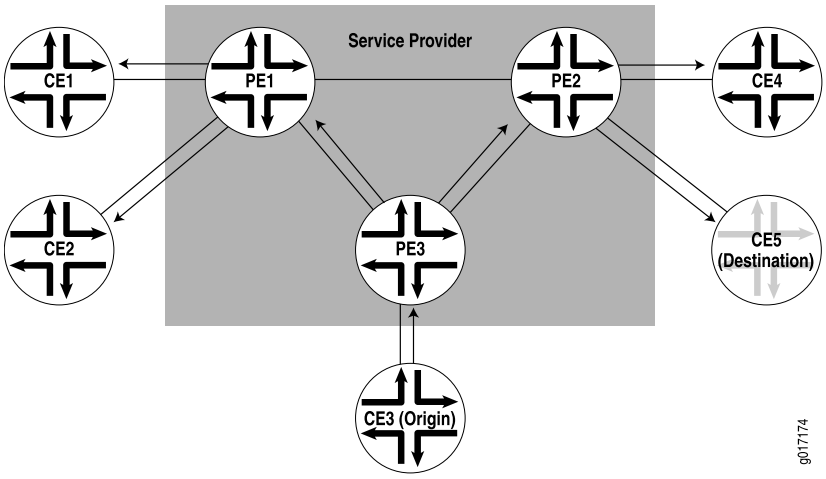
When a PE router receives a packet from another PE router, it first determines whether it has the destination of the VPLS packet in the appropriate routing table. If it does, the PE router either forwards the packet or drops it depending on whether the destination is a local or remote CE device:

- If the destination is a local CE device, the PE router forwards the packet to it.
- If the destination is a remote CE device (connected to another PE router), the PE router discards the packet.

If the PE router cannot determine the destination of the VPLS packet, it floods the packet to all attached CE devices.

This process is illustrated in Figure 45 on page 385.

Figure 45: Flooding a Packet with an Unknown Destination to All PE Routers in the VPLS Instance



A VPLS can be directly connected to an Ethernet switch. Layer 2 information gathered by an Ethernet switch (for example, media access control [MAC] addresses and interface ports) is included in the VPLS routing instance table. However, instead of all VPLS interfaces being physical switch ports, the router allows remote traffic for a VPLS instance to be delivered across an MPLS 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 in almost the same way as traffic is sent to a local port.

The VPLS routing table learns MAC address and interface information for both physical and virtual ports. The main difference between a physical port and a virtual port is that the router captures additional information from the 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. The virtual port is generated dynamically on a Tunnel Services Physical Interface Card (PIC) when you configure VPLS on the router.

You can also configure VPLS without a Tunnel Services PIC. To do so, you use a label-switched interface (LSI) to provide VPLS functionality. An LSI MPLS label is used as the inner label for VPLS. This label maps to a VPLS routing instance. On the PE router, the LSI label is stripped and then mapped to a logical LSI interface. The Layer 2 Ethernet frame is then forwarded using the LSI interface to the correct VPLS routing instance.

One restriction on 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. However, if a CE Ethernet switch has two or more connections to the same PE router, you must enable the Spanning Tree Protocol (STP) on the CE switch to prevent loops. STP is supported on MX-series routers only.

The JUNOS software allows standard Bridge Protocol Data Unit (BPDU) frames to pass through emulated Layer 2 connections, such as those configured with Layer 2 VPNs, Layer 2 circuits, and VPLS routing instances. However, CE Ethernet switches that generate proprietary BPDU frames might not be able to run STP across Juniper Networks routing platforms configured for these emulated Layer 2 connections.

NOTE: Under certain circumstances, VPLS Provider routers might duplicate an Internet Control Message Protocol (ICMP) reply from a CE router when a PE router has to flood an ICMP request because the destination MAC address has not yet been learned. The duplicate ICMP reply can be triggered when a CE router with promiscuous mode enabled is connected to a PE router. The PE router automatically floods the promiscuous mode enabled CE router, which then returns the ICMP request to the VPLS Provider routers. The VPLS Provider routers consider the ICMP request to be new and flood the request again, creating a duplicate ping reply.

VPLS and Aggregated Ethernet Interfaces

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.

Forwarding is based on a lookup of the DA MAC address. For the remote site, if a packet needs to be forwarded over an LSP, the packet is encapsulated and forwarded through the LSP. If the packet destination is a local site, it is forwarded over appropriate local site interface. For an aggregated Ethernet interface on the local site, packets are sent out of the load-balanced child interface. The Packet Forwarding Engine acquires the child link to transmit the data.

When a received packet does not have a match to a MAC address in the forwarding database, the packet is forwarded over a set of interfaces determined from a lookup

in the flooding database based on the incoming interface. This is denoted by a flood next hop. The flood next hop can include the aggregated Ethernet interface as the set of interfaces to flood the packet.

Each VPLS routing instance configured on a PE router has its own forwarding database entries that associate all of the MAC addresses the VPLS routing instance acquires with each corresponding port. A route is added to the kernel with a MAC address as the prefix and the next hop used to reach the destination. The route is an interface if the destination is local. For a remote destination, the route is a next hop for the remote site.

For local aggregated Ethernet interfaces on M-series and T-series routers, learning is based on the parent aggregated Ethernet logical interface. To age out MAC addresses for aggregated Ethernet interfaces, each Packet Forwarding Engine is queried to determine where the individual child interfaces are located. MAC addresses are aged out based on the age of the original interface.

For MX-series routers, when a Dense Port Concentrator (DPC) learns a MAC address it causes the Routing Engine to age out the entry. This behavior applies to all logical interfaces. For an aggregated Ethernet logical interface, once all the member DPCs have aged out the entry, the entry is deleted from the Routing Engine.

For information on how to configure aggregated Ethernet interfaces for VPLS routing instances, see “Configuring Aggregated Ethernet Interfaces for VPLS” on page 409.

VPLS Multihoming

VPLS multihoming allows you to connect a customer site to multiple PE routers to provide redundant connectivity while preventing the formation of Layer 2 loops in the service provider’s network. A VPLS site multihomed to two or more PE routers provides redundant connectivity in the event of a PE router-to-CE device link failure or the failure of a PE router.

When multihoming a VPLS site (potentially in different autonomous systems [ASs]), the PE routers connected to the same site can either be configured with the same VPLS edge (VE) device identifier or with different VE device identifiers. In the latter case, you must run STP on the CE device, and possibly on the PE routers, to construct a loop-free VPLS topology.

If the PE routers are connected to the same site and assigned the same VE device identifier, a loop-free topology is constructed using a routing mechanism such as BGP path selection. When a BGP speaker receives two equivalent network layer reachability information (NLRI) advertisements, it applies standard path selection criteria such as local preference and AS path length to determine which NLRI to choose; it selects only one.

Because a PE router picks one of the received NLRI advertisements with a particular VE device identifier, it establishes pseudowires to only one of the remote PE routers, the PE router that originated the winning advertisement. This prevents multiple paths from being created in the network between sites, preventing the formation of Layer 2 loops in the network. If the selected PE router fails, all PE routers in the network automatically switch to the backup PE router and establish pseudowires through the backup PE router.

Two VPLS NLRIs are considered equivalent from a path selection perspective if the following are the same:

- Route distinguisher
- VE device identifier
- VE block offset

If two PE routers are assigned the same VE device identifier in a given VPLS, they must also advertise the same VE block size for a given VE offset. The PE routers can be configured with the same route distinguisher or with distinct route distinguishers.

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.

NOTE: Traffic loss can occur when the old pseudowires are brought down and new ones established.

Interoperability between BGP Signaling and LDP Signaling in VPLS

You can configure a VPLS routing instance where some of the PE routers use BGP for signaling and some use LDP for signaling.

The following concepts form the basis of the configuration needed to include both BGP-signaled and LDP-signaled PE routers in a VPLS routing instance:

- PE router mesh group—Consists of a set of routers participating in a VPLS routing instance that share the same signaling protocol, either BGP or LDP, and are also fully meshed. Each VPLS routing instance can have just one BGP mesh group. However, you can configure multiple LDP mesh groups for each routing instance.
- Border router—A PE router that must be reachable by all of the other PE routers participating in a VPLS routing instance, whether they are LDP-signaled or BGP-signaled. Bidirectional pseudowires are created between the border router and all of these PE routers. The border router is aware of the composition of each PE mesh group configured as a part of the VPLS routing instance. It can also have direct connections to local CE routers, allowing it to act as a typical PE router in a VPLS routing instance.

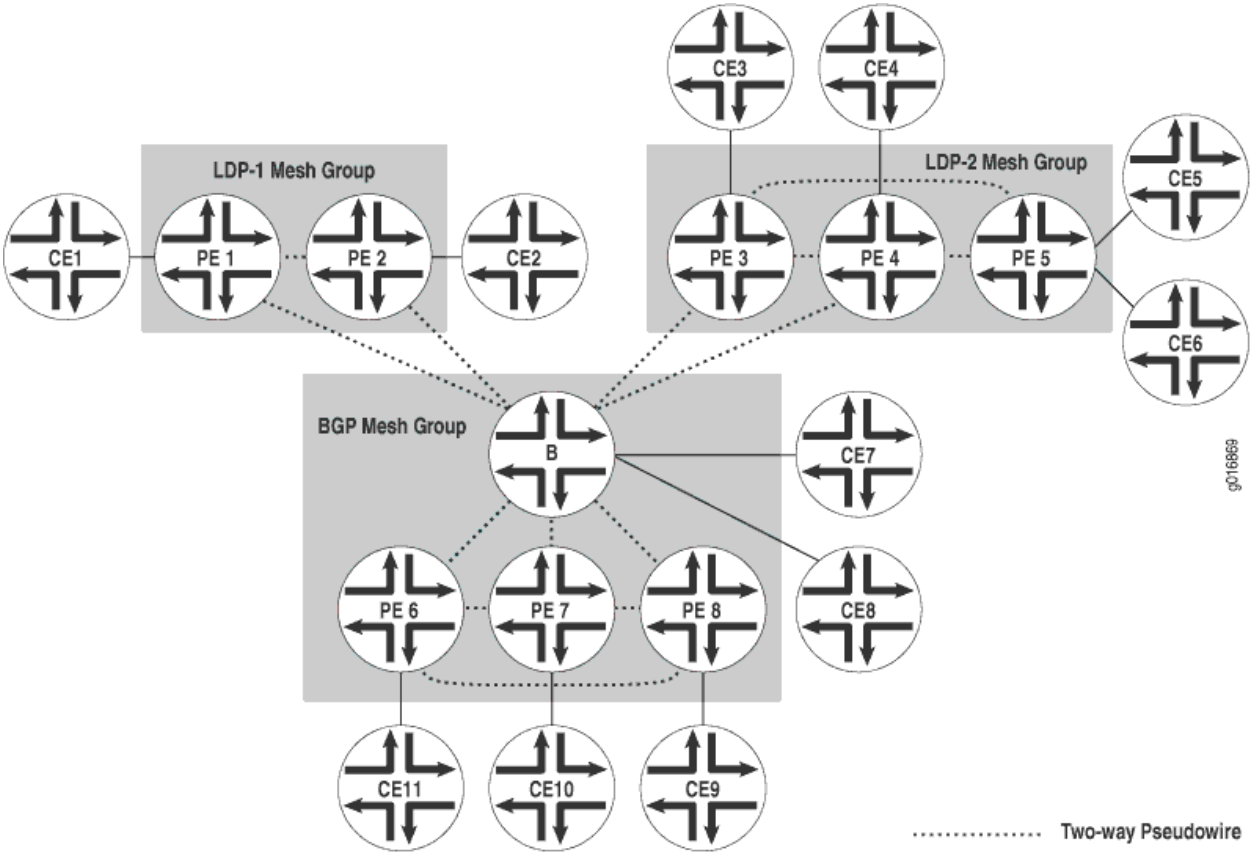
The following sections describe how the LDP-signaled and BGP-signaled PE routers function when configured to interoperate within a VPLS routing instance:

- LDP-Signaled and BGP-Signaled PE Router Topology on page 389
- Flooding Unknown Packets Across Mesh Groups on page 390
- Unicast Packet Forwarding on page 390

LDP-Signaled and BGP-Signaled PE Router Topology

Figure 46 on page 389 illustrates a topology for a VPLS routing instance configured to support both BGP and LDP signaling. Router B is the border router. 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, PE8, and router B (the border router) are in the BGP-signaled mesh group. The border router also acts as a standard VPLS PE router (having local connections to CE routers). All of the PE routers shown are within the same VPLS routing instance.

Figure 46: BGP and LDP Signaling for a VPLS Routing Instance



Two-way pseudowires are established between the PE routers in each mesh group and between each PE router in the VPLS routing instance and the border router. In Figure 46 on page 389, two-way pseudowires are established between routers PE1 and PE2 in mesh group LDP-1, routers PE3, PE4, and PE5 in mesh group LDP-2, and routers PE6, PE7, and PE8 in the BGP mesh group. Routers PE1 through PE8 also all have two-way pseudowires to the Border router. Based on this topology, the LDP-signaled routers are able to interoperate with the BGP-signaled routers. Both the LDP-signaled and BGP-signaled PE routers can logically function within a single VPLS routing instance.

NOTE: The following features are not supported for VPLS routing instances configured with both BGP and LDP signaling:

- Point-to-multipoint LSPs
- Integrated routing and bridging
- IGMP snooping

Flooding Unknown Packets Across Mesh Groups

Broadcast, multicast, and unicast packets of unknown origin received from a PE router are flooded to all local CE routers. They are also flooded to all of the PE routers in the VPLS routing instance except the PE routers that are a part of the originating PE router mesh group.

For example, if a multicast packet is received by the border router in Figure 46 on page 389, it is flooded to the two local CE routers. It is also flooded to routers PE1 and PE2 in the LDP-1 mesh group and to routers PE3, PE4, and PE5 in the LDP-2 mesh group. However, the packet is not flooded to routers PE6, PE7, and PE8 in the BGP mesh group.

Unicast Packet Forwarding

The PE border router is made aware of the composition of each PE router mesh group. From the data plane, each PE router mesh group is viewed as a virtual pseudowire LAN. The border router is configured to interconnect all of the PE router mesh groups belonging to a single VPLS routing instance. To interconnect the mesh groups, a common MAC table is created on the border router.

Unicast packets originating within a mesh group are dropped if the destination is another PE router within the same mesh group. However, if the destination MAC address of the unicast packet is a PE router located in a different mesh group, the packet is forwarded to that PE router.

PE Router Mesh Groups for VPLS Routing Instances

A PE router mesh group consists of a set of routers participating in a VPLS routing instance that share the same signaling protocol, either BGP or LDP. Each VPLS routing instance can have just one BGP mesh group. However, you can configure multiple LDP mesh groups for each routing instance.

The JUNOS software can support up to 16 mesh groups on MX-series routers and up to 128 on M-series and T-series routers. However, 2 mesh groups are created by default, 1 for the CE routers and 1 for the PE routers. Therefore, the maximum number of user-defined mesh groups is 14 for MX-series routers and 126 for M-series and T-series routers. PE router mesh groups are not supported on J-series routers.

The following describes the default behavior of mesh groups in regards to BGP-signaled PE routers discovered automatically and LDP-signaled PE routers configured statically:

- BGP-signaled PE routers—Automatically discovered PE routers that use BGP for signaling are associated with the default VE mesh group. You cannot configure the JUNOS software to associate these routers with a user-defined VE mesh group.
- LDP-signaled PE routers—PE routers statically configured using forwarding equivalence class (FEC)-128 LDP signaling are placed in a default mesh group. However, you can configure a VE mesh group and associate each LDP FEC-128 neighbor with it. Each configured VE mesh group contains a set of VEs that are in the same interior gateway protocol (IGP) routing instance and are fully meshed with each other in the control and data planes.

Chapter 19

Configuring VPLS

Virtual private LAN service (VPLS) allows you to provide a point-to-multipoint LAN between a set of sites in a virtual private network (VPN).

To configure VPLS functionality, you must enable VPLS support on the provider edge (PE) router. You must also configure PE routers to distribute routing information to the other PE routers in the VPLS and configure the circuits between the PE routers and the customer edge (CE) routers.

Each VPLS is configured under a routing instance of type vpls. A vpls routing instance can transparently carry Ethernet traffic across the service provider's network. As with other routing instances, all logical interfaces belonging to a VPLS routing instance are listed under that instance.

To configure VPLS, include the following statements:

```
description text;
forwarding-options {
    family vpls {
        filter input input-filter-name;
        flood input flood-filter-name;
    }
}
instance-type vpls;
interface interface-name;
route-distinguisher (as-number:id | ip-address:id);
vrf-export [ policy-names ];
vrf-import [ policy-names ];
vrf-target target:target-id;
protocols {
    vpls {
        active-interface {
            any;
            primary interface-name;
        }
        connectivity-type (ce | irb);
        interface-mac-limit limit;
        mac-table-aging-time time;
        mac-table-size size;
        neighbor neighbor-id;
        no-tunnel-services;
        site site-name {
            active-interface {
                any;
```

```
            primary interface-name;
        }
        interface interface-name {
            interface-mac-limit limit;
        }
        multi-homing;
        site-identifier identifier;
        site-preference preference-value;
    }
    site-range number;
    traceoptions {
        file filename <files number> <size size> <world-readable | no-world-readable>;
        flag flag <flag-modifier> <disable>;
    }
    tunnel-services {
        devices device-names;
        primary primary-device-name;
    }
    vpls-id vpls-id;
  }
}
provider-tunnel {
  rsvp-te {
    label-switched-path-template (default-template | lsp-template-name);
    static-lsp lsp-name;
  }
}
```

You can include these statements at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*]

For VPLS, only some of the statements in the [edit routing-instances] hierarchy are valid. For the full hierarchy, see the *JUNOS Routing Protocols Configuration Guide*.

In addition to these statements, you must configure Multiprotocol Label Switching (MPLS) label-switched paths (LSPs) between the PE routers, internal BGP (IBGP) sessions between the PE routers, and an interior gateway protocol (IGP) on the PE and provider (P) routers.

By default, VPLS is disabled.

Many configuration procedures for VPLS are identical to the procedures for Layer 2 VPNs and Layer 3 VPNs. These procedures are described in detail in “Configuring VPNs” on page 13 and include the following:

This chapter describes how to configure VPLS, discussing the following topics:

Configuring the VPLS Routing Instance

To configure a VPLS routing instance, include the vpls statement:

```
vpls {
    active-interface {
        any;
        primary interface-name;
    }
    connectivity-type (ce | irb);
    interface-mac-limit limit;
    mac-table-aging-time time;
    mac-table-size size;
    neighbor neighbor-id;
    no-tunnel-services;
    site site-name {
        active-interface {
            any;
            primary interface-name;
        }
        interface interface-name {
            interface-mac-limit limit;
        }
        multi-homing;
        site-identifier identifier;
        site-preference preference-value;
    }
    site-range number;
    traceoptions {
        file filename <files number> <size size> <world-readable | no-world-readable>;
        flag flag <flag-modifier> <disable>;
    }
    tunnel-services {
        devices device-names;
        primary primary-device-name;
    }
    vpls-id vpls-id;
}
```

You can include this statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols]

The configuration for the VPLS routing instance statements is explained in the following sections:

Configuring BGP Signaling for VPLS

You can configure BGP signaling for the VPLS routing instance. BGP is used to signal the pseudowires linking each of the PE routers participating in the VPLS routing instance. The pseudowires carry VPLS traffic across the service provider's network between the VPLS sites.

NOTE: You cannot configure both BGP signaling and LDP signaling for the same VPLS routing instance. If you attempt to configure the statements that enable BGP signaling for the VPLS routing instance (the site, site-identifier, and site-range statements) and the statements that enable LDP signaling for the same instance (the neighbor and vpls-id statements), the commit operation fails.

Configure BGP signaling for the VPLS routing instance by completing the steps in the following sections:

Configuring the VPLS Site Name and Site Identifier

When you configure BGP signaling for the VPLS routing instance, on each PE router you must configure each VPLS site that has a connection to the PE router. All the Layer 2 circuits provisioned for a VPLS site are listed as the set of logical interfaces (using the interface statement) within the site statement.

You must configure a site name and site identifier for each VPLS site.

To configure the site name and the site identifier, include the site and the site-identifier statements:

```
site site-name {
    interface interface-name {
        interface-mac-limit limit;
    }
    site-identifier identifier;
}
```

The numerical identifier can be any number from 1 through 65,534 that uniquely identifies the VPLS site.

You can include these statements at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols vpls]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls]

Configuring Automatic Site Identifiers for VPLS

When you enable automatic site identifiers, the JUNOS software automatically assigns site identifiers to VPLS sites. To configure automatic site identifiers for a VPLS routing instance, include the automatic-site-id statement:

```
automatic-site-id {
    collision-detect-time seconds;
    new-site-wait-time seconds;
    reclaim-wait-time minimum seconds maximum seconds;
    startup-wait-time seconds;
}
```

You can include this statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols vpls site *site-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls site *site-name*]

The automatic-site-id statement includes a number of options that control different delays in network layer reachability information (NLRI) advertisements. All of these options are configured with default values. See the statement summary for the automatic-site-id statement for more information.

The automatic-site-id statement includes the following options:

- collision-detect-time—The time in seconds to wait after a claim advertisement is sent to the other routers in a VPLS instance before a PE router can begin using a site identifier. If the PE router receives a competing claim advertisement for the same site identifier during this time period, it initiates the collision resolution procedure for site identifiers.
- new-site-wait-time—The time in seconds to wait to receive VPLS information for a newly configured routing instance or a new site. This time interval is also applied whenever the automatic site identifier feature is activated on a VPLS routing instance other than at startup. Effectively, this timer indicates how long to wait before an attempt is made to allocate a site identifier. This timer is also triggered whenever a VPLS routing instance is enabled.
- reclaim-wait-time—The time to wait before attempting to claim a site identifier after a collision. A collision occurs whenever an attempt is made to claim a site identifier by two separate VPLS sites.
- startup-wait-time—The time in seconds to wait at startup to receive all the VPLS information for the route targets configured on the other PE routers included in the VPLS routing instance.

Configuring the Site Range

When you enable BGP signaling for each VPLS routing instance, you need to configure a site range. The site range specifies the total number of sites in the VPLS. To configure a site range, include the site-range statement:

```
site-range number;
```

You can include the site-range statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols vpls]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls]

NOTE: When you configure the site range, you need to specify a value that is greater than the site identifier. The following configuration example illustrates this issue:

```
protocols {
   vpls {
      site-range 20;
      no-tunnel-services;
      site sample {
         site-identifier 3;
      }
   }
}
```

This configuration is valid. However, if the site identifier was a value greater than 20, the VPLS connection would fail.

Configuring the VPLS Site Interfaces

All the Layer 2 circuits you configure for a VPLS site are listed as a set of logical interfaces within the VPLS site configuration.

To configure a logical interface for the VPLS site, include the interface statement:

```
interface interface-name {
   interface-mac-limit limit;
}
```

You can include these statements at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols vpls site *site-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls]

You can also configure a limit on the number of MAC addresses that can be learned from the specified interface. See “Limiting the Number of MAC Addresses Learned from an Interface” on page 403 for more information.

Configuring the VPLS Site Preference

You can specify the preference value advertised for a particular VPLS site. The site preference value is encoded in the BGP local preference attribute. When a PE router receives multiple advertisements with the same VPLS edge (VE) device identifier, the advertisement with the highest local preference value is preferred.

To configure the VPLS site preference, include the site-preference statement:

```
site-preference preference-value;
```

You can also specify either the backup option or the primary option for the site-preference statement. The backup option specifies the preference value as 1,

the lowest possible value, ensuring that the VPLS site is the least likely to be selected. The primary option specifies the preference value as 65,535, the highest possible value, ensuring that the VPLS site is the most likely to be selected.

For a list of hierarchy levels at which you can configure the site-preference statement, see the statement summary section for this statement.

Configuring LDP Signaling for VPLS

You can configure LDP as the signaling protocol for a VPLS routing instance. This functionality is described in RFC 4762, *Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling*.

The JUNOS software does not support all of RFC 4762. When enabling LDP signaling for a VPLS routing instance, network engineers should be aware that only the following values are supported:

- FEC—128 (supports only FEC-128)
- Control bit—0
- Ethernet pseudowire type—0x0005 (hexadecimal)

To enable LDP signaling for the set of PE routers participating in the same VPLS routing instance, you need to use the vpls-id statement configured at the [edit routing-instances *routing-instance-name* protocols vpls] hierarchy level to configure the same VPLS identifier on each of the PE routers. The VPLS identifier must be globally unique. When each VPLS routing instance (domain) has a unique VPLS identifier, it is possible to configure multiple VPLS routing instances between a given pair of PE routers.

LDP signaling requires that you configure a full-mesh LDP session between the PE routers in the same VPLS routing instance. Neighboring PE routers are statically configured. Tunnels are created between the neighboring PE routers to aggregate traffic from one PE router to another. Pseudowires are then signaled to demultiplex traffic between VPLS routing instances. These PE routers exchange the pseudowire label, the MPLS label that acts as the VPLS pseudowire demultiplexer field, by using LDP forwarding equivalence classes (FECs). Tunnels based on both MPLS and generic routing encapsulation (GRE) are supported.

NOTE: You cannot configure both BGP signaling and LDP signaling for the same VPLS routing instance. If you attempt to configure the statements that enable BGP signaling for the VPLS routing instance (the site, site-identifier, and site-range statements), and the statements that enable LDP signaling for the same instance, neighbor and vpls-id, the commit operation fails.

To enable LDP signaling for the VPLS routing instance, complete the steps in the following sections:

- Configuring LDP Signaling for the VPLS Routing Instance on page 401
- Configuring LDP Signaling on the Router on page 401

Configuring LDP Signaling for the VPLS Routing Instance

To configure the VPLS routing instance to use LDP signaling, you must configure the same VPLS identifier on each PE router participating in the instance. Specify the VPLS identifier with the vpls-id statement:

vpls-id *vpls-id*;

You can include this statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols vpls]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls]

To configure the VPLS routing instance to use LDP signaling, you also must use the neighbor statement to specify each of the neighboring PE routers that are a part of this VPLS domain:

neighbor *neighbor-id*;

You can include this statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols vpls]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls]

Configuring LDP Signaling on the Router

To enable LDP signaling, you need to configure LDP on each PE router participating in the VPLS routing instance. A minimal configuration is to enable LDP on the loopback interface, which includes the router identifier (router-id), on the PE router using the interface statement:

interface *interface-name*;

You can configure this statement at the following hierarchy levels:

- [edit protocols ldp]
- [edit logical-systems *logical-system-name* protocols ldp]

You can enable LDP on all the interfaces on the router using the all option for the interfaces statement. For more information on how to configure LDP, see the *JUNOS MPLS Applications Configuration Guide*.

Configuring VPLS Routing Instance and VPLS Interface Connectivity

You can configure the VPLS routing instance to take down or maintain its VPLS connections depending on the status of the interfaces configured for the VPLS routing instance. By default, the VPLS connection is taken down whenever a customer-facing interface configured for the VPLS routing instance fails. This behavior can be explicitly configured by specifying the ce option for the connectivity-type statement:

connectivity-type ce;

You can alternatively specify that the VPLS connection remain up so long as an Integrated Routing and Bridging (IRB) interface is configured for the VPLS routing instance by specifying the irb option for the connectivity-type statement:

```
connectivity-type irb;
```

You can configure the connectivity-type statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols vpls]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls]

Configuring the VPLS MAC Table Timeout Interval

You can modify the timeout interval for the VPLS table. We recommend you configure longer values for small, stable VPLS networks and shorter values for large, dynamic VPLS networks. If the VPLS table does not receive any updates during the timeout interval, the router waits one additional interval before automatically clearing the MAC address entries from the VPLS table.

To modify the timeout interval for the VPLS table, include the mac-table-aging-time statement:

```
mac-table-aging-time seconds;
```

You can include this statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols vpls]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls]

NOTE: You cannot configure the mac-table-aging-time statement on MX-series routers.

Configuring the Size of the VPLS MAC Address Table

You can modify the size of the VPLS media access control (MAC) address table. The default table size is 512 MAC addresses, the minimum is 16 addresses, and the maximum is 65,536 addresses.

If the MAC table limit is reached, new MAC addresses can no longer be added to the table. Eventually the oldest MAC addresses are removed from the MAC address table automatically. This frees space in the table, allowing new entries to be added. However, as long as the table is full, new MAC addresses are dropped.

To change the VPLS MAC table size for each VPLS or VPN routing instance, include the mac-table-size statement:

```
mac-table-size size;
```

You can include the mac-table-size statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols vpls]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls]

The interfaces affected by the mac-table-size statement include all of the interfaces within the VPLS routing instance, including the local interfaces, the LSI interfaces, and the VT interfaces.

Limiting the Number of MAC Addresses Learned from an Interface

You can configure a limit on the number of MAC addresses learned by a VPLS routing instance using the mac-table-size statement. If the MAC table limit is reached, new MAC addresses can no longer be added to the table. Eventually the oldest MAC addresses are removed from the MAC address table automatically. This frees space in the table, allowing new entries to be added. However, as long as the table is full, new MAC addresses are dropped.

Because this limit applies to each VPLS routing instance, the MAC addresses of a single interface can consume all the available space in the table, preventing the routing instance from acquiring addresses from other interfaces.

You can limit the number of MAC addresses learned from each interface configured for a VPLS routing instance. To do so, include the interface-mac-limit statement:

```
interface-mac-limit limit;
```

You can include this statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols vpls]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls]

The interface-mac-limit statement affects the local interfaces only (the interfaces facing CE devices).

Configuring the interface-mac-limit statement at the [edit routing-instances *routing-instance-name* protocols vpls] hierarchy level causes the same limit to be applied to all of the interfaces configured for that specific routing instance.

You can also limit the number of MAC addresses learned by a specific interface configured for a VPLS routing instance. This gives you the ability to limit particular interfaces that you expect might generate a lot of MAC addresses.

To limit the number of MAC addresses learned by a specific interface, include the interface-mac-limit statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols vpls site *site-name* interfaces *interface-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls site *site-name* interfaces *interface-name*]

The MAC limit configured for an individual interface at this hierarchy level overrides any value configured at the [edit routing-instances *routing-instance-name* protocols vpls] hierarchy level. Also, the MAC limit configured using the mac-table-size statement can override the limit configured using the interface-mac-limit statement.

The MAC address limit applies to customer-facing interfaces only.

Removing Addresses from the MAC Address Database

You can remove the MAC addresses from the MAC address database that have been learned dynamically. By removing these MAC addresses, MAC address convergence requires less time. To remove MAC addresses from the database, an address withdraw message can be sent with a list of all of the MAC list TLVs to be removed.

To enable MAC TLV receive processing, include the mac-tlv-receive statement:

```
mac-tlv-receive;
```

For a list of the hierarchy levels at which you can include this statement, see the summary section for this statement. You can clear the MAC addresses globally by including the mac-tlv-receive statement at the [edit routing-instances *routing-instance-name* protocols vpls] hierarchy level. To clear MAC address on the routers in a specific mesh group, include the mac-tlv-receive statements at the [edit routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name*] hierarchy level.

To enable MAC TLV send processing, include the mac-tlv-send statement:

```
mac-tlv-send;
```

For a list of the hierarchy levels at which you can include this statement, see the summary section for this statement. You can clear the MAC addresses globally by including the mac-tlv-send statement at the [edit routing-instances *routing-instance-name* protocols vpls] hierarchy level. To clear MAC address on the routers in a specific mesh group, include the mac-tlv-send statements at the [edit routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name*] hierarchy level.

Configuring EXP-Based Traffic Classification for VPLS

You can enable EXP classification on traffic entering core facing VPLS LSI interfaces on a VPLS routing instance by configuring either the vrf-table-label statement or the no-tunnel-services statement. By configuring either of these statements, a default EXP classifier is enabled on every core facing interface that includes family mpls in its configuration. This feature works on MX-series routers only. You can configure an EXP classifier explicitly at the [edit class-of-service] hierarchy level. For more information about EXP classifiers, see the *JUNOS Class of Service Configuration Guide*.

To enable EXP classification on traffic entering core facing VPLS LSI interfaces on a VPLS routing instance, include the vrf-table-label statement:

```
vrf-table-label;
```

You can include this statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*]

You can also enable EXP classification on traffic entering core facing VPLS LSI interfaces on a VPLS routing instance by including the no-tunnel-services statement:

```
no-tunnel-services;
```

You can include this statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols vpls]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls]

Configuring Interfaces for VPLS Routing

On each PE router and for each VPLS routing instance, specify which interfaces are intended for the VPLS traffic traveling between PE and CE routers. To specify the interface for VPLS traffic, include the interface statement in the routing instance configuration:

```
interface interface-name;
```

You can include this statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*]

You must also define each interface by including the following statements:

```
vlan-tagging;
encapsulation encapsulation-type;
unit logical-unit-number {
   vlan-id vlan-id-number;
   family vpls (Interfaces);
}
```

You can include these statements at the following hierarchy levels:

- [edit interfaces *interface-name*]
- [edit logical-systems *logical-system-name* interfaces *interface-name*]

The following sections provide enough information to enable you to configure interfaces for VPLS routing. For detailed information on configuring interfaces and the statements at the [edit interfaces] hierarchy level, see the *JUNOS Network Interfaces Configuration Guide*.

To configure an interface for VPLS, you perform the steps in the following sections:

- Configuring the Interface Name on page 406
- Configuring the VPLS Interface Encapsulation on page 406
- Enabling VLAN Tagging on page 408
- Configuring Aggregated Ethernet Interfaces for VPLS on page 409

Configuring the Interface Name

Specify both the physical and logical portions of the interface name, in the following format:

physical.*logical*

For example, in ge-1/2/1.2, ge-1/2/1 is the physical portion of the interface name and 2 is the logical portion. If you do not specify the logical portion of the interface name, 0 is set by default.

A logical interface can be associated with only one routing instance.

If you enable a routing protocol on all instances by specifying interfaces all when configuring the master instance of the protocol at the [edit protocols] hierarchy level, and you configure a specific interface for VPLS routing at the [edit routing-instances *routing-instance-name*] hierarchy level, the latter interface statement takes precedence and the interface is used exclusively for VPLS.

If you explicitly configure the same interface name at both the [edit protocols] and [edit routing-instances *routing-instance-name*] hierarchy levels and then attempt to commit the configuration, the commit operation will fail.

Configuring the VPLS Interface Encapsulation

You need to specify an encapsulation type for each PE-router-to-CE-router interface configured for VPLS. This section describes the encapsulation statement configuration options available for VPLS. For a full description of all of the options available for this statement, see the *JUNOS Network Interfaces Configuration Guide*.

To configure the encapsulation type on the physical interface, include the encapsulation statement:

```
encapsulation (ethernet-vpls | extended-vlan-vpls | vlan-vpls);
```

You can include the encapsulation statement for physical interfaces at the following hierarchy levels:

- [edit interfaces *interface-name*]
- [edit logical-systems *logical-system-name* interfaces *interface-name*]

You can configure the following physical interface encapsulations for VPLS routing instances:

- ethernet-vpls—Use Ethernet VPLS encapsulation on Ethernet interfaces that have VPLS enabled and that must accept packets carrying standard Tag Protocol ID (TPID) values. On M-series routers (except the M320), the 4-port Fast Ethernet TX PIC and the 1-port, 2-port, and 4-port, 4-slot Gigabit Ethernet PICs can use the Ethernet VPLS encapsulation type.
- extended-vlan-vpls—Use extended virtual LAN (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. On M-series routers (except the M320), the 4-port Fast Ethernet TX PIC and the 1-port, 2-port, and 4-port, 4-slot Gigabit Ethernet PICs can use the Ethernet VPLS encapsulation type.

NOTE: The built-in Gigabit Ethernet PIC on an M7i router does not support extended VLAN VPLS encapsulation.

- vlan-vpls—Use VLAN VPLS encapsulation on Ethernet interfaces with VLAN tagging and VPLS enabled. Interfaces with VLAN VPLS encapsulation accept packets carrying standard TPID values only. On M-series routers (except the M320), the 4-port Fast Ethernet TX PIC and the 1-port, 2-port, and 4-port, 4-slot Gigabit Ethernet PICs can use the Ethernet VPLS encapsulation type.

To configure the encapsulation type for logical interfaces, include the encapsulation statement:

```
encapsulation (ether-vpls-over-atm-llc | vlan-vpls);
```

You can include the encapsulation statement for logical interfaces at the following hierarchy levels:

- [edit interfaces *interface-name* unit *number*]
- [edit logical-systems logical-system-name interfaces *interface-name* unit *number*]

You can configure the following logical interface encapsulations for VPLS routing instances:

- ether-vpls-over-atm-llc—Use Ethernet VPLS over Asynchronous Transfer Mode (ATM) logical link control (LLC) encapsulation to bridge Ethernet interfaces and ATM interfaces over a VPLS routing instance (as described in RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*). Packets from the ATM interfaces are converted to standard ENET2/802.3-encapsulated Ethernet frames with the frame check sequence (FCS) field removed. This encapsulation type is supported on ATM intelligent queuing (IQ) interfaces only.
- vlan-vpls—Use VLAN VPLS encapsulation on Ethernet interfaces with VLAN tagging and VPLS enabled. Interfaces with VLAN VPLS encapsulation accept packets carrying standard TPID values only. On M-series routers (except the M320), the 4-port Fast Ethernet TX PIC and the 1-port, 2-port, and 4-port, 4-slot Gigabit Ethernet PICs can use the Ethernet VPLS encapsulation type.

When you configure the physical interface encapsulation as vlan-vpls, you also need to configure the same interface encapsulation for the logical interface. You need to configure the vlan-vpls encapsulation on the logical interface because the vlan-vpls encapsulation allows you to configure a mixed mode, where some of the logical interfaces use regular Ethernet encapsulation (the default for logical interfaces) and some use vlan-vpls. For more information, see the *JUNOS Network Interfaces Configuration Guide*.

Enabling VLAN Tagging

The JUNOS software supports receiving and forwarding routed Ethernet frames with 802.1Q virtual local area network (VLAN) tags and running Virtual Router Redundancy Protocol (VRRP) over 802.1Q-tagged interfaces. For VPLS to function properly, configure the router to receive and forward frames with 802.1Q VLAN tags by including the vlan-tagging statement:

```
vlan-tagging;
```

You can include the vlan-tagging statement at the [edit interfaces *interface-name*] hierarchy level.

Gigabit Ethernet interfaces can be partitioned; you can assign up to 4095 different logical interfaces, one for each VLAN, but you are limited to a maximum of 1024 VLANs on any single Gigabit Ethernet or 10-Gigabit Ethernet port. Fast Ethernet interfaces can also be partitioned, with a maximum of 1024 logical interfaces for the 4-port FE PIC and 16 logical interfaces for the M40e Internet router. Table 12 on page 408 lists VLAN ID range by interface type.

Table 12: VLAN ID Range by Interface Type

Interface Type	VLAN ID Range
Fast Ethernet	512 through 1023
Gigabit Ethernet	512 through 4094

To bind a VLAN ID to a logical interface, include the vlan-id statement:

```
vlan-id number;
```

You can include this statement at the following hierarchy levels:

- [edit interfaces *interface-name* unit *logical-unit-number*]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]

For more information on how to configure VLANs, see the *JUNOS Network Interfaces Configuration Guide*. For detailed information about how VLAN identifiers in a VPLS routing instance are processed and translated, see the *MX-series Layer 2 Configuration Guide*.

Configuring Aggregated Ethernet 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.

For more information on how aggregated Ethernet interfaces function in the context of VPLS, see "VPLS and Aggregated Ethernet Interfaces" on page 386.

To configure aggregated Ethernet interfaces for VPLS, configure the interface for the VPLS routing instance as follows:

```
interfaces aex {
    vlan-tagging;
    encapsulation encapsulation-type;
    unit logical-unit-number {
        vlan-id number;
    }
}
```

The encapsulation type can be ethernet-vpls, vlan-vpls or extended-vlan-vpls. For the interface configuration statement, in ae*x*, the *x* represents the interface instance number to complete the link association; *x* can be from 0 through 127, for a total of 128 aggregated interfaces.

For more information on how to configure aggregated Ethernet interfaces, see the *JUNOS Network Interfaces Configuration Guide*.

The aggregated Ethernet interface must also be configured for the VPLS routing instance as shown in the following example:

```
[edit]
routing-instances {
    green {
        instance-type vpls;
        interface ae0.0;
        route-distinguisher 10.255.234.34:1;
        vrf-target target:11111:1;
        protocols {
            vpls {
                site-range 10;
                site green3 {
                    site-identifier 3;
                }
            }
        }
    }
}
```

Interface ae0.0 represents the aggregated Ethernet interface in the routing instance configuration. The VPLS routing instance configuration is otherwise standard.

Configuring VPLS Load Balancing

By default, when there are multiple equal-cost paths to the same destination for the active route, the JUNOS software uses a hash algorithm to select one of the next-hop addresses to install in the forwarding table. Whenever the set of next hops for a destination changes, the next-hop address is reselected using the hash algorithm.

You can configure the JUNOS software so that, for the active route, all next-hop addresses for a destination are installed in the forwarding table. This feature is called per-packet load balancing. You can use load balancing to spread traffic across multiple paths between routers. You can also configure per-packet load balancing to optimize VPLS traffic flows across multiple paths.

You can load-balance VPLS traffic based on Layer 2 media access control (MAC) information, IP information and MPLS labels, or MPLS labels only.

NOTE: This feature is not supported on J-series Services Routers or MX-series routers. VPLS Load Balancing based on IP information and MPLS labels is supported only on the M120 and M320 routers.

To optimize VPLS traffic flows across multiple paths, include the family multiservice statement at the [edit forwarding-options hash-key] hierarchy level:

```
family multiservice {
    destination-mac;
    label-1;
    label-2;
    payload {
        ip {
            layer-3-only;
        }
    }
    source-mac;
}
```

To load-balance based on Layer 2 information, include the following configuration options:

- destination-mac—Include the destination MAC address in the hash key used to load-balance the VPLS traffic.
- source-mac—Include the source MAC address in the hash key used to load-balance the VPLS traffic.

You can include the source MAC address in the hash key, the destination MAC address, or both.

For IPv4 traffic, only the IP source and destination addresses are included in the hash key. For MPLS and IPv4 traffic, one or two MPLS labels and IPv4 source and destination addresses are included. For MPLS Ethernet pseudowires, only one or two MPLS labels are included in the hash key. Optionally, you can include only Layer 3 information the IPv4 payload in the hash key.

To load-balance based on IP information and MPLS lablels, include the following configuration options:

- label-1—Include the first MPLS label in the hash key used to load-balance VPLS traffic.
- label-2—Include the second MPLS label in the hash key used to load-balance VPLS traffic.
- payload—Include bits from the IP payload in the hash key used to load-balance VPLS traffic.
- ip—Include the IP address of the IPv4 payload in the hash key used to load-balance VPLS traffic.
- layer-3-only—Include only Layer 3 information in the hash key used to load-balance VPLS traffic

For more information about how to configure per-packet load balancing, see the *JUNOS Policy Framework Configuration Guide*.

Configuring VPLS Without a Tunnel Services PIC

VPLS normally uses a dynamic virtual tunnel logical interface on a Tunnel Services PIC to model traffic from a remote site (a site on a remote PE router that is in a VPLS domain). All traffic coming from a remote site is treated as coming in over the virtual port representing this remote site, for the purposes of Ethernet flooding, forwarding, and learning. An MPLS lookup based on the inner VPN label is done on a PE router. The label is stripped and the Layer 2 Ethernet frame contained within is forwarded to a Tunnel Services PIC. The PIC loops back the packet and then a lookup based on Ethernet MAC addresses is completed. This approach requires that the router have a Tunnel Services PIC and that the PE router completes two protocol lookups.

You can configure VPLS without a Tunnel Services PIC. To do so, you use a label-switched interface (LSI) to provide VPLS functionality. An LSI MPLS label is used as the inner label for VPLS. This label maps to a VPLS routing instance. On the PE router, the LSI label is stripped and then mapped to a logical LSI interface. The Layer 2 Ethernet frame is then forwarded using the LSI interface to the correct VPLS routing instance.

By default, VPLS requires a Tunnel Services PIC. To configure VPLS on a router without a Tunnel Services PIC, include the no-tunnel-services statement:

```
no-tunnel-services;
```

You can configure this statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols vpls]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls]

When you configure VPLS without a Tunnel Services PIC by including the no-tunnel-services statement, the following limitations apply:

- An Enhanced FPC is required.
- Aggregated SONET/SDH interfaces used as core-facing interfaces are not supported.
- Channelized interfaces used as core-facing interfaces are not supported.
- ATM1 interfaces are not supported.

Configuring an Ethernet Switch as the CE Device

For VPLS configurations, the CE device does not necessarily need to be a router. You can link the PE routers directly to Ethernet switches. However, there are a few configuration issues to be aware of:

- When you configure VPLS routing instances and establish two or more connections between a CE Ethernet switch and a PE router, you must enable the Spanning Tree Protocol (STP) on the switch to prevent loops.
- The JUNOS software allows standard Bridge Protocol Data Unit (BPDU) frames to pass through emulated Layer 2 connections, such as those configured with Layer 2 VPNs, Layer 2 circuits, and VPLS instances. However, CE Ethernet switches that generate proprietary BPDU frames might not be able to run STP across Juniper Networks routing platforms configured for these emulated Layer 2 connections.

Mapping VPLS Traffic to a Specific LSP

You can map VPLS traffic to specific LSPs by configuring forwarding table policies. This procedure is optional but can be useful. The following example illustrates how you can map lower priority VPLS routing instances to slower LSPs while mapping other higher priority VPLS routing instances to faster LSPs. In this example configuration, a-to-b1 and a-to-c1 are high-priority LSPs between the PE routers, while a-to-b2 and a-to-c2 are low-priority LSPs between the PE routers.

To map VPLS traffic, include the policy-statement vpls-priority statement:

```
policy-statement vpls-priority {
    term a {
        from {
            rib mpls.0;
            community company-1;
        }
        then {
            install-nexthop lsp [ a-to-b1 a-to-c1 ];
            accept;
        }
    }
    term b {
        from {
            rib mpls.0;
            community company-2;
        }
        then {
```

```
            install-nexthop lsp-regex [ "^a-to-b2$" "^a-to-c2$" ];
            accept;
        }
    }
}
community company-1 members target:11111:1;
community company-2 members target:11111:2;
```

You can include the policy-statement vpls-priority statement at the following hierarchy levels:

- [edit policy-options]
- [edit logical-systems *logical-system-name* policy-options]

Include the export statement to apply the vpls-priority policy to the forwarding table:

```
export vpls-priority;
```

You can include this statement at the following hierarchy levels:

- [edit routing-options forwarding-table]
- [edit logical-systems *logical-system-name* routing-options forwarding-table]

For more information on how to configure routing policies, see the *JUNOS Policy Framework Configuration Guide*.

Configuring VPLS Filters and Policers

You can configure both firewall filters and policers for VPLS. Firewall filters allow you to filter packets based on their components and to perform an action on packets that match the filter. Policers allow you to limit the amount of traffic that passes into or out of an interface.

You can apply VPLS filters and policers on the PE router to customer-facing interfaces only.

The following sections explain how to configure filters and policers for VPLS:

- Configuring a VPLS Filter on page 413
- Configuring a VPLS Policer on page 418

Configuring a VPLS Filter

To configure a filter for VPLS, include the filter statement at the [edit firewall family vpls] hierarchy level:

```
[edit firewall family vpls]
filter filter-name {
    interface-specific;
    term term-name {
        from {
            match-conditions;
```

```
        }
        then {
            actions;
        }
    }
}
```

For more information on how to configure firewall filters, see the *JUNOS Policy Framework Configuration Guide*.

To configure a filter for VPLS traffic, you complete the following tasks:

Configuring an Interface-Specific Counter for VPLS

When you configure a firewall filter for VPLS and apply it to multiple interfaces, you can specify individual counters specific to each interface. This allows you to collect separate statistics on the traffic transiting each interface.

To generate an interface-specific counter for VPLS, you configure the interface-specific statement. A separate instantiation of the filter is generated. This filter instance has a different name (based on the interface name) and collects statistics on the interface specified only.

To configure interface-specific counters, include the interface-specific statement at the [edit firewall family vpls filter *filter-name*] hierarchy level:

```
[edit firewall family vpls filter filter-name]
interface-specific;
```

NOTE: The counter name is restricted to 24 bytes. If the renamed counter exceeds this maximum length, it might be rejected.

For more information on the interface-specific statement and an example of how to configure it, see the *JUNOS Policy Framework Configuration Guide*.

Configuring the VPLS Filter Match Conditions

In the from statement in the VPLS filter term, you specify conditions that the packet must match for the action in the then statement to be taken. All conditions in the

from statement must match for the action to be taken. The order in which you specify match conditions is not important, because a packet must match all the conditions in a term for a match to occur.

If you specify no match conditions in a term, that term matches all packets.

An individual condition in a from statement can contain a list of values. For example, you can specify numeric ranges or multiple source or destination addresses. When a condition defines a list of values, a match occurs if one of the values in the list matches the packet.

Individual conditions in a from statement can be negated. When you negate a condition, you are defining an explicit mismatch. For example, the negated match condition for forwarding-class is forwarding-class-except. If a packet matches a negated condition, it is immediately considered not to match the from statement, and the next term in the filter is evaluated, if there is one; if there are no more terms, the packet is discarded.

To specify the match conditions for a VPLS filter term, include the from statement at the [edit firewall family vpls filter *filter-name* term *term-name*] hierarchy level. Table 13 on page 415 describes the match conditions available for VPLS filters.

```
[edit firewall family vpls filter filter-name term term-name]
from match-conditions;
```

Table 13: VPLS Filter Match Conditions

Match Condition	Description
destination-mac-address *mac-address*	Specified destination MAC address.
ether-type *value*	Ethernet packets. Configure the ether-type match condition when the encapsulation of the associated interfaces is ethernet-vpls.
forwarding-class *value*	Specified forwarding class.
interface-group *index*	Interface group on which the packet was received. An interface group is a set of one or more logical interfaces.
source-mac-address *mac-address*	Source MAC address.
vlan-ether-type *value*	VLAN Ethernet packets. Configure the vlan-ether-type match condition when the encapsulation of the associated interfaces is either vlan-vpls or extended-vlan-vpls.

Configuring an Action for the VPLS Filter

You can configure the following actions for a VPLS filter at the [edit firewall family vpls filter *filter-name* term *term-name* then] hierarchy level: accept, count, discard, forwarding-class, loss-priority, next, policer.

Configuring VPLS FTFs

Forwarding table filters (FTFs) are filters configured for forwarding tables. For VPLS, they are attached to the destination MAC (DMAC) forwarding table of the VPLS routing instance. You define VPLS FTFs in the same manner as any other type of FTF. You can only apply a VPLS FTF as an input filter.

To specify a VPLS FTF, include the filter input statement at the [edit routing-instance *routing-instance-name* forwarding-options family vpls] hierarchy level:

```
[edit routing-instance routing-instance-name forwarding-options family vpls]
filter input filter-name;
```

For the statement summaries of these statements, see the *JUNOS Policy Framework Configuration Guide*.

Changing Precedence for Spanning Tree BPDU Packets

Spanning tree BPDU packets are automatically set to a high precedence. The queue number on these packets is set to 3. On M-series routers (except the M320) by default, a queue value of 3 indicates high precedence. To enable this higher precedence on BPDU packets, an instance-specific BPDU precedence filter named default_bpdu_filter is automatically attached to the VPLS DMAC table. This filter places a high precedence on all packets sent to 01:80:c2:00:00:00/24.

You can overwrite this filter by configuring a VPLS FTF filter and applying it to the VPLS routing instance. For more information, see “Configuring VPLS FTFs” on page 416 and “Applying a VPLS Filter to a VPLS Routing Instance” on page 417.

Applying a VPLS Filter to an Interface

To apply a VPLS filter to an interface, include the filter statement:

```
filter {
    input input-filter-name;
    output output-filter-name;
    group index;
}
```

You can include the filter statement at the following hierarchy levels:

- [edit interfaces *interface-name* unit *number* family vpls]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *number* family vpls]

In the input statement, list the name of the VPLS filter to be evaluated when packets are received on the interface. In the output statement, list the name of the VPLS filter to be evaluated when packets are transmitted on the interface.

NOTE: For output interface filters, MAC addresses are learned after the filter action is completed. When an output interface filter's action is discard, the packet is dropped before the MAC address is learned. However, an input interface filter learns the MAC address before discarding the packet.

For the statement summaries for these statements, see the *JUNOS Network Interfaces Configuration Guide*.

Applying a VPLS Filter to a VPLS Routing Instance

You can apply a VPLS filter to a VPLS routing instance. The filter checks traffic passing through the specified routing instance.

Input routing instance filters learn the MAC address before the filter action is completed, so if the filter action is discard, the MAC address is learned before the packet is dropped.

To apply a VPLS filter to packets arriving at a VPLS routing instance and specify the filter, include the filter input statement at the [edit routing-instances *routing-instance-name* forwarding-options family vpls] hierarchy level:

```
[edit routing-instances routing-instance-name forwarding-options family vpls]
filter input input-filter-name;
```

Configuring a Filter for Flooded Traffic

You can configure a VPLS filter to filter flooded packets. CE routers typically flood the following types of packets to PE routers in VPLS routing instances:

- Layer 2 broadcast packets
- Layer 2 multicast packets
- Layer 2 unicast packets with an unknown destination MAC address
- Layer 2 packets with a MAC entry in the DMAC routing table

You can configure filters to manage how these flooded packets are distributed to the other PE routers in the VPLS routing instance.

To apply a flooding filter to packets arriving at the PE router in the VPLS routing instance, and specify the filter, include the flood input statement:

```
flood input filter-name;
```

You can include this statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* forwarding-options family vpls]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* forwarding-options family vpls]

Configuring a VPLS Policer

You can configure a policer for VPLS traffic. The VPLS policer configuration is similar to the configuration of any other type of policer.

VPLS policers have the following characteristics:

- You cannot police the default VPLS routes stored in the flood table from PE router–sourced flood traffic.
- When specifying policing bandwidth, the VPLS policer considers all Layer 2 bytes in a packet to determine the packet length.

To configure a VPLS policer, include the policer statement at the [edit firewall] hierarchy level:

```
[edit firewall]
policer policer-name {
    bandwidth-limit limit;
    burst-size-limit limit;
    then action;
}
```

For the statement summaries of these statements and more information on how to configure policers, see the *JUNOS Policy Framework Configuration Guide*.

To apply a VPLS policer to an interface, include the policer statement:

```
policer {
    input input-policer-name;
    output output-policer-name;
}
```

You can include the policer statement at the following hierarchy levels:

- [edit interfaces *interface-name* unit *number* family vpls]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *number* family vpls

In the input statement, list the name of the VPLS policer to be evaluated when packets are received on the interface. In the output statement, list the name of the VPLS policer to be evaluated when packets are transmitted on the interface. This type of VPLS policer can only apply to unicast packets. For information on how to filter flood packets, see "Configuring a Filter for Flooded Traffic" on page 417.

For the statement summaries for these statements, see the *JUNOS Network Interfaces Configuration Guide*.

Specifying the VT Interfaces Used by VPLS Routing Instances

By default, the JUNOS software automatically selects one of the virtual tunnel (VT) interfaces available to the router for de-encapsulating traffic from a remote site. The JUNOS software cycles through the currently available VT interfaces, regularly updating

the list of available VT interfaces as new remote sites are discovered and new connections are brought up. However, you can also explicitly configure which VT interfaces will receive the VPLS traffic.

By configuring the tunnel-services statement at the [edit routing-instances *routing-instance-name* protocols vpls] hierarchy level, you can specify that traffic for particular VPLS routing instances be forwarded to specific VT interfaces. Doing so allows you to load-balance VPLS traffic among all the available VT interfaces on the router.

The tunnel-services statement includes the following options:

- devices—Specifies the VT interfaces acceptable for use by the VPLS routing instance. If you do not configure this option, all VT interfaces available to the router can be used for de-encapsulating traffic for this instance.
- primary—Specifies the primary VT interface to be used by the VPLS routing instance. The VT interface specified is used to de-encapsulate all VPLS traffic from the MPLS core network for this routing instance. If the VT interface specified is unavailable, then one of the other acceptable VT interfaces (specified in the devices option) is used for handling the VPLS traffic. If you do not configure this option, any acceptable VT interface can be used to de-encapsulate VPLS traffic from the core.

To specify that traffic for a particular VPLS routing instance be forwarded to specific VT interfaces, include the tunnel-services statement:

```
tunnel-services {
    devices device-names;
    primary primary-device-name;
}
```

These statements can be configured at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols vpls]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls]

Configuring VPLS Multihoming

VPLS multihoming allows you to connect a customer site to multiple PE routers to provide redundant connectivity while preventing the formation of Layer 2 loops in the service provider's network. A VPLS site multihomed to two or more PE routers provides redundant connectivity in the event of a PE router-to-CE device link failure or the failure of a PE router. For more information on VPLS multihoming, see "VPLS Multihoming" on page 387.

NOTE: If you want to enable multihoming for a VPLS routing instance, you cannot also enable LDP signaling. You can only enable BGP signaling.

The following sections describe how to configure VPLS multihoming. Some information is also provided on single-homed site configuration versus multihomed site configuration.

VPLS Multihomed Site Configuration

The following describes the requirements for a VPLS multihomed site configuration:

- Assign the same site ID on all PE routers connected to the same CE devices.
- Assign the same route distinguisher on all PE routers connected to the same CE devices.
- Reference all interfaces assigned to the multihomed VPLS site on each PE router. Only one of these interfaces is used to send and receive traffic for this site at a time.
- Either designate a primary interface or allow the router to select the interface to be used as the primary interface.

  If the router selects the interface, the interface used to connect the PE router to the site depends on the order in which interfaces are listed in the PE router's configuration. The first operational interface in the set of configured interfaces is chosen to be the designated interface. If this interface fails, the next interface in the list is selected to send and receive traffic for the site.
- Configure multihoming for the site.

The following configuration shows the statements you need to configure to enable VPLS multihoming:

```
[edit routing-instances routing-instance-name]
instance-type vpls;
interface interface-name;
interface interface-name;
protocols vpls {
   site site-name {
      active-interface {
         any;
         primary interface-name;
      }
      interface interface-name;
      interface interface-name;
      multi-homing;
      site-identifier number;
   }
}
route-distinguisher (as-number:id | ip-address:id);
```

NOTE: If you add a direct connection between CE devices that are multihomed to the same VPLS site on different PE routers, the traffic can loop and a loss of connectivity might occur. We do not recommend this topology.

Most of these statements are explained in more detail in the rest of this chapter. The following sections explain how to configure the statements that are specific to VPLS multihoming:

- Specifying an Interface as the Active Interface on page 421
- Configuring Multihoming on the PE Router on page 421

Specifying an Interface as the Active Interface

You need to specify one of the interfaces for the multihomed site as the primary interface. If there are multiple interfaces, the remaining interfaces are activated only when the primary interface goes down. If no active interfaces are configured at the site level, all traffic for a VPLS site travels through a single, non-multihomed PE router.

You must configure one of the following options for the active-interface statement:

- any—One configured interface is randomly designated as the active interface for the VPLS site.
- primary—Specify the name of the multihomed interface to be used as the primary interface by the VPLS site.

To specify a multihomed interface as the primary interface for the VPLS site, include the active-interface statement:

```
active-interface {
    any;
    primary interface-name;
}
```

You can configure this statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols vpls site *site-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls site *site-name*]

Configuring Multihoming on the PE Router

When a CE device is connected to the same VPLS site on more than one PE router, include the multi-homing statement on all associated PE routers. Configuration of this statement tracks BGP peers. If no BGP peer is available, VPLS deactivates all active interfaces for a site. To specify that the PE router is part of a multihomed VPLS site, include the multi-homing statement:

```
multi-homing;
```

You can configure this statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols vpls site *site-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls site *site-name*]

Include the multi-homing statement on all PE routers associated with a particular VPLS site.

VPLS Single-Homed Site Configuration

All VPLS single-homed sites are connected to the same default VE device. All interfaces in a VPLS routing instance that are not configured as part of a multihomed site are assumed to be single-homed to the default VE device.

Flooding Unknown Traffic Using Point-to-Multipoint LSPs

For a VPLS routing instance, you can flood unknown unicast, broadcast, and multicast traffic using point-to-multipoint (also called *P2MP*) LSPs. By default, VPLS relies upon ingress replication to flood unknown traffic to the members of a VPLS routing instance. This can cause replication of data at routing nodes shared by multiple VPLS members, as shown in Figure 47 on page 422. The flood data is tripled between PE router PE1 and provider router P1 and doubled between provider routers P1 and P2. By configuring point-to-multipoint LSPs to handle flood traffic, the VPLS routing instance can avoid this type of traffic replication in the network, as shown in Figure 48 on page 422.

Figure 47: Flooding Unknown VPLS Traffic Using Ingress Replication

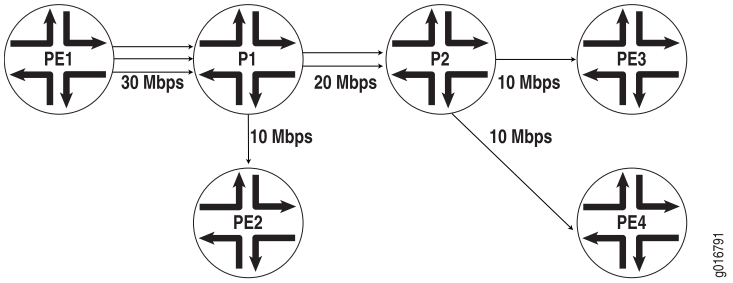
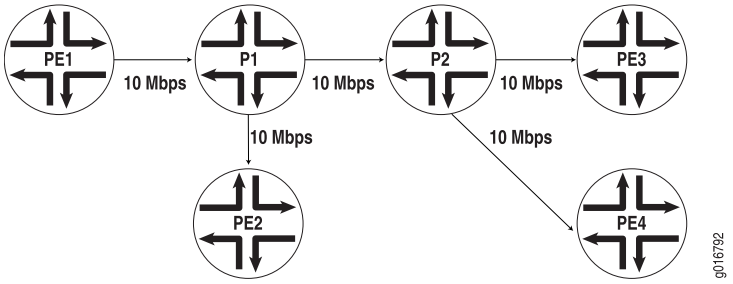


Figure 48: Flooding Unknown VPLS Traffic Using a Point-to-Multipoint LSP



The point-to-multipoint LSP used for VPLS flooding can be either static or dynamic. In either case, for each VPLS routing instance, the PE router creates a dedicated point-to-multipoint LSP. All of the neighbors of the VPLS routing instance are added to the point-to-multipoint LSP when the feature is enabled. If there are *n* PE routers in the VPLS routing instance, *n* point-to-multipoint LSPs are created in the network where each PE router is the root of the point-to-multipoint tree and includes the rest of the *n – 1* PE routers as leaf nodes. If you configured static point-to-multipoint LSPs for flooding, any additional VPLS neighbors added to the routing instance later are not automatically added to the point-to-multipoint LSP. You will need to manually add the new VPLS neighbors to the static point-to-multipoint flooding LSP. If you configure dynamic point-to-multipoint LSPs, whenever VPLS discovers a new neighbor through BGP, a sub-LSP for this neighbor is added to the point-to-multipoint LSP for the routing instance.

This feature can be enabled incrementally on any PE router that is part of a specific VPLS routing instance. The PE routers can then use point-to-multipoint LSPs to flood traffic, whereas other PE routers in the same VPLS routing instance can still use ingress replication to flood traffic. However, when this feature is enabled on any PE router, you must ensure that all PE routers in the VPLS routing instance that participate in the flooding of traffic over point-to-multipoint LSPs are upgraded to JUNOS Release 8.3 or later to support this feature.

To flood unknown unicast, broadcast, and multicast traffic using point-to-multipoint LSPs, configure the rsvp-te statement as follows:

```
rsvp-te {
    label-switched-path-template (default-template | p2mp-lsp-template-name);
    static-lsp lsp-name;
}
```

You can configure this statement at the following hierarchy levels:

- [edit routing-instance *routing-instance-name* provider-tunnel]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel]

You can configure either a static point-to-multipoint LSP for VPLS flooding or a dynamic point-to-multipoint LSP.

NOTE: You cannot specify both the static and label-switched-path-template statements at the same time.

The following sections describe how to configure static and dynamic point-to-multipoint LSPs for flooding unknown traffic in a VPLS routing instance:

- Configuring Static Point-to-Multipoint Flooding LSPs on page 424
- Configuring Dynamic Point-to-Multipoint Flooding LSPs on page 424

Configuring Static Point-to-Multipoint Flooding LSPs

The static-lsp option creates a static flooding point-to-multipoint LSP that includes all of the neighbors in the VPLS routing instance. Flood traffic is sent to all of the VPLS neighbors using the generated point-to-multipoint LSP. VPLS neighbors added to the routing instance later are not automatically added to the point-to-multipoint LSP. You will need to manually add the new VPLS neighbors to the static point-to-multipoint flooding LSP. By configuring static point-to-multipoint LSPs for flooding, you have more control over which path each sub-LSP follows.

To configure a static flooding point-to-multipoint LSP, specify the name of the static flooding point-to-multipoint LSP by including the static-lsp statement:

```
static-lsp lsp-name;
```

You can configure this statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* provider-tunnel rsvp-te]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel rsvp-te]

Configuring Dynamic Point-to-Multipoint Flooding LSPs

To configure a dynamic point-to-multipoint flooding LSP, include the label-switched-path-template statement option at the [edit routing-instances *routing-instance-name* provider-tunnel rsvp-te] hierarchy level:

```
[edit routing-instances routing-instance-name provider-tunnel rsvp-te]
label-switched-path-template (default-template | lsp-template-name);
```

You can automatically generate the point-to-multipoint LSP to be used for flooding unknown traffic or you can manually configure the point-to-multipoint LSP:

- Configuring Dynamic Point-to-Multipoint Flooding LSPs with the Default Template on page 424
- Configuring Dynamic Point-to-Multipoint Flooding LSPs with a Preconfigured Template on page 425

Configuring Dynamic Point-to-Multipoint Flooding LSPs with the Default Template

The default-template option, specified at the [edit routing-instances *routing-instance-name* provider-tunnel rsvp-te label-switched-path-template] hierarchy level causes the point-to-multipoint LSPs to be created with the default parameters. The default parameters are for a minimally configured point-to-multipoint LSP. The name of this point-to-multipoint LSP is also generated automatically and is based on the following model:

id:vpls:*router-id*:*routing-instance-name*

The following show command output for show mpls lsp p2mp illustrates how a point-to-multipoint flood LSP name could appear if you configure the label-switched-path-template statement with the default-template option:

```
user@host> show mpls lsp p2mp ingress
Ingress LSP: 2 sessions P2MP name: static, P2MP branch count: 3
To              From            State Rt ActivePath       P     LSPname
10.255.14.181   10.255.14.172   Up     0                  *     vpn02-vpn11
10.255.14.177   10.255.14.172   Up     0 path2            *     vpn02-vpn07
10.255.14.174   10.255.14.172   Up     0 path3            *     vpn02-vpn04
P2MP name: 9:vpls:10.255.14.172:green, P2MP branch count: 2
To              From            State Rt ActivePath       P     LSPname
10.255.14.177   10.255.14.172   Up     0                  *
11:vpls:10.255.14.172:green
10.255.14.174   10.255.14.172   Up     0                  *
10:vpls:10.255.14.172:green
Total 5 displayed, Up 5, Down 0
```

The dynamically generated point-to-multipoint LSP name is 9:vpls:10.255.14.172:green.

Configuring Dynamic Point-to-Multipoint Flooding LSPs with a Preconfigured Template

You can configure a point-to-multipoint flooding LSP template for the VPLS routing instance. The template allows you to specify the properties of the dynamic point-to-multipoint LSPs that are used to flood traffic for the VPLS routing instance. You can specify all of the standard options available for a point-to-multipoint LSP within this template. These properties are inherited by the dynamic point-to-multipoint flood LSPs.

To configure a point-to-multipoint LSP template for flooding VPLS traffic, specify all of the properties you want to include in a point-to-multipoint LSP configuration. To specify this LSP as a point-to-multipoint flooding template, include the p2mp and template statements:

```
p2mp;
template;
```

You can configure these statements at the following hierarchy levels:

- [edit protocols mpls label-switched-path *p2mp-lsp-template-name*]
- [edit logical-systems *logical-system-name* protocols mpls label-switched-path *p2mp-lsp-template-name*]

For more information about how to configure the p2mp statement and point-to-multipoint LSPs, see the *JUNOS MPLS Applications Configuration Guide*.

Once you have configured the point-to-multipoint LSP template, specify the name of the point-to-multipoint LSP template with the label-switched-path-template statement:

```
label-switched-path-template p2mp-lsp-template-name;
```

You can configure this statement at the following hierarchy levels:

- [edit routing-instance *routing-instance-name* provider-tunnel rsvp-te]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel rsvp-te]

Configuring VPLS and Integrated Routing and Bridging

Traditional Layer 2 switching environments consist of Layer 2 devices (such as switches) that partition data into broadcast domains. The broadcast domains can be created through physical topologies or logically through virtual local area networks (VLANs). For MX-series routers, you can logically configure broadcast domains within virtual switch routing instances, VPLS routing instances, or bridging domains. The individual routing instances or bridging domains are differentiated through VLAN identifiers and these instances or domains function much like traditional VLANs.

For detailed information and configuration instructions on bridging domains and spanning tree protocol, see the *JUNOS Network Interfaces Configuration Guide*, the *JUNOS Routing Protocols Configuration Guide*, and the *JUNOS Feature Guide*.

The following sections provide configuration information specific to VPLS in regards to integrated routing and bridging:

- Configuring MAC Address Flooding and Learning for VPLS on page 426
- Configuring MSTP for VPLS on page 427

Configuring MAC Address Flooding and Learning for VPLS

In a VPLS routing instance or bridge domain, when a frame is received from a CE interface, it is flooded to the other CE interfaces and all of the VE interfaces if the destination MAC address is not learned or if the frame is either broadcast or multicast. If the destination MAC address is learned on another CE device, such a frame is unicasted to the CE interface on which the MAC address is learned. This might not be desirable if the service provider does not want CE devices to communicate with each other directly.

To prevent CE devices from communicating directly include the no-local-switching statement. If the no-local-switching statement is configured, frames arriving on a CE interface are sent to VE or core-facing interfaces only.

```
no-local-switching;
```

You can include this statement at the [edit bridge-domains *bridge-domain-name*] hierarchy level. The no-local-switching statement is only available on MX-series routing platforms.

Configuring MSTP for VPLS

When you configure integrated routing and bridging, you might also need to configure the Multiple Spanning Tree Protocol (MSTP). When you configure MSTP on a provider edge (PE) router running VPLS, you must also configure ethernet-vpls encapsulation on the customer-facing interfaces. VLAN-based VPLS interface encapsulations are not supported with MSTP.

Configuring Interoperability Between BGP Signaling and LDP Signaling in VPLS

A single VPLS routing instance can encompass one set of PE routers that use BGP for signaling and another set of PE routers that use LDP for signaling. Within each set, all of the PE routers are fully meshed in both the control and data planes and have a bidirectional pseudowire to each of the other routers in the set. However, the BGP-signaled routers cannot be directly connected to the LDP-signaled routers. To be able to manage the two separate sets of PE routers in a single VPLS routing instance, a border PE router must be configured to interconnect the two sets of routers.

The VPLS RFCs and Internet drafts require that all of the PE routers participating in a single VPLS routing instance must be fully meshed in the data plane. In the control plane, each fully meshed set of PE routers in a VPLS routing instance is called a PE router mesh group. The border PE router must be reachable by and have bidirectional pseudowires to all of the PE routers that are a part of the VPLS routing instance, both the LDP-signaled and BGP-signaled routers.

For LDP BGP interworking to function, LDP-signaled routers can only be configured with forwarding equivalence class (FEC) 128.

The following sections describe how to configure BGP LDP interworking for VPLS:

- Configuring VPLS Mesh Groups for LDP BGP Interworking on page 427
- Configuring Switching Between Pseudowires Using VPLS Mesh Groups on page 428
- Configuring Integrated Routing and Bridging Support for LDP BGP Interworking with VPLS on page 428
- Configuring Inter-AS VPLS with MAC Processing at the ASBR on page 429

Configuring VPLS Mesh Groups for LDP BGP Interworking

To configure LDP BGP interworking for VPLS, include the mesh-group statement in the VPLS routing instance configuration of the PE border router:

```
mesh-group mesh-group-name {
    local-switching;
    mac-tlv-receive
    mac-tlv-send
    neighbor address;
    peer-as all;
    vpls-id number;
}
```

You can configure this statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols vpls]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls]

Using the neighbor statement, configure each PE router that is a part of the mesh group. You must separate the LDP-signaled routers and the BGP-signaled routers into their own respective mesh groups. The LDP-signaled routers can be divided into multiple mesh groups. The BGP-signaled routers must be configured within a single mesh group for each routing instance.

Configuring Switching Between Pseudowires Using VPLS Mesh Groups

To configure switching between Layer 2 circuit pseudowires using VPLS mesh groups, you can do either of the following:

- Configure a mesh group for each Layer 2 circuit pseudowire terminating at a VPLS routing instance. You can configure a maximum of 16 mesh groups on MX-series routers and a maximum of 256 mesh groups for M-series and T-series routers.
- Configure a single mesh group, terminate all the Layer 2 circuit pseudowires into it, and enable local switching between the pseudowires by including the local-switching statement at the [edit routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name*] hierarchy level. By default, you cannot configure local switching for mesh groups (except for the CE mesh group) because all of the VPLS PE routers must be configured in a full mesh. However, local switching is useful if you are terminating Layer 2 circuit pseudowires in a mesh group configured for an LDP signaled VPLS routing instance.

NOTE: Do not include the local-switching statement on PE routers configured in a full mesh VPLS network.

To terminate multiple pseudowires at a single VPLS mesh group, include the local-switching statement:

```
local-switching;
```

You can include this statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name*]

Configuring Integrated Routing and Bridging Support for LDP BGP Interworking with VPLS

Beginning with JUNOS Release 9.4, you can configure an integrated routing and bridging (IRB) interface on a router that functions as an autonomous system border

router (ASBR) in an inter-AS VPLS environment between BGP-signaled VPLS and LDP-signaled VPLS. Previously, IRB interfaces were supported only on Provider Edge (PE) routers.

To configure a IRB support for LDP BGP Interworking with VPLS, include the routing-interface *interface-name* statement.

You can include this statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-routers *logical-router-name* routing-instances *routing-instance-name*]

Configuring Inter-AS VPLS with MAC Processing at the ASBR

Inter-AS VPLS with MAC processing at the ASBR enables you to interconnect customer sites that are located in different ASs. In addition, you can configure the ASs with different signaling protocols. You can configure one of the ASs with BGP-signaled VPLS and the other with LDP-signaled VPLS. For more information about how to configure LDP-signaled and BGP signaled VPLS, see "Configuring Interoperability Between BGP Signaling and LDP Signaling in VPLS" on page 427.

For inter-AS VPLS to function properly, you need to configure IBGP peering between the PE routers, including the ASBRs in each AS, just as you do for a typical VPLS configuration. You also need to configure EBGP peering between the ASBRs in the separate ASs. The EBGP peering is needed between the ASBRs only. The link between the ASBR routers does not have to be Ethernet. You can also connect a CE router directly to one of the ASBRs, meaning you do not have to have a PE router between the ASBR and the CE router.

The configuration for the connection between the ASBRs makes inter-AS VPLS with MAC operations unique. The other elements of the configuration are described in other sections of this manual. An extensive configuration example for inter-AS VPLS with MAC operations is provided in the *JUNOS Feature Guide*.

The following sections describe how to configure inter-AS VPLS with MAC operations:

- Inter-AS VPLS with MAC Operations Configuration Summary on page 429
- Configuring the ASBRs for Inter-AS VPLS on page 430

Inter-AS VPLS with MAC Operations Configuration Summary

This section provides a summary of all of the elements which must be configured to enable inter-AS VPLS with MAC operations. These procedures are described in detail later in this chapter and in other parts of the *JUNOS VPNs Configuration Guide*.

The following lists all of major elements of an inter-AS VPLS with MAC operations configuration:

- Configure IBGP between all of the routers within each AS, including the ASBRs.
- Configure EBGP between the ASBRs in the separated ASs. The EBGP configuration includes the configuration that interconnects the ASs.

- Configure a full mesh of LSPs between the ASBRs.
- Configure a VPLS routing instance encompassing the ASBR routers. The ASBRs are VPLS peers and are linked by a single pseudowire. Multihoming between ASs is not supported. A full mesh of pseudowires is needed between the ASBR routers in all of the interconnected ASs.
- Configure the VPLS routing instances using either BGP signaling or LDP signaling. LDP BGP interworking is supported for inter-AS VPLS with MAC operations, so it is possible to interconnect the BGP-signaled VPLS routing instances with the LDP-signaled VPLS routing instances.
- Configure a single VPLS mesh group for all of the ASBRs interconnected using inter-AS VPLS.

Configuring the ASBRs for Inter-AS VPLS

This section describes the configuration on the ASBRs needed to enable inter-AS VPLS with MAC operations.

On each ASBR, you need to configure a VPLS mesh group within the VPLS routing instance which needs to include all of the PE routers within the AS, in addition to the ASBR. You need to configure the same mesh group for each of the ASs you want to interconnect using inter-AS VPLS. The mesh group name should be identical on each AS. You also must configure the peer-as all statement. This statement enables the router to establish a single pseudowire to each of the other ASBRs.

Configure the mesh group on each ASBR by including the mesh-group and peer-as all statements:

```
mesh-group mesh-group-name {
    peer-as all;
}
```

You can include these statements at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols vpls]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls]

Tracing VPLS Traffic and Operations

To trace VPLS traffic, include the traceoptions statement:

```
traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
}
```

You can include the traceoptions statement at the following hierarchy levels:

- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls]
- [edit routing-instances *routing-instance-name* protocols vpls]

The following trace flags display the operations associated with VPLS:

- all—All VPLS tracing options
- connections—VPLS connections (events and state changes)
- error—Error conditions
- nlri—VPLS advertisements received or sent using BGP
- route—Trace-routing information
- topology—VPLS topology changes caused by reconsideration or advertisements received from other PE routers using BGP

Chapter 20

Summary of VPLS Configuration Statements

The following sections explain the major routing-instances and interfaces configuration statements that apply specifically to virtual private LAN service (VPLS). The statements are organized alphabetically. The routing instance statements at the [edit routing-instances *routing-instance-name*] hierarchy level are explained in the *JUNOS Routing Protocols Configuration Guide*. The interface statements at the [edit interfaces *interface-name*] hierarchy level are explained in the *JUNOS Network Interfaces Configuration Guide*.

active-interface

Syntax

```
active-interface {
    any;
    primary interface-name;
}
```

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls site *site-name*],
[edit routing-instances *routing-instance-name* protocols vpls site *site-name*]

Release Information Statement introduced in JUNOS Release 7.5.

Description Specify a multihomed interface as the primary interface for the VPLS site. If there are multiple interfaces, the remaining interfaces are activated only when the primary interface goes down. 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.

Options

any—One configured interface is randomly designated as the active interface for the VPLS site.

primary *interface-name*—Specify the name of the multihomed interface to be used as the primary interface by the VPLS site.

Usage Guidelines See "Specifying an Interface as the Active Interface" on page 421.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

automatic-site-id

Syntax

```
automatic-site-id {
    collision-detect-time seconds;
    new-site-wait-time seconds;
    reclaim-wait-time minimum seconds maximum seconds;
    startup-wait-time seconds;
}
```

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls site *site-name*],
[edit routing-instances *routing-instance-name* protocols vpls site *site-name*]

Release Information Statement introduced in JUNOS Release 9.1.

Description Enables automatic site identifiers for VPLS routing instances.

Options

collision-detect-time—The time in seconds to wait after a claim advertisement is sent to the other routers in a VPLS instance before a PE router can begin using a site identifier. If the PE router receives a competing claim advertisement for the same site identifier during this time period, it initiates the collision resolution procedure for site identifiers.

new-site-wait-time—The time in seconds to wait to receive VPLS information for a newly configured routing instance or a new site. This time interval is also applied whenever the automatic site identifier feature is activated on a VPLS routing instance other than at startup. Effectively, this timer indicates how long to wait before an attempt is made to allocate a site identifier. This timer is also triggered whenever a VPLS routing instance is enabled.

reclaim-wait-time—The time in seconds to wait to receive VPLS information for a newly configured routing instance or a new site. This time interval is also applied whenever the automatic site identifier feature is activated on a VPLS routing instance other than at startup. Effectively, this timer indicates how long to wait before an attempt is made to allocate a site identifier. This timer is also triggered whenever a VPLS routing instance is enabled. You can configure two values for this option: the minimum wait time and the maximum wait time.

startup-wait-time—The time in seconds to wait at startup to receive all the VPLS information for the route targets configured on the other PE routers included in the VPLS routing instance.

Usage Guidelines See “Configuring Automatic Site Identifiers for VPLS” on page 397.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

connectivity-type

Syntax connectivity-type (ce | irb);

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls],
[edit routing-instances *routing-instance-name* protocols vpls]

Release Information Statement introduced in JUNOS Software 9.1. The irb option was introduced in JUNOS Software 9.3.

Description Specifies when a VPLS connection is taken down depending on whether or not the interface for the VPLS routing instance is customer facing or Integrated Routing and Bridging (IRB).

Default ce

Options ce—Requires that for the VPLS connection to be up, the customer facing interface for the VPLS routing instance must also be up. If the customer facing interface fails, the VPLS connection is taken down.

irb—Allows a VPLS connection to remain up so long as an IRB interface is configured for the VPLS routing instance.

Usage Guidelines See "Configuring VPLS Routing Instance and VPLS Interface Connectivity" on page 401.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

encapsulation

Syntax encapsulation (ethernet-vpls | ether-vpls-over-atm-llc | extended-vlan-vpls | vlan-vpls);

Hierarchy Level [edit interfaces *interface-name*],
[edit logical-systems *logical-system-name* interfaces *interface-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Physical link-layer encapsulation type for VPLS interfaces. This statement summary for the encapsulation statement describes encapsulations supported for VPLS only. For a full description of the encapsulation-type statement, see encapsulation-type.

Options

ethernet-vpls—Use Ethernet VPLS encapsulation on Ethernet interfaces that have VPLS enabled and that must accept packets carrying standard Tag Protocol ID (TPID) values. On M-series routing platforms, except the M320, the 4-port Fast Ethernet TX PIC and the 1-port, 2-port, and 4-port, 4-slot Gigabit Ethernet PICs can use the Ethernet VPLS encapsulation type.

ether-vpls-over-atm-llc—For ATM intelligent queuing (IQ) interfaces only, use the Ethernet virtual private LAN service (VPLS) over ATM LLC encapsulation to bridge Ethernet interfaces and ATM interfaces over a VPLS routing instance (as described in RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*). Packets from the ATM interfaces are converted to standard ENET2/802.3 encapsulated Ethernet frames with the frame check sequence (FCS) field removed.

extended-vlan-vpls—Use extended virtual local area network (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. On M-series routing platforms, except the M320, the 4-port Fast Ethernet TX PIC and the 1-port, 2-port, and 4-port, 4-slot Gigabit Ethernet PICs can use the Ethernet VPLS encapsulation type.

vlan-vpls—Use VLAN VPLS encapsulation on Ethernet interfaces with VLAN tagging and VPLS enabled. Interfaces with VLAN VPLS encapsulation accept packets carrying standard TPID values only. On M-series routing platforms, except the M320, the 4-port Fast Ethernet TX PIC and the 1-port, 2-port, and 4-port, 4-slot Gigabit Ethernet PICs can use the Ethernet VPLS encapsulation type.

Usage Guidelines See “Configuring the VPLS Interface Encapsulation” on page 406.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

family multiservice

Syntax

```
family multiservice {
    destination-mac;
    label-1;
    label-2;
    payload {
        ip {
            layer-3-only
        }
    }
    source-mac;
}
```

Hierarchy Level [edit forwarding-options hash-key]

Release Information Statement introduced in JUNOS Release 8.0.
label-1 statement introduced in JUNOS Release 9.4.
label-2 statement introduced in JUNOS Release 9.4.
payload statement introduced in JUNOS Release 9.4.
ip statement introduced in JUNOS Release 9.4.
layer-3-only statement introduced in JUNOS Release 9.4.

Description Configure per-packet load balancing based on the MAC addresses.

Options

destination-mac—Include the destination MAC address in the hash key used to load balance the VPLS traffic.

label-1 (M120 and M320 routers only)—Include the first MPLS label in the hash key used to load balance VPLS traffic.

label-2 (M120 and M320 routers only)—Include the second MPLS label in the hash key used to load balance VPLS traffic.

payload (M120, and M320 routers only)—Include bits from the IP payload in the hash key used to load balance VPLS traffic.

ip (M120, and M320 routers only)—Include the IP address of the IPv4 payload in the hash key used to load balance VPLS traffic.

layer-3-only (M120, and M320 routers only)—Include only the Layer 3 inforamtion from the packet's Ip payload in the hash key used to load balance VPLS traffic.

source-mac—Include the source MAC address in the hash key used to load balance the VPLS traffic.

Usage Guidelines See "Configuring VPLS Load Balancing" on page 410.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

interface

Syntax
```
interface interface-name {
    interface-mac-limit limit;
}
```

Hierarchy Level
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls site *site-name*],
[edit routing-instances *routing-instance-name* protocols vpls site *site-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure the Layer 2 circuit pseudowires for a VPLS site as logical interfaces within the VPLS site configuration.

Options *interface-name*—Specify the name of the interface used by the VPLS site.

The other option is explained separately.

Usage Guidelines See "Configuring the VPLS Site Interfaces" on page 399.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

interface-mac-limit

Syntax
```
interface-mac-limit limit;
```

Hierarchy Level
[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 site *site-name* interfaces *interface-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Specify the maximum number of media access control (MAC) addresses that can be learned by the VPLS routing instance. You can configure the same limit for all interfaces configured for a routing instance. You can also configure a limit for a specific interface.

Options *limit*—Specify the number of MAC addresses that can be learned from each interface.
Range: 16 through 65,536 MAC addresses
Default: 512 addresses

Usage Guidelines See "Limiting the Number of MAC Addresses Learned from an Interface" on page 403.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Topics mac-table-size

label-switched-path-template

Syntax label-switched-path-template (default-template | *p2mp-lsp-template-name*);

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel],
[edit routing-instances *routing-instance-name* provider-tunnel]

Release Information Statement introduced in JUNOS Release 8.3.

Description Enables dynamic point-to-multipoint LSPs to be used for flooding VPLS traffic. You can specify either a default template or a preconfigured template.

Options default-template—Create a point-to-multipoint LSP with the default parameters.

p2mp-lsp-template-name—Name of the point-to-multipoint LSP template.

Usage Guidelines See "Configuring Dynamic Point-to-Multipoint Flooding LSPs" on page 424.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

local-switching

Syntax local-switching;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name*],
[edit routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name*]

Release Information Statement introduced in JUNOS Release 9.3.

Description Allows you to terminate multiple Layer 2 circuit pseudowires at a single VPLS mesh group.

Usage Guidelines See "Configuring Switching Between Pseudowires Using VPLS Mesh Groups" on page 428.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

mac-tlv-receive

Syntax mac-tlv-receive;

Hierarchy Level [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 mesh-group *mesh-group-name*]
[edit routing-instances *routing-instance-name* protocols vpls]
[edit routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name*]

Release Information Statement introduced in JUNOS Release 9.3.

Description Remove MAC addresses from the MAC address database that have been learned dynamically. By removing MAC addresses, MAC address convergence can happen faster. You can clear MAC addresses globally across all devices participating in the routing instance or you can also specify a specific mesh group.

Usage Guidelines See "Removing Addresses from the MAC Address Database" on page 404.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

mac-tlv-send

Syntax mac-tlv-send;

Hierarchy Level [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 mesh-group *mesh-group-name*]
[edit routing-instances *routing-instance-name* protocols vpls]
[edit routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name*]

Release Information Statement introduced in JUNOS Release 9.3.

Description Remove MAC addresses from the MAC address database that have been learned dynamically. By removing MAC addresses, MAC address convergence can happen faster. You can clear MAC addresses globally across all devices participating in the routing instance or you can also specify a specific mesh group.

Usage Guidelines See "Removing Addresses from the MAC Address Database" on page 404.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

mac-table-aging-time

Syntax mac-table-aging-time *time*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls],
[edit routing-instances *routing-instance-name* protocols vpls]

Release Information Statement introduced in JUNOS Release 7.4.

Description Modify the timeout interval for the VPLS table.

Options *time*—Specify the number of seconds to wait between VPLS table clearings.
Range: 10 through 1,000,000 seconds
Default: 300 seconds

Usage Guidelines See "Configuring the VPLS MAC Table Timeout Interval" on page 402.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

mac-table-size

Syntax mac-table-size *size*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls],
[edit routing-instances *routing-instance-name* protocols vpls]

Release Information Statement introduced before JUNOS Release 7.4.

Description Modify the size of the VPLS MAC address table.

Options *size*—Specify the size of the MAC address table.
Range: 16 through 65,536 MAC addresses
Default: 512 MAC addresses

Usage Guidelines See "Configuring the Size of the VPLS MAC Address Table" on page 402.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

mesh-group

Syntax

```
mesh-group mesh-group-name {
    local-switching;
    mac-tlv-receive;
    mac-tlv-send;
    neighbor address;
    peer-as all;
    vpls-id number;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls],
[edit routing-instances *routing-instance-name* protocols vpls]

Release Information

Statement introduced in JUNOS Release 9.0. The local-switching, mac-tlv-receive, mac-tlv-send, and peer-as options were added in JUNOS Release 9.3.

Description

Specify the VPLS mesh group. The statement options allow you to specify each PE router that is a member of the mesh group. This statement is also used in the configuration of inter-AS VPLS with MAC operations.

Options

mesh-group-name—Specify the name of the VPLS mesh group.

neighbor *address*—Specify the address of each PE router which is a member of the mesh group.

vpls-id *number*—Specify a VPLS identifier for the mesh group.

The other statements are explained separately.

Usage Guidelines

See "Configuring Interoperability Between BGP Signaling and LDP Signaling in VPLS" on page 427 and "Configuring Inter-AS VPLS with MAC Processing at the ASBR" on page 429.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

multi-homing

Syntax multi-homing;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls site *site-name*],
[edit routing-instances *routing-instance-name* protocols vpls site *site-name*]

Release Information Statement introduced in JUNOS Release 7.5.

Description Specify the PE router as being a part of a multihomed site. Include this statement on all PE routers associated with a particular site. Configuration of this statement tracks BGP peers. If no BGP peer is available, all active interfaces for a site are deactivated..

Usage Guidelines See "Configuring Multihoming on the PE Router" on page 421.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

neighbor

Syntax neighbor *neighbor-id*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls],
[edit routing-instances *routing-instance-name* protocols vpls]

Release Information Statement introduced in JUNOS Release 8.4.

Description Specify each of the PE routers participating in the VPLS domain. Configuring this statement enables LDP for signaling VPLS.

Options *neighbor-id*—Specify the neighbor identifier for each PE router participating in the VPLS domain.

Usage Guidelines See "Configuring LDP Signaling for VPLS" on page 400.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

no-local-switching

Syntax no-local-switching;

Hierarchy Level [edit bridge-domains *bridge-domain-name*]

Release Information Statement introduced in JUNOS Release 8.5.

Description Prevents CE devices from communicating directly with each other. If the no-local-switching statement is configured, frames arriving on a CE interface are sent to VE device or core-facing interfaces only.

Usage Guidelines See "Configuring VPLS and Integrated Routing and Bridging" on page 426.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

no-tunnel-services

Syntax no-tunnel-services;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls],
[edit routing-instances *routing-instance-name* protocols vpls]

Release Information Statement introduced in JUNOS Release 7.6.

Description Configure VPLS on a router without a Tunnel Services PIC.

Usage Guidelines See "Configuring VPLS Without a Tunnel Services PIC" on page 411 and "Configuring EXP-Based Traffic Classification for VPLS" on page 404.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

peer-as

Syntax
```
peer-as {
    all;
}
```

Hierarchy Level
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name*],
[edit routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name*]

Release Information Statement introduced in JUNOS Release 9.3.

Description Enable the autonomous system border router (ASBR) to establish a single pseudowire to each of the other ASBRs interconnected using inter-AS VPLS with MAC processing at the ASBR.

Options all—This option is required. All peer routers, the ASBRs, are placed within the same VPLS mesh group.

Usage Guidelines See "Configuring Inter-AS VPLS with MAC Processing at the ASBR" on page 429.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

rsvp-te

Syntax
```
rsvp-te {
    label-switched-path-template (default-template | lsp-template-name);
    static-lsp lsp-name;
}
```

Hierarchy Level
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel],
[edit routing-instances *routing-instance-name* provider-tunnel]

Release Information Statement introduced in JUNOS Release 8.3.

Description Configure VPLS unknown unicast, broadcast, and multicast traffic flooding using point-to-multipoint LSPs.

Options static-lsp *lsp-name*—Create a static point-to-multipoint LSP and automatically include all of the neighbors in the VPLS routing instance.

The remaining option is explained separately.

Usage Guidelines See "Flooding Unknown Traffic Using Point-to-Multipoint LSPs" on page 422.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

site

Syntax

```
site site-name {
    interface interface-name {
        interface-mac-limit limit;
    }
    site-identifier identifier;
    site-preference preference-value;
}
```

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls],
[edit routing-instances *routing-instance-name* protocols vpls]

Release Information Statement introduced before JUNOS Release 7.4.

Description Specify the site name and site identifier for a site. Allows you to configure a remote site ID for remote sites.

Options *site-name*—Name of the site.

The remaining statements are explained separately.

Usage Guidelines See "Configuring the VPLS Site Name and Site Identifier" on page 396.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

site-identifier

Syntax site-identifier *identifier*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls site *site-name*],
[edit routing-instances *routing-instance-name* protocols vpls site *site-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Specify the numerical identifier for the VPLS site used as a default reference for the remote site ID.

Options *identifier*—Specify the numerical identifier for the VPLS site. The identifier must be an unsigned 16-bit number greater than zero.

Usage Guidelines See "Configuring the VPLS Site Name and Site Identifier" on page 396.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

site-preference

Syntax
site-preference *preference-value* {
  backup;
  primary;
}

Hierarchy Level
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn site *site-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls site *site-name*],
[edit routing-instances *routing-instance-name* protocols l2vpn site *site-name*],
[edit routing-instances *routing-instance-name* protocols vpls site *site-name*]

Description
Specify the preference value advertised for a particular Layer 2 VPN or VPLS site. The site preference value is encoded in the BGP local preference attribute. When a PE router receives multiple advertisements with the same VE identifier, the advertisement with the highest local preference value is preferred.

Options
preference-value—Specify the preference value advertised for a Layer 2 VPN or VPLS site.
Range: 1 through 65,535

backup—Set the preference value to 1.

primary—Set the preference value to 65,535.

Usage Guidelines
See "Configuring the VPLS Site Preference" on page 399

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

site-range

Syntax
site-range *number*;

Hierarchy Level
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls],
[edit routing-instances *routing-instance-name* protocols vpls]

Release Information
Statement introduced before JUNOS Release 7.4.

Description
Specify the maximum number of sites allowed for the VPLS domain. The value must be from 1 through 65,534.

Usage Guidelines
See "Configuring the Site Range" on page 398.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

template

Syntax template;

Hierarchy Level [edit logical-systems *logical-system-name* protocols mpls label-switched-path *p2mp-lsp-template-name*],
[edit protocols mpls label-switched-path *p2mp-lsp-template-name*

Release Information Statement introduced in JUNOS Release 8.3.

Description Specify a template for the dynamically generated point-to-multipoint LSPs used for VPLS flooding.

Usage Guidelines See "Configuring Dynamic Point-to-Multipoint Flooding LSPs with a Preconfigured Template" on page 425.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

traceoptions

Syntax

```
traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls],
[edit routing-instances *routing-instance-name* protocols vpls]

Release Information

Statement introduced before JUNOS Release 7.4.

Description

Trace traffic flowing through a VPLS routing instance.

Options

disable—(Optional) Disable the tracing operation. You can use this option to disable a single operation when you have defined a broad group of tracing operations, such as all.

file *filename*—Name of the file to receive the output of the tracing operation. Enclose the name in quotation marks (" ").

files *number*—(Optional) Maximum number of trace files. When a trace file named *trace-file* reaches this size, it is renamed *trace-file*.0. When *trace-file* again reaches its maximum size, *trace-file*.0 is renamed *trace-file*.1 and *trace-file* is renamed *trace-file*.0. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum number of files, you also must specify a maximum file size with the size option.
Range: 2 through 1000 files
Default: 2 files

flag *flag*—Tracing operation to perform. To specify more than one tracing operation, include multiple flag statements. You can specify the following tracing flags:

- all—All VPLS tracing options
- connections—VPLS connections (events and state changes)
- error—Error conditions
- nlri—VPLS advertisements received or sent by means of the BGP
- route—Routing information
- topology—VPLS topology changes caused by reconfiguration or advertisements received from other provider edge (PE) routers using BGP

flag-modifier—(Optional) Modifier for the tracing flag. You can specify the following modifiers:

- detail—Provide detailed trace information

- disable—Disable the tracing flag
- receive—Trace received packets
- send—Trace sent packets

no-world-readable—Do not allow any user to read the log file.

size *size*—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named *trace-file* reaches this size, it is renamed *trace-file*.0. When *trace-file* again reaches its maximum size, *trace-file*.0 is renamed *trace-file*.1 and *trace-file* is renamed *trace-file*.0. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum file size, you also must specify a maximum number of trace files with the files option.
Syntax: *x*k to specify kilobytes, *x*m to specify megabytes, or *x*g to specify gigabytes
Range: 10 KB through the maximum file size supported on your system
Default: 1 MB

world-readable—Allow any user to read the log file.

Usage Guidelines See "Tracing VPLS Traffic and Operations" on page 430.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

tunnel-services

Syntax

```
tunnel-services {
    devices device-names;
    primary primary-device-name;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls]
[edit routing-instances *routing-instance-name* protocols vpls]

Release Information

Statement introduced before JUNOS Release 7.4.

Description

Specify that traffic for particular VPLS routing instances be forwarded to specific virtual tunnel (VT) interfaces, allowing you to load-balance VPLS traffic among all the available VT interfaces on the router.

Options

devices *device-names*—Specify the VT interfaces acceptable for use by the VPLS routing instance. If you do not configure this option, all VT interfaces available to the router can be used for de-encapsulating traffic for this instance.

primary *primary-device-name*—Specify the primary VT interface to be used by the VPLS routing instance. The VT interface specified is used to de-encapsulate all VPLS traffic from the MPLS core network for this routing instance. If the VT interface specified is unavailable, then one of the other acceptable VT interfaces is used for handling the VPLS traffic. If you do not configure this option, any acceptable VT interface can be used to de-encapsulate VPLS traffic from the core.

Usage Guidelines

See "Specifying the VT Interfaces Used by VPLS Routing Instances" on page 418.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

vlan-id

Syntax vlan-id *number*;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]

Release Information Statement introduced before JUNOS Release 7.4.

Description For Fast Ethernet and Gigabit Ethernet interfaces only, bind an 802.1Q VLAN tag ID to a logical interface.

Options *number*—A valid VLAN identifier.
Range: For 4-port Fast Ethernet PICs configured to handle VPLS traffic, 512 through 1023. For 1-port and 10-port Gigabit Ethernet PICs configured to handle VPLS traffic, 512 through 4094.

Usage Guidelines See “Enabling VLAN Tagging” on page 408.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

vlan-tagging

Syntax vlan-tagging;

Hierarchy Level [edit interfaces *interface-name*],
[edit logical-systems *logical-system-name* interfaces *interface-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description For Fast Ethernet and Gigabit Ethernet interfaces only, enable the reception and transmission of 802.1Q VLAN-tagged frames on the interface.

Usage Guidelines See “Enabling VLAN Tagging” on page 408.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

vpls

See the following sections:

vpls (Interfaces)

Syntax vpls;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* family],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family]

Release Information Statement introduced before JUNOS Release 7.4.

Description Specify the VPLS protocol family information for the logical interface.

Usage Guidelines See "Configuring Interfaces for VPLS Routing" on page 405.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

vpls (Routing Instance)

Syntax

```
vpls {
    active-interface {
        any;
        primary interface-name;
    }
    connectivity-type (ce | irb);
    interface-mac-limit limit;
    mac-table-aging-time time;
    mac-table-size size;
    mac-tlv-receive;
    mac-tlv-send;
    mesh-group mesh-group-name {
        local-switching;
        mac-tlv-receive;
        mac-tlv-send;
        neighbor address;
        peer-as all;
    vpls-id number;
    no-tunnel-services;
    site site-name {
        interface interface-name {
            interface-mac-limit limit;
        }
        multi-homing;
        site-identifier identifier;
        site-preference preference-value;
    }
    site-range number;
    traceoptions {
        file filename <files number> <size size> <world-readable | no-world-readable>;
        flag flag <flag-modifier> <disable>;
    }
    tunnel-services {
        devices device-names;
        primary primary-device-name;
    }
}
```

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols],
[edit routing-instances *routing-instance-name* protocols]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure a VPLS routing instance.

The remaining statements are explained separately.

Usage Guidelines See “Configuring the VPLS Routing Instance” on page 395.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

vpls-id

Syntax vpls-id *vpls-id*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls],
[edit routing-instances *routing-instance-name* protocols vpls]

Release Information Statement introduced in JUNOS Release 8.4.

Description Identify the virtual circuit identifier used for the VPLS routing instance. This statement is a part of the configuration to enable LDP signaling for VPLS.

Options *vpls-id*—Specify a valid identifier for the VPLS routing instance.

Usage Guidelines See "Configuring LDP Signaling for VPLS" on page 400.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Part 6
Interprovider and Carrier-of-Carriers

Chapter 21

Interprovider and Carrier-of-Carriers VPNs Overview

This chapter describes in detail the operation of interprovider and carrier-of-carriers virtual private networks (VPNs) as described in RFC 2547bis, *BGP/MPLS VPNs*. As VPNs are deployed on the Internet, the customer of a VPN service provider might be another service provider rather than an end customer. The customer service provider depends on the VPN service provider to deliver a VPN transport service between the customer service provider's points of presence (POPs) or regional networks.

If the customer service provider's sites have different autonomous system (AS) numbers, then the VPN transit service provider supports carrier-of-carrier VPN service for the interprovider VPN service. If the customer service provider's sites have the same AS number, then the VPN transit service provider delivers a carrier-of-carriers VPN service.

This chapter discusses the following topics, which provide background information about carrier-of-carriers VPNs:

Interprovider and Carrier-of-Carriers VPN Standards

Interprovider and carrier-of-carriers VPNs are defined by the following documents:

- RFC 3107, *Carrying Label Information in BGP-4*.
- RFC 4364, *BGP/MPLS IP Virtual Private Networks (VPNs)*

To access Internet RFCs and drafts, go to the IETF Web site at http://www.ietf.org/.

Traditional VPNs, Interprovider VPNs, and Carrier-of-Carriers VPNs

The sections that follow provide an overview of traditional VPNs, interprovider and carrier-of-carriers VPNs, and the differences in how external and internal routes are handled in each of these environments.

In traditional IP routing architectures, there is a clear distinction between internal routes and external routes. From the perspective of an Internet service provider (ISP), internal routes include all the provider's internal links (including BGP next hops) and loopback interfaces. These internal routes are exchanged with other routing platforms in the ISP's network by means of an interior gateway protocol (IGP), such as Open Shortest Path First (OSPF) or Intermediate System-to-Intermediate System (IS-IS). All routes learned at Internet peering points or from customer sites are classified as external routes and are distributed by means of an exterior gateway protocol (EGP) such as BGP. In traditional IP routing architectures, the number of internal routes is typically much smaller than the number of external routes.

Standard VPNs

The traditional distinction between internal routes and external routes also applies to VPN routing architectures. As shown in Figure 1 on page 4, the provider (P) routers maintain only the service provider's internal routes (to provider edge [PE] routers and other P routers); they do not maintain VPN routes. PE routers are the only devices in the provider network that are required to maintain external routes.

The BGP next hop connects the external routes to the internal routes in traditional VPNs:

- The BGP next hop is advertised with each external route in BGP advertisements.
- The route to the BGP next hop is an internal route that is advertised by the IGP.
- Multiprotocol Label Switching (MPLS) provides packet forwarding from the ingress PE router to the BGP next-hop egress PE router.

Interprovider and Carrier-of-Carriers VPNs

All interprovider and carrier-of-carriers VPNs share the following characteristics:

- Each interprovider or carrier-of-carriers VPN customer must distinguish between internal and external customer routes.
- Internal customer routes must be maintained by the VPN service provider in its PE routers.
- External customer routes are carried only by the customer's routing platforms, not by the VPN service provider's routing platforms.

The key difference between interprovider and carrier-of-carriers VPNs is whether the customer sites belong to the same AS or to separate ASs:

- "Interprovider VPNs" on page 461—The customer sites belong to different ASs. You need to configure external BGP (EBGP) to exchange the customer's external routes.
- "Carrier-of-Carriers VPNs" on page 463—The customer sites belong to the same AS. You need to configure internal BGP (IBGP) to exchange the customer's external routes.

In general, each service provider in a VPN hierarchy is required to maintain its own internal routes in its P routers, and the internal routes of its customers in its PE routers. By recursively applying this rule, it is possible to create a hierarchy of VPNs.

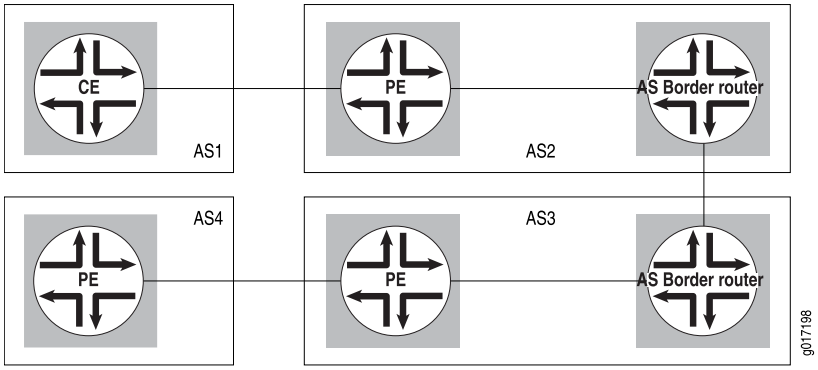
The following are definitions of the types of PE routers specific to interprovider and carrier-of-carriers VPNs:

- The AS border router is located at the AS border and handles traffic leaving and entering the AS.
- The end PE router is the PE router in the customer VPN; it is connected to the CE router at the end customer's site.

Interprovider VPNs

Interprovider VPNs provide connectivity between separate ASs. This functionality might be used by a VPN customer who has connections to several different ISPs, or different connections to the same ISP in different geographic regions, each of which has a different AS. Figure 49 on page 461 illustrates the type of network topology used by an interprovider VPN.

Figure 49: Interprovider VPN Network Topology



The following sections describe the ways you can configure an interprovider VPN:

- Linking VRF Tables Between Autonomous Systems on page 461
- Configuring MP-EBGP Between AS Border Routers on page 462
- Configuring Multihop MP-EBGP Between AS Border Routers on page 462

Linking VRF Tables Between Autonomous Systems

You can connect two separate ASs by simply linking the VPN routing and forwarding (VRF) table in the AS border router of one AS to the VRF table in the AS border router in the other AS. Each AS border router must contain a VRF instance for every VPN configured in both service provider networks. You then configure an IP session between the two AS border routers. In effect, the AS border routers treat each other as customer edge (CE) routers.

Because of the complexity of the configuration, particularly with regard to scaling, this method is not recommended. The details of this configuration are not provided in this manual.

Configuring MP-EBGP Between AS Border Routers

In this approach, the PE routers within an AS use multiprotocol external BGP (MP-EBGP) to distribute labeled VPN–Internet Protocol version 4 (IPv4) routes to an AS border router or to a route reflector of which the AS border router is a client. The AS border router uses multiprotocol external BGP (MP-EBGP) to distribute the labeled VPN-IPv4 routes to its peer AS border router in the neighboring AS. The peer AS border router then uses MP-IBGP to distribute labeled VPN-IPv4 routes to PE routers, or to a route reflector of which the PE routers are a client.

This approach enhances the scalability of an EBGP VRF-to-VRF configuration because it eliminates the need to configure all the VPNs on every AS border router. However, it also introduces some complexity:

- All the VRF routes must be stored in the AS border router.
- An LSP must be established from ingress PE routers to egress PE routers.
- Secure connections must exist among the ASs along the path from the ingress PE router to the egress PE router.
- The ASs must be configured to store information about which AS border routers receive routes with specific route target attributes.

Configuring Multihop MP-EBGP Between AS Border Routers

In this type of interprovider VPN configuration, P routers do not need to store all the routes in all the VPNs. Only the PE routers must have all the VPN routes. The P routers simply forward traffic to the PE routers—they do not store or process any information about the packets' destination. The connections between the AS border routers in separate ASs forward traffic between the ASs, much as a label-switched path (LSP) works.

The following are the basic steps you take to configure an interprovider VPN in this manner:

1. Configure multihop EBGP redistribution of labeled VPN-IPv4 routes between the source and destination ASs.
2. Configure EBGP to redistribute labeled IPv4 routes from its AS to neighboring ASs.
3. Configure MPLS on the end PE routers of the VPNs.

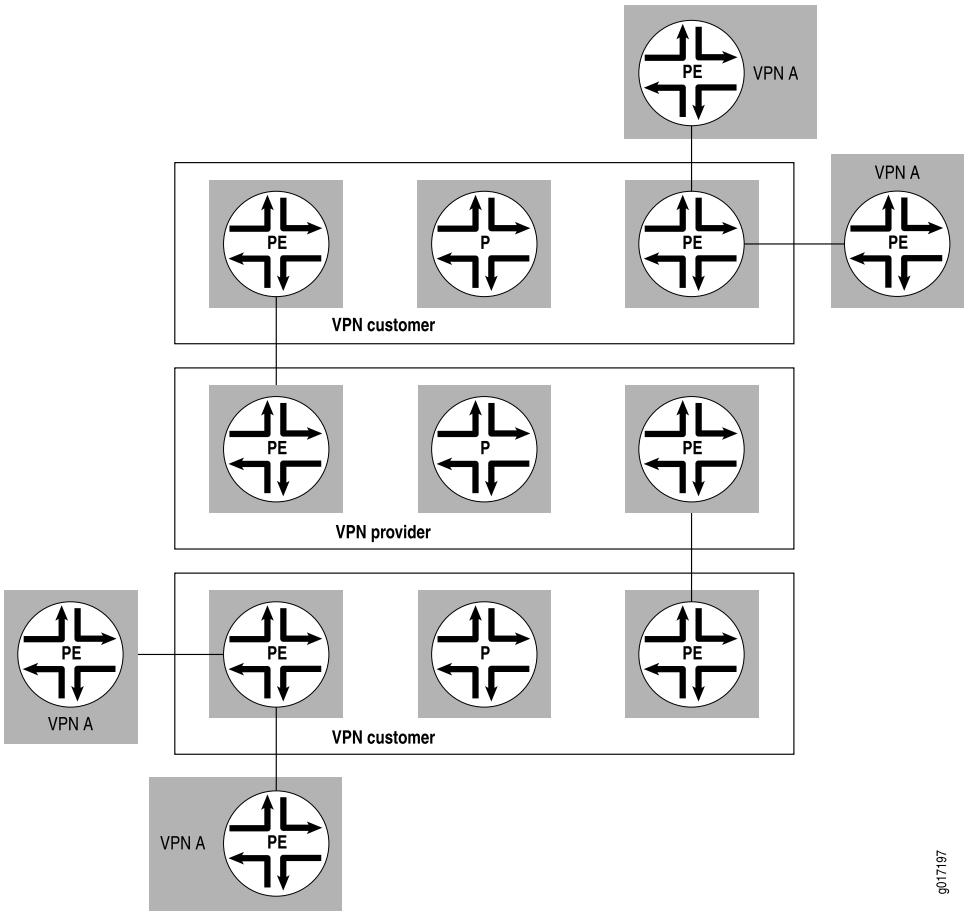
Carrier-of-Carriers VPNs

The customer of a VPN service provider might be a service provider for the end customer. The following are the two main types of carrier-of-carriers VPNs (as described in RFC 4364:

- "Internet Service Provider as the Customer" on page 464—The VPN customer is an ISP that uses the VPN service provider's network to connect its geographically disparate regional networks. The customer does not have to configure MPLS within its regional networks.
- "VPN Service Provider as the Customer" on page 464—The VPN customer is itself a VPN service provider offering VPN service to its customers. The carrier-of-carriers VPN service customer relies on the backbone VPN service provider for inter-site connectivity. The customer VPN service provider is required to run MPLS within its regional networks.

Figure 50 on page 463 illustrates the network architecture used for a carrier-of-carriers VPN service.

Figure 50: Carrier-of-Carriers VPN Architecture



Internet Service Provider as the Customer

In this type of carrier-of-carriers VPN configuration, ISP A configures its network to provide Internet service to ISP B. ISP B provides the connection to the customer wanting Internet service, but the actual Internet service is provided by ISP A.

This type of carrier-of-carriers VPN configuration has the following characteristics:

- The carrier-of-carriers VPN service customer (ISP B) does not need to configure MPLS on its network.
- The carrier-of-carriers VPN service provider (ISP A) must configure MPLS on its network.
- MPLS must also be configured on the CE routers and PE routers connected together in the carrier-of-carriers VPN service customer's and carrier-of-carriers VPN service provider's networks.

VPN Service Provider as the Customer

A VPN service provider can have customers that are themselves VPN service providers. In this type of configuration, also called a hierarchical or recursive VPN, the customer VPN service provider's VPN-IPv4 routes are considered external routes, and the backbone VPN service provider does not import them into its VRF table. The backbone VPN service provider imports only the customer VPN service provider's internal routes into its VRF table.

This type of configuration is similar to the configuration described in the "Internet Service Provider as the Customer" on page 464 section. The similarities and differences are shown in Table 14 on page 464.

Table 14: Comparison of Interprovider and Carrier-of-Carriers VPNs

Feature	ISP Customer	VPN Service Provider Customer
Customer edge device	AS border router	PE router
IBGP sessions	Carry IPv4 routes	Carry external VPN-IPv4 routes with associated labels
Forwarding within the customer network	MPLS is optional	MPLS is required

Chapter 22

Configuring Interprovider and Carrier-of-Carriers VPNs

To configure interprovider or carrier-of-carriers virtual private network (VPN) functionality, you typically need to include the labeled-unicast statement in the configuration for the BGP on the autonomous system (AS) border routers of an interprovider VPN or the provider edge (PE) and customer edge (CE) routers of a carrier-of-carriers VPN. You must also configure the provider (P) routers in the service provider's and service customer's networks.

To configure interprovider or carrier-of-carriers VPN functionality, include the bgp statement:

```
bgp {
    group group-name {
        type internal;
        local-address address;
        family inet {
            labeled-unicast {
                resolve-vpn;
            }
        }
        neighbor address;
    }
}
```

You can include the bgp statement at the following hierarchy levels:

- [edit protocols]
- [edit logical-systems *logical-system-name* protocols]

This chapter describes how to configure interprovider and carrier-of-carriers VPNs, discussing the following topics:

Configuring Interprovider VPNs

You can configure interprovider VPN service using either multiprotocol external BGP (MP-EBGP) or multihop MP-EBGP:

- Configuring Interprovider VPNs Using MP-EBGP on page 466
- Configuring Interprovider VPNs Using Multihop MP-EBGP on page 468

Configuring Interprovider VPNs Using MP-EBGP

To configure interprovider VPN service using MP-EBGP, you need to configure the AS border routers of each AS. For an illustration of how the routers interconnect in an interprovider VPN service, see Figure 49 on page 461.

The configuration of the AS border routers in each AS is nearly identical. To configure each AS border router, you perform the steps in the following sections:

- Configuring RSVP on page 466
- Configuring MPLS on page 466
- Configuring BGP on page 467
- Configuring OSPF on page 467

Configuring RSVP

You need to configure the interprovider VPN interface in Resource Reservation Protocol (RSVP). This interface on the PE router, which handles VPN traffic in the current AS, receives VPN traffic from the other AS.

Configure the interface for RSVP by including the interface statement:

```
interface interface-name;
```

You can include the interface statement at the following hierarchy levels:

- [edit protocols rsvp]
- [edit logical-systems *logical-system-name* protocols rsvp]

Configuring MPLS

Configure a label-switched path (LSP) to the PE router. Also configure the interfaces handling VPN traffic from the other AS and to the PE router in the current AS.

```
mpls {
    label-switched-path path-name {
        to address;
    }
    interface interface-name;
    interface interface-name;
}
```

You can include the mpls statement at the following hierarchy levels:

- [edit protocols]
- [edit logical-systems *logical-system-name* protocols]

Configuring BGP

Configure an MP-EBGP session on the AS border router. This session exchanges VPN Internet Protocol version 4 (IPv4) routes with the AS border router in the other AS.

To configure a group to handle internal BGP (IBGP) and a group to handle external BGP (EBGP), include the bgp statement:

```
bgp {
    keep all;
    group group-name {
        type internal;
        local-address address;
        family inet-vpn {
            unicast;
        }
        neighbor address;
    }
    group group-name {
        type external;
        family inet-vpn {
            unicast;
        }
        neighbor address {
            peer-as as number;
        }
    }
}
```

You can include the bgp statement at the following hierarchy levels:

- [edit protocols]
- [edit logical-systems *logical-system-name* protocols]

Configuring OSPF

To configure Open Shortest Path First (OSPF) on the AS border router, include the ospf statement:

```
ospf {
    traffic engineering;
    area address {
        interface interface-name;
        interface interface-name {
            passive;
        }
    }
}
```

You can include the ospf statement at the following hierarchy levels:

- [edit protocols]
- [edit logical-systems *logical-system-name* protocols]

Configuring Interprovider VPNs Using Multihop MP-EBGP

To configure a network to provide interprovider VPN service using multihop MP-EBGP, you need to set up the AS border routers and the PE routers connected to the end customer's CE routers. For an illustration of how the routers interconnect in an interprovider VPN service, see Figure 49 on page 461.

The following sections describe how to configure a network to provide interprovider VPN service using multihop MP-EBGP:

- Configuring the AS Border Routers on page 468
- Configuring the PE Router on page 469

Configuring the AS Border Routers

The configuration of the AS border routers in each AS is nearly identical. To configure each AS border router, you perform the steps in the following sections:

- Configuring BGP on page 468
- Configuring Policy Options on page 469

Configuring BGP

Configure BGP on the AS border routers. To configure a group for IBGP to the PE router, include the bgp statement:

```
bgp {
   group group-name {
      type internal;
      local-address address;
      family inet {
         labeled-unicast {
            resolve-vpn;
         }
      }
      neighbor address;
   }
}
```

To configure a group for EBGP to the AS border router in the adjacent AS router, include the bgp statement:

```
bgp {
   group group-name {
      type external;
      family inet {
         labeled-unicast;
      }
```

```
        export internal;
        neighbor address {
            peer-as as-number;
        }
    }
}
```

You can include the bgp statement at the following hierarchy levels:

- [edit protocols]
- [edit logical-systems *logical-system-name* protocols]

Configuring Policy Options

For the policy configuration on the AS border routers, you only need to advertise the loopbacks of the PE routers. If the AS border router is also a PE router, configure from protocol ospf direct at the [edit policy-options policy-statement *policy-name* term *term-name*] hierarchy level.

To configure the policy options on the AS border routers, include the policy-statement statement:

```
policy-statement policy-name {
    term term-name {
        from {
            protocol ospf direct;
            route-filter pe-router-loopback-address exact accept;
        }
        then reject;
    }
}
```

You can include these statements at the following hierarchy levels:

- [edit policy-options]
- [edit logical-systems *logical-system-name* policy-options]

Configuring the PE Router

Configure a multihop MP-EBGP session on the PE router connected to the end customer's CE router.

To pass labeled IPv4 routes, include the labeled-unicast statement:

```
labeled-unicast {
    resolve-vpn;
}
```

You can include the labeled-unicast statement at the following hierarchy levels:

- [edit protocols bgp group *group-name* family inet]

- [edit logical-systems *logical-system-name* protocols bgp group *group-name* family inet]

To configure a group to handle an EBGP multihop session with the remote PE router (that is, to pass VPN-IPv4 routes), include the bgp statement:

```
bgp {
   group group-name {
      multihop {
         ttl 10;
      }
      family inet-vpn {
         unicast;
      }
   }
   neighbor address {
      peer-as as-number;
   }
}
```

You can include the bgp statement at the following hierarchy levels:

- [edit protocols]
- [edit logical-systems *logical-system-name* protocols]

Configuring Carrier-of-Carriers VPNs

You can configure a carrier-of-carriers VPN service for customers who want to provide Internet service or for customers who want to provide VPN service:

- Configuring Carrier-of-Carriers VPN—Customer Provides Internet Service on page 470
- Configuring Carrier-of-Carriers VPN—Customer Provides VPN Service on page 476

Configuring Carrier-of-Carriers VPN—Customer Provides Internet Service

In this type of carrier-of-carriers VPN service configuration, the customer provides basic Internet service. The carrier-of-carriers VPN service provider must configure Multiprotocol Label Switching (MPLS) in its network, although this configuration is optional for the carrier service customer. Figure 50 on page 463 shows how the routers in this type of service interconnect.

To configure a carrier-of-carriers VPN, perform the tasks described in the following sections:

- Configuring the Carrier-of-Carriers VPN Service Customer's CE Router on page 471
- Configuring the Carrier-of-Carriers VPN Service Provider's PE Routers on page 473

Configuring the Carrier-of-Carriers VPN Service Customer's CE Router

The carrier-of-carriers VPN service customer's router acts as a CE router with respect to the service provider's PE router. The following sections describe how to configure the carrier-of-carriers VPN service customer's CE router:

Configuring MPLS

To configure MPLS on the customer's CE router, include the mpls statement:

```
mpls {
    traffic-engineering bgp-igp;
    interface interface-name;
}
```

You can include the mpls statement at the following hierarchy levels:

- [edit protocols]
- [edit logical-systems *logical-system-name* protocols]

Configuring BGP

To configure a group to collate the customer's internal routes, include the bgp statement:

```
bgp {
    group group-name {
        type internal;
        local-address address;
        neighbor address;
    }
}
```

You can include the bgp statement at the following hierarchy levels:

- [edit protocols]
- [edit logical-systems *logical-system-name* protocols]

The customer's CE router must be able to send labels to the VPN service provider's router. Enable this by including the labeled-unicast statement under the bgp statement:

```
bgp {
    group group-name {
        export internal;
        peer-as as-number;
        neighbor address {
```

```
            family inet {
               labeled-unicast;
            }
         }
      }
   }
```

You can include the bgp statement at the following hierarchy levels:

- [edit logical-systems *logical-system-name* protocols]
- [edit protocols]

Configuring OSPF

To configure OSPF on the customer's CE router, include the ospf statement:

```
ospf {
   area area-id {
      interface interface-name {
         passive;
      }
      interface interface-name;
   }
}
```

You can include the ospf statement at the following hierarchy levels:

- [edit protocols]
- [edit logical-systems *logical-system-name* protocols]

Configuring Policy Options

To configure policy options on the customer's CE router, include the policy-statement statement:

```
policy-statement statement-name {
   term term-name {
      from protocol [ospf direct ldp];
      then accept;
   }
   term term-name {
      then reject;
   }
}
```

You can include the policy-statement statement at the following hierarchy levels:

- [edit policy-options]
- [edit logical-systems *logical-system-name* policy-options]

Configuring the Carrier-of-Carriers VPN Service Provider's PE Routers

The service provider's PE routers connect to the customer's CE routers and forward the customer's VPN traffic across the provider's network.

The following sections describe how to configure the carrier-of-carriers VPN service provider's PE routers:

Configuring MPLS

To configure MPLS on the provider's PE routers, include the mpls statement:

```
mpls {
    interface interface-name;
    interface interface-name;
}
```

You can include the mpls statement at the following hierarchy levels:

- [edit protocols]
- [edit logical-systems *logical-system-name* protocols]

Configuring BGP

To configure a BGP session with the provider PE router at the other end of the provider's network, include the bgp statement:

```
bgp {
    group group-name {
        type internal;
        local-address address;
        family inet-vpn {
            any;
        }
        neighbor address;
    }
}
```

You can include the bgp statement at the following hierarchy levels:

- [edit protocols]
- [edit logical-systems *logical-system-name* protocols]

Configuring IS-IS

To configure Intermediate System-to-Intermediate System (IS-IS) on the provider's PE routers, include the isis statement:

```
isis {
    interface interface-name;
    interface interface-name {
        passive;
    }
}
```

You can include the isis statement at the following hierarchy levels:

- [edit protocols]
- [edit logical-systems *logical-system-name* protocols]

Configuring LDP

To configure the Label Distribution Protocol (LDP) on the provider's PE routers, include the ldp statement:

```
ldp {
    interface interface-name;
}
```

You can include the ldp statement at the following hierarchy levels:

- [edit protocols]
- [edit logical-systems *logical-system-name* protocols]

Configuring a Routing Instance

To configure Layer 3 VPN service with the customer's CE router, include the labeled-unicast statement within the routing instance so the PE router can send labels to the customer's CE router:

```
routing-instance-name {
    instance-type vrf;
    interface interface-name;
    route-distinguisher address;
    vrf-import policy-name;
    vrf-export policy-name;
    protocols {
        bgp {
            group group-name {
                peer-as as-number;
                neighbor address {
                    family inet {
                        labeled-unicast;
                    }
                }
            }
```

```
        }
    }
}
```

You can include these statements at the following hierarchy levels:

- [edit routing-instances]
- [edit logical-systems *logical-system-name* routing-instances]

Configuring Policy Options

To configure a policy statement to import routes from the customer's CE router, include the policy-statement statement:

```
policy-statement policy-name {
    term term-name {
        from {
            protocol bgp;
            community community-name;
        }
        then accept;
    }
    term term-name {
        then reject;
    }
}
```

You can include the policy-statement statement at the following hierarchy levels:

- [edit policy-options]
- [edit logical-systems *logical-system-name* policy-options]

To configure a policy statement to export routes to the customer's CE router, include the policy-statement and community statements:

```
policy-statement policy-name {
    term term-name {
        from protocol bgp;
        then {
            community add community-name;
            accept;
        }
    }
    term term-name {
        then reject;
    }
}
community community-name members value;
```

You can include these statements at the following hierarchy levels:

- [edit policy-options]
- [edit logical-systems *logical-system-name* policy-options]

Configuring Carrier-of-Carriers VPN—Customer Provides VPN Service

Figure 50 on page 463 shows how the routers in this type of service interconnect.

To configure the following routers in the customer's and provider's networks to enable carrier-of-carriers VPN service, you perform the steps in the following sections:

- Configuring the Carrier-of-Carriers Customer's PE Router on page 476
- Configuring the Carrier-of-Carriers Customer's CE Router on page 479
- Configuring the Provider's PE Router on page 481

Configuring the Carrier-of-Carriers Customer's PE Router

The carrier-of-carriers customer's PE router is connected to the end customer's CE router.

The following sections describe how to configure the carrier-of-carriers customer's PE router:

- Configuring MPLS on page 476
- Configuring BGP on page 476
- Configuring OSPF on page 477
- Configuring LDP on page 477
- Configuring VPN Service in the Routing Instance on page 478
- Configuring Policy Options on page 478

Configuring MPLS

To configure MPLS on the carrier-of-carriers customer's PE router, include the mpls statement:

```
mpls {
    interface interface-name;
    interface interface-name;
}
```

You can include the mpls statement at the following hierarchy levels:

- [edit protocols]
- [edit logical-systems *logical-system-name* protocols]

Configuring BGP

Configure the labeled-unicast statement on the IBGP session to the carrier-of-carriers customer's CE router (see "Configuring the Carrier-of-Carriers Customer's CE Router" on page 479), and configure the family-inet-vpn statement for the IBGP session to the carrier-of-carriers PE router on the other side of the network:

```
bgp {
    group group-name {
```

```
        type internal;
        local-address address;
        neighbor address {
            family inet {
                labeled-unicast;
                resolve-vpn;
            }
        }
    }
    neighbor address {
        family inet-vpn {
            any;
        }
    }
}
```

You can include these statements at the following hierarchy levels:

- [edit protocols]
- [edit logical-systems *logical-system-name* protocols]

Configuring OSPF

To configure OSPF on the carrier-of-carriers customer's PE router, include the ospf statement:

```
ospf {
    area area-id {
        interface interface-name {
            passive;
        }
        interface interface-name;
    }
}
```

You can include the ospf statement at the following hierarchy levels:

- [edit protocols]
- [edit logical-systems *logical-system-name* protocols]

Configuring LDP

To configure LDP on the carrier-of-carriers customer's PE router, include the ldp statement:

```
ldp {
    interface interface-name;
}
```

You can include the ldp statement at the following hierarchy levels:

- [edit protocols]
- [edit logical-systems *logical-system-name* protocols]

Configuring VPN Service in the Routing Instance

To configure VPN service for the end customer's CE router on the carrier-of-carriers customer's PE router, include the following statements:

```
instance-type vrf;
interface interface-name;
route-distinguisher address;
vrf-import policy-name;
vrf-export policy-name;
protocols {
   bgp {
      group group-name {
         peer-as as-number;
         neighbor address;
      }
   }
}
```

You can include these statements at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*]

Configuring Policy Options

To configure policy options to import and export routes to and from the end customer's CE router, include the policy-statement and community statements:

```
policy-statement policy-name {
   term term-name {
      from {
         protocol bgp;
         community community-name;
      }
      then accept;
   }
   term term-name {
      then reject;
   }
}
policy-statement policy-name {
   term term-name {
      from protocol bgp;
      then {
         community add community-name;
         accept;
      }
   }
   term term-name {
      then reject;
   }
}
community community-name members value;
```

You can include these statements at the following hierarchy levels:

- [edit policy-options]
- [edit logical-systems *logical-system-name* policy-options]

Configuring the Carrier-of-Carriers Customer's CE Router

The carrier-of-carriers customer's CE router connects to the provider's PE router. Complete the instructions in the following sections to configure the carrier-of-carriers customers' CE router:

- Configuring MPLS on page 479
- Configuring BGP on page 479
- Configuring OSPF and LDP on page 480
- Configuring Policy Options on page 480

Configuring MPLS

In the MPLS configuration for the carrier-of-carriers customer's CE router, include the interfaces to the provider's PE router and to a P router in the customer's network:

```
mpls {
    traffic-engineering bgp-igp;
    interface interface-name;
    interface interface-name;
}
```

You can include these statements at the following hierarchy levels:

- [edit protocols]
- [edit logical-systems *logical-system-name* protocols]

Configuring BGP

In the BGP configuration for the carrier-of-carriers customer's CE router, configure a group that includes the labeled-unicast statement to extend VPN service to the PE router connected to the end customer's CE router:

```
bgp {
    group group-name {
        type internal;
        local-address address;
        neighbor address {
            family inet {
                labeled-unicast;
            }
        }
    }
}
```

You can include the bgp statement at the following hierarchy levels:

- [edit protocols]
- [edit logical-systems *logical-system-name* protocols]

To configure a group to send labeled internal routes to the provider's PE router, include the bgp statement:

```
bgp {
   group group-name {
      export internal;
      peer-as as-number;
      neighbor address {
         family inet {
            labeled-unicast;
         }
      }
   }
}
```

You can include the bgp statement at the following hierarchy levels:

- [edit logical-systems *logical-system-name* protocols]
- [edit protocols]

Configuring OSPF and LDP

To configure OSPF and LDP on the carrier-of-carriers customer's CE router, include the ospf and ldp statements:

```
ospf {
   area area-id {
      interface interface-name {
         passive;
      }
      interface interface-name;
   }
}
ldp {
   interface interface-name;
}
```

You can include these statements at the following hierarchy levels:

- [edit protocols]
- [edit logical-systems *logical-system-name* protocols]

Configuring Policy Options

To configure the policy options on the carrier-of-carriers customer's CE router, include the policy-statement statement:

```
policy-statement policy-statement-name {
   term term-name {
```

```
        from protocol [ ospf direct ldp ];
        then accept;
    }
    term term-name {
        then reject;
    }
}
```

You can include the policy-statement statement at the following hierarchy levels:

- [edit policy-options]
- [edit logical-systems *logical-system-name* policy-options]

Configuring the Provider's PE Router

The carrier-of-carriers provider's PE routers connect to the carrier customer's CE routers. Complete the instructions in the following sections to configure the provider's PE router:

- Configuring MPLS on page 481
- Configuring a PE-Router-to-PE-Router BGP Session on page 481
- Configuring IS-IS and LDP on page 482
- Configuring Policy Options on page 482
- Configuring a Routing Instance to Send Routes to the CE Router on page 483

Configuring MPLS

In the MPLS configuration, specify at least two interfaces—one to the customer's CE router and one to connect to the provider's PE router on the other side of the provider's network:

```
interface interface-name;
interface interface-name;
```

You can include these statements at the following hierarchy levels:

- [edit protocols mpls]
- [edit logical-systems *logical-system-name* protocols mpls]

Configuring a PE-Router-to-PE-Router BGP Session

To configure a PE-router-to-PE-router BGP session on the provider's PE routers to allow VPN-IPv4 routes to pass between the PE routers, include the bgp statement:

```
bgp {
    group group-name {
        type internal;
        local-address address;
        family inet-vpn {
            any;
        }
```

```
        neighbor address;
    }
}
```

You can include the bgp statement at the following hierarchy levels:

- [edit protocols]
- [edit logical-systems *logical-system-name* protocols]

Configuring IS-IS and LDP

To configure IS-IS and LDP on the provider's PE routers, include the isis and ldp statements:

```
isis {
    interface interface-name;
    interface interface-name {
        passive;
    }
}
ldp {
    interface interface-name;
}
```

You can include these statements at the following hierarchy levels:

- [edit protocols]
- [edit logical-systems *logical-system-name* protocols]

Configuring Policy Options

To configure policy statements on the provider's PE router to export routes to and import routes from the carrier customer's network, include the policy-statement and community statements:

```
policy-statement statement-name {
    term term-name {
        from {
            protocol bgp;
            community community-name;
        }
        then accept;
    }
    term term-name {
        then reject;
    }
}
policy-statement statement-name {
    term term-name {
        from protocol bgp;
        then {
            community add community-name;
            accept;
```

```
        }
    }
    term term-name {
        then reject;
    }
}
community community-name members value;
```

You can include these statements at the following hierarchy levels:

- [edit policy-options]
- [edit logical-systems *logical-system-name* policy-options]

Configuring a Routing Instance to Send Routes to the CE Router

To configure the routing instance on the provider's PE router to send labeled routes to the carrier customer's CE router, include the following statements:

```
instance-type vrf;
interface interface-name;
route-distinguisher value;
vrf-import policy-name;
vrf-export policy-name;
protocols {
    bgp {
        group group-name {
            peer-as as-number;
            neighbor address {
                family inet {
                    labeled-unicast;
                }
            }
        }
    }
}
```

You can include these statements at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*]

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

You can configure BGP to gather traffic statistics for interprovider and carrier-of-carriers VPNs.

To configure BGP to gather traffic statistics for interprovider and carrier-of-carriers VPNs, include the **traffic-statistics** statement:

```
traffic-statistics {
    file filename <world-readable | no-world-readable>;
    interval seconds;
```

}

For a list of the hierarchy levels at which you can include the traffic-statistics statement, see the summary section for this statement.

NOTE: Traffic statistics for interprovider and carrier-of-carriers VPNs are available only for IPv4. IPv6 is not supported.

If you do not specify a filename, the statistics are not written to a file. However, if you have included the traffic-statistics statement in the BGP configuration, the statistics are still available and can be accessed by means of the show bgp group traffic-statistics *group-name* command.

To account for traffic from each customer separately, separate labels must be advertised for the same prefix to the peer routers in different groups. To enable separate traffic accounting, you need to include the per-group-label statement in the configuration for each BGP group. By including this statement, statistics are collected and displayed that account for traffic sent by the peers of the specified BGP group.

If you configure the statement at the [edit protocols bgp family inet] hierarchy level, rather than configuring it for a specific BGP group, then the traffic statistics are shared with all BGP groups configured with the traffic-statistics statement but not configured with the per-group-label statement.

To account for traffic from each customer separately, include the per-group-label statement in the configuration for each BGP group:

```
per-group-label;
```

For a list of the hierarchy levels at which you can include the per-group-label statement, see the summary section for this statement.

The following shows a sample of the output to the traffic statistics file:

```
Dec 19 10:39:54 Statistics for BGP group ext2 (Index 1) NLRI inet-labeled-unicast
Dec 19 10:39:54  FEC               Packets         Bytes   EgressAS   FECLabel
Dec 19 10:39:54 10.255.245.55            0             0          I     100160
Dec 19 10:39:54 10.255.245.57            0             0          I     100112
Dec 19 10:39:54 100.101.0.0              0             0         25     100080
Dec 19 10:39:54 100.102.0.0              0             0         25     100080
Dec 19 10:39:54 100.103.0.0            109          9592         25     100048
Dec 19 10:39:54 100.104.0.0            109          9592         25     100048
Dec 19 10:39:54 192.168.25.0             0             0          I     100064
Dec 19 10:39:54  Dec 19 10:39:54, read statistics for 5 FECs in 00:00:00 seconds
 (10 queries) for BGP group ext2 (Index 1) NLRI inet-labeled-unicast
```

Chapter 23

Configuration Examples for Interprovider and Carrier-of-Carriers VPNs

This chapter contains examples that illustrate how to configure interprovider and carrier-of-carriers virtual private networks (VPNs). It includes the following sections:

Example Terminology

The following terminology is used in these examples and is specific to Juniper Networks:

- bgp.l3vpn.0—The table on the provider edge (PE) router in which the VPN-IPv4 routes that are received from another PE router are stored. Incoming routes are checked against the vrf-import statements from all the VPNs configured on the PE router. If there is a match, the VPN–Internet Protocol version 4 (IPv4) route is added to the bgp.l3vpn.0 table. To view the bgp.l3vpn.0 table, issue the show route table bgp.l3vpn.0 command.
- *routing-instance-name*.inet.0—The routing table for a specific routing instance. For example, a routing instance called VPN-A has a routing table called VPN-A.inet.0. Routes are added to this table in the following ways:
  - They are sent from a customer edge (CE) router configured within the VPN-A routing instance.
  - They are advertised from a remote PE router that passes the vrf-import policy configured within VPN-A (to view the route, run the show route command). IPv4 (not VPN-IPv4) routes are stored in this table.
- vrf-import *policy-name*—An import policy configured on a particular routing instance on a PE router. This policy is required for the configuration of interprovider and carrier-of-carriers VPNs. It is applied to VPN-IPv4 routes learned from another PE router or a route reflector.
- vrf-export *policy-name*—An export policy configured on a particular routing instance on a PE router. It is required for the configuration of interprovider and carrier-of-carriers VPNs. It is applied to VPN-IPv4 routes (originally learned from

locally connected CE routers as IPv4 routes), which are advertised to another PE router or route reflector.

- MP-EBGP—The multiprotocol external BGP (MP-EBGP) mechanism is used to export VPN-IPv4 routes across an autonomous system (AS) boundary. To apply this mechanism, use the labeled-unicast statement at the [edit protocols bgp group *group-name* family inet] hierarchy level.

Interprovider VPN Examples

The following examples illustrate how to configure interprovider VPNs:

- Interprovider VPN Example—MP-EBGP Between ISP Peer Routers on page 486
- Interprovider VPN Example—Multihop MP-EBGP with P Routers on page 493

Interprovider VPN Example—MP-EBGP Between ISP Peer Routers

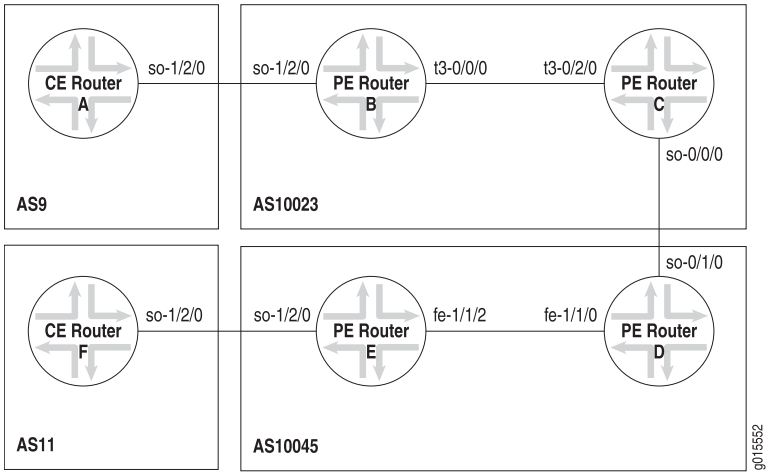
In this example, all routes learned from the CE routers are sent over both service provider networks as VPN-IPv4 routes. The routes are initially learned by the PE routers (Router B and Router E) from the CE routers (Router A and Router F) and are announced by the PE routers to the AS border routers (Router C and Router D). The AS border routers are then configured with an MP-EBGP session, enabling them to pass the VPN-IPv4 routes with each other. When an AS border router—Router C for example—learns VPN-IPv4 routes from an internal BGP (IBGP) PE, the following occurs:

1. Router C sets itself as the next hop for the route and creates a label for that route.
2. Router C advertises the VPN-IPv4 route to PE Router D in AS 10045.
3. Router D sets the next hop to itself, creates another label, and then forwards the label and the route to its IBGP PE router (Router E).

This example has scaling limitations because of restrictions on the number of labels each PE router needs to allocate at the AS border.

Figure 51 on page 487 illustrates the network topology used in this VPN example.

Figure 51: Network Topology of Interprovider VPN Example



Configuration for Router A

Configure a family inet EBGP session with Router B and export the direct routes:

```
[edit]
protocols {
   bgp {
      group to-provider {
         export attached;
         peer-as 10023;
         neighbor 192.168.198.2;
      }
   }
}
policy-options {
   policy-statement attached {
      from protocol direct;
      then accept;
   }
}
```

Configuration for Router B

Router A is configured as a CE router (using the routing-instances statement) in the configuration for Router B. Because they exchange VPN-IPv4 routes, Router D and Router C are configured as PE routers.

Configure Router B:

```
[edit]
protocols {
   rsvp {
      interface t3-0/0/0.0;
   }
   mpls {
      label-switched-path to-routerC {
```

```
            to 10.255.14.171;
            description "to-routerC for use with VPNs";
        }
        interface t3-0/0/0.0;
        interface so-1/2/0.0;
    }
    bgp {
        group to-ibgp {
            type internal;
            local-address 10.255.14.175;
            family inet-vpn {
                unicast;
            }
            neighbor 10.255.14.171;
        }
    }
    ospf {
        traffic-engineering;
        reference-bandwidth 4g;
        area 0.0.0.0 {
            interface t3-0/0/0.0;
            interface lo0.0 {
                passive;
            }
        }
    }
}
routing-instances {
    vpna {
        instance-type vrf;
        interface so-1/2/0.0;
        route-distinguisher 10.255.14.175:9;
        vrf-import vpna-import;
        vrf-export vpna-export;
        protocols {
            bgp {
                group to-ce {
                    peer-as 9;
                    neighbor 192.168.198.1;
                }
            }
        }
    }
}
policy-options {
    policy-statement vpna-import {
        term 1 {
            from {
                protocol bgp;
                community vpna-comm;
            }
            then accept;
        }
        term 2 {
            then reject;
        }
```

```
    }
    policy-statement vpna-export {
        term 1 {
            from protocol bgp;
            then {
                community add vpna-comm;
                accept;
            }
        }
        term 2 {
            then reject;
        }
    }
    community vpna-comm members target:100:1001;
}
```

Configuration for Router C

In the BGP protocol configuration for Router C, include the **keep all** statement. When this statement is included, BGP must store every route learned through BGP. Configure two BGP sessions (configure **family inet-vpn** on both sessions):

- IBGP session to Router B (group **to-ibgp** in this example)
- EBGP session to Router D (group **to-ebgp-pe** in this example)

Interface **t3-0/2/0** is added at the **[edit protocols mpls]** hierarchy level, allowing BGP to announce routes with labels over the EBGP session.

Configure Router C:

```
[edit]
protocols {
    rsvp {
        interface t3-0/2/0.0;
    }
    mpls {
        label-switched-path to-routerB {
            to 10.255.14.175;
            description "to-routerB for use with vpns";
        }
        interface t3-0/2/0.0;
        interface so-0/0/0.0;
    }
    bgp {
        keep all;
        group to-ibgp {
            type internal;
            local-address 10.255.14.171;
            family inet-vpn {
                unicast;
            }
            neighbor 10.255.14.175;
        }
        group to-ebgp-pe {
```

```
        type external;
        family inet-vpn {
          unicast;
        }
        neighbor 192.168.197.22 {
          peer-as 10045;
        }
      }
    }
    ospf {
      traffic-engineering;
      reference-bandwidth 4g;
      area 0.0.0.0 {
        interface t3-0/2/0.0;
        interface lo0.0 {
          passive;
        }
      }
    }
}
```

Configuration for Router D

The configuration for Router D is almost identical to that of Router C:

```
[edit]
protocols {
    rsvp {
      interface fe-1/1/0.0;
    }
    mpls {
      label-switched-path to-E {
        to 10.255.14.177;
        description "to-routerE for vpna";
      }
      interface fe-1/1/0.0;
      interface so-0/1/0.0;
    }
    bgp {
      keep all;
      group to-ibgp-pe {
        type internal;
        family inet-vpn {
          unicast;
        }
        neighbor 10.255.14.177;
      }
      group to-ebgp-pe {
        type external;
        family inet-vpn {
          unicast;
        }
        peer-as 10023;
        neighbor 192.168.197.21;
      }
```

```
    }
    ospf {
        traffic-engineering;
        reference-bandwidth 4g;
        area 0.0.0.0 {
            interface fe-1/1/0.0;
            interface lo0.0 {
                passive;
            }
        }
    }
}
```

Configuration for Router E

The configuration for Router E is very similar to the configuration for Router B:

```
[edit]
protocols {
    rsvp {
        interface fe-1/1/2.0;
    }
    mpls {
        label-switched-path to-routerD {
            to 10.255.14.173;
            description "to-routerD for use with VPNa";
        }
        interface fe-1/1/2.0;
        interface so-1/2/0.0;
    }
    bgp {
        group to-ibgp-pe {
            type internal;
            local-address 10.255.14.177;
            family inet-vpn {
                unicast;
            }
            neighbor 10.255.14.173;
        }
    }
    ospf {
        traffic-engineering;
        reference-bandwidth 4g;
        area 0.0.0.0 {
            interface fe-1/1/2.0;
            interface lo0.0 {
                passive;
            }
        }
    }
}
routing-instances {
    vpna {
        instance-type vrf;
        interface so-1/2/0.0;
```

```
        route-distinguisher 10.255.14.177:11;
        vrf-import vpna-import;
        vrf-export vpna-export;
        protocols {
            bgp {
                group to-routerF-ce {
                    neighbor 192.168.198.14 {
                        peer-as 11;
                    }
                }
            }
        }
    }
}
policy-options {
    policy-statement vpna-import {
        term 1 {
            from {
                protocol bgp;
                community vpna-comm;
            }
            then accept;
        }
        term 2 {
            then reject;
        }
    }
    policy-statement vpna-export {
        term 1 {
            from protocol bgp;
            then {
                community add vpna-comm;
                accept;
            }
        }
        term 2 {
            then reject;
        }
    }
    community vpna-comm members target:100:1001;
}
```

Configuration for Router F

Configure Router F as a CE router; the configuration is similar to that for Router A:

```
[edit]
protocols {
    bgp {
        group to-provider {
            type external;
            export attached;
            neighbor 192.168.198.13 {
                peer-as 10045;
            }
```

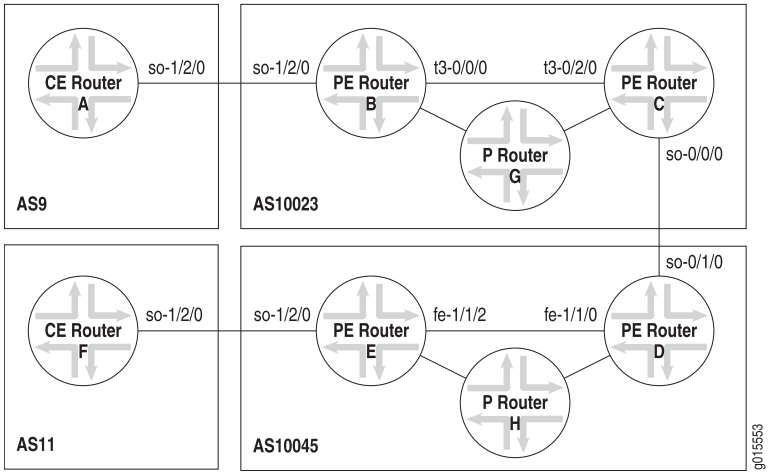
```
        }
      }
    }
    policy-options {
      policy-statement attached {
        from protocol direct;
        then accept;
      }
    }

}
```

Interprovider VPN Example—Multihop MP-EBGP with P Routers

In this example, labeled IPv4 (not VPN-IPv4), routes are exchanged by the AS border routers (Router C and Router D) to provide Multiprotocol Label Switching (MPLS) connectivity between the PE routers. Router G and H are provider routers.

Figure 52 on page 493 illustrates the network topology used in this VPN example.

Figure 52: Network Topology of Interprovider VPN Example—Multihop MP-EBGP



Only routes internal to the service provider networks should be announced between Router C and Router D. Configure this by including the family inet labeled-unicast statement in the IBGP and EBGP configuration on the PE routers. When you set family inet labeled-unicast, the local router announces internal routes from inet.0 in the following manner:

- If a label exists for the route, the local router creates a label, performs a swap, and announces the route from inet.0 with the label.
- If a label does not exist for the route, the local router creates a label, performs a pop, and announces the route from inet.0 with the label.

Routes learned from the labeled-unicast session are placed into the inet.0 routing table.

In addition, you configure a multihop MP-EBGP session between the end PE routers (Router B and Router E). This additional MP-EBGP session allows the announcement of VPN-IPv4 routes, and allows you to maintain VPN connectivity while keeping VPN-IPv4 routes out of the core of the network.

For configuration information, see the following sections:

- Configuration for Router A on page 494
- Configuration for Router B on page 494
- Configuration for Router C on page 496
- Configuration for Router D on page 497
- Configuration for Router E on page 498
- Configuration for Router F on page 500

Configuration for Router A

The configuration for Router A in this example is identical to the configuration for Router A in the section "Interprovider VPN Example—MP-EBGP Between ISP Peer Routers" on page 486. See "Configuration for Router A" on page 487.

Configuration for Router B

Router A is configured as a CE router (using the routing-instances statement) in the configuration for Router B. Because they exchange VPN-IPv4 routes, Router C and Router D are configured as PE routers.

In the BGP group to-ibgp, include the family inet labeled-unicast statement to pass labeled IPv4 routes, and configure an EBGP multihop session to pass VPN-IPv4 routes:

```
[edit]
protocols {
   bgp {
      group to-ibgp {
         type internal;
         local-address 10.255.14.175;
         family inet {
            labeled-unicast {
               resolve-vpn;
            }
         }
         neighbor 10.255.14.171;
      }
      group to-remote-pe {
         multihop {
            ttl 10;
         }
         family inet-vpn {
            unicast;
         }
         neighbor 10.255.14.177 {
            peer-as 10045;
         }
```

```
    }
    mpls {
        label-switched-path to-routerC {
            to 10.255.14.171;
            description "to-routerC for use with VPNs";
        }
        interface t3-0/0/0.0;
        interface so-1/2/0.0;
    }
    ospf {
        traffic-engineering;
        reference-bandwidth 4g;
        area 0.0.0.0 {
            interface t3-0/0/0.0;
            interface lo0.0 {
                passive;
            }
        }
    }
    rsvp {
        interface t3-0/0/0.0;
    }
}
routing-instances {
    vpna {
        instance-type vrf;
        interface so-1/2/0.0;
        route-distinguisher 10.255.14.175:9;
        vrf-import vpna-import;
        vrf-export vpna-export;
        protocols {
            bgp {
                group to-ce {
                    peer-as 9;
                    neighbor 192.168.198.1;
                }
            }
        }
    }
}
policy-options {
    policy-statement vpna-import {
        term 1 {
            from {
                protocol bgp;
                community vpna-comm;
            }
            then accept;
        }
        term 2 {
            then reject;
        }
    }
    policy-statement vpna-export {
        term 1 {
            from protocol bgp;
```

```
            then {
                community add vpna-comm;
                accept;
            }
        }
        term 2 {
            then reject;
        }
    }
    community vpna-comm members target:100:1001;
  }
}
```

Configuration for Router C

Configure two BGP sessions (configure family inet-vpn on both sessions):

- IBGP session to Router B (group to-ibgp in this example)
- EBGP session to Router D (group to-ebgp-pe in this example)

Interface t3-0/2/0 is added at the [edit protocols mpls] hierarchy level, allowing BGP to announce routes with labels over the EBGP session.

Configure Router C:

```
[edit]
protocols {
    bgp {
        group to-ibgp {
            type internal;
            local-address 10.255.14.171;
            family inet {
                labeled-unicast;
            }
            neighbor 10.255.14.175;
        }
        group to-ebgp-pe {
            type external;
            family inet {
                labeled-unicast;
            }
            export internal;
            neighbor 192.168.197.22 {
                peer-as 10045;
            }
        }
        mpls {
            label-switched-path to-routerB {
                to 10.255.14.175;
                description "to-routerB for use with vpns";
            }
            interface t3-0/2/0.0;
            interface so-0/0/0.0;
            traffic-engineering bgp-igp;
```

```
        }
        rsvp {
            interface t3-0/2/0.0;
        }
        ospf {
            traffic-engineering;
            reference-bandwidth 4g;
            area 0.0.0.0 {
                interface t3-0/2/0.0;
                interface lo0.0 {
                    passive;
                }
            }
        }
    }
    policy-options {
        policy-statement internal {
            term 1 {
                from protocol [ospf direct ldp];
                then accept;
            }
            term 2 {
                then reject;
            }
        }
    }
}
```

Configuration for Router D

Configure Router D:

```
[edit]
protocols {
    bgp {
        group to-ibgp-pe {
            type internal;
            family inet {
                labeled-unicast;
            }
            neighbor 10.255.14.177;
        }
        group to-ebgp-pe {
            type external;
            family inet {
                labeled-unicast;
            }
            export internal;
            peer-as 10023;
            neighbor 192.168.197.21;
        }
        mpls {
            label-switched-path to-E {
                to 10.255.14.177;
                description "to-routerE for vpna";
```

```
            }
            interface fe-1/1/0.0;
            interface so-0/1/0.0;
            traffic-engineering bgp-igp;
        }
        ospf {
            traffic-engineering;
            reference-bandwidth 4g;
            area 0.0.0.0 {
                interface fe-1/1/0.0;
                interface lo0.0 {
                    passive;
                }
            }
        }
        rsvp {
            interface fe-1/1/0.0;
        }
    }
    policy-options {
        policy-statement internal {
            term 1 {
                from protocol [ospf direct ldp];
                then accept;
            }
            term 2 {
                then reject;
            }
        }
    }
}
```

Configuration for Router E

The configuration for Router E is very similar to the configuration for Router B:

```
[edit]
protocols {
    bgp {
        group to-ibgp-pe {
            type internal;
            local-address 10.255.14.177;
            family inet {
                labeled-unicast;
            }
            neighbor 10.255.14.173;
        }
        group to-remote-pe {
            multihop {
                ttl 10;
            }
            family inet-vpn {
                unicast;
            }
            neighbor 10.255.14.175 {
```

```
            peer-as 10023;
        }
    }
    mpls {
        label-switched-path to-routerD {
            to 10.255.14.173;
            description "to-routerD for use with VPNa";
        }
        interface fe-1/1/2.0;
        interface so-1/2/0.0;
    }
    ospf {
        traffic-engineering;
        reference-bandwidth 4g;
        area 0.0.0.0 {
            interface fe-1/1/2.0;
            interface lo0.0 {
                passive;
            }
        }
    }
    rsvp {
        interface fe-1/1/2.0;
    }
}
routing-instances {
    vpna {
        instance-type vrf;
        interface so-1/2/0.0;
        route-distinguisher 10.255.14.177:11;
        vrf-import vpna-import;
        vrf-export vpna-export;
        protocols {
            bgp {
                group to-routerF-ce {
                    neighbor 192.168.198.14 {
                        peer-as 11;
                    }
                }
            }
        }
    }
}
policy-options {
    policy-statement vpna-import {
        term 1 {
            from {
                protocol bgp;
                community vpna-comm;
            }
            then accept;
        }
        term 2 {
            then reject;
        }
    }
```

```
        policy-statement vpna-export {
            term 1 {
                from protocol bgp;
                then {
                    community add vpna-comm;
                    accept;
                }
            }
            term 2 {
                then reject;
            }
        }
        community vpna-comm members target:100:1001;
    }
}
```

Configuration for Router F

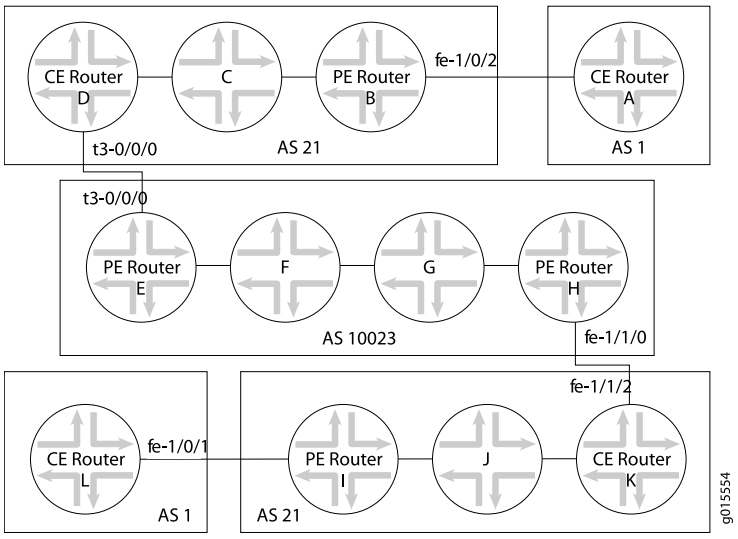
The configuration for Router F in this example is identical to the configuration for Router F in the section "Interprovider VPN Example—MP-EBGP Between ISP Peer Routers" on page 486. See "Configuration for Router F" on page 492.

Carrier-of-Carriers VPN Examples

A carrier-of-carriers service allows an Internet service provider (ISP) to connect to a transparent outsourced backbone at multiple locations.

Figure 53 on page 500 shows the network topology in both carrier-of-carriers examples.

Figure 53: Carrier-of-Carriers VPN Example Network Topology



There are two variations of this example:

- Carrier-of-Carriers VPN Example—Customer Provides Internet Service on page 501
- Carrier-of-Carriers VPN Example—Customer Provides VPN Service on page 510

Carrier-of-Carriers VPN Example—Customer Provides Internet Service

In this example, the carrier customer is not required to configure MPLS and Label Distribution Protocol (LDP) on its network. However, the carrier provider must configure MPLS and LDP on its network.

Configuration for Router A

In this example, Router A represents an end customer. You configure this router as a CE device.

```
[edit]
protocols {
    bgp {
        group to-routerB {
            export attached;
            peer-as 21;
            neighbor 192.168.197.169;
        }
    }
}
policy-options {
    policy-statement attached {
        from protocol direct;
        then accept;
    }
}
```

Configuration for Router B

Router B can act as the gateway router, responsible for aggregating end customers and connecting them to the network. If a full-mesh IBGP session is configured, you can use route reflectors.

```
[edit]
protocols {
    bgp {
        group int {
            type internal;
            local-address 10.255.14.179;
            neighbor 10.255.14.175;
            neighbor 10.255.14.181;
            neighbor 10.255.14.176;
            neighbor 10.255.14.178;
            neighbor 10.255.14.177;
        }
        group to-vpn-blue {
            peer-as 1;
            neighbor 192.168.197.170;
```

```
        }
    }
    ospf {
        area 0.0.0.0 {
            interface lo0.0 {
                passive;
            }
            interface fe-1/0/3.0;
            interface fe-1/0/2.0 {
                passive;
            }
        }
    }
}
```

Configuration for Router C

Configure Router C:

```
[edit]
protocols {
    bgp {
        group int {
            type internal;
            local-address 10.255.14.176;
            neighbor 10.255.14.179;
            neighbor 10.255.14.175;
            neighbor 10.255.14.177;
            neighbor 10.255.14.178;
            neighbor 10.255.14.181;
        }
    }
    ospf {
        area 0.0.0.0 {
            interface lo0.0 {
                passive;
            }
            interface fe-0/3/3.0;
            interface fe-0/3/0.0;
        }
    }
}
```

Configuration for Router D

Router D is the CE router with respect to AS 10023. In a carrier-of-carriers VPN, the CE router must be able to send labels to the carrier provider; this is done with the labeled-unicast statement in group to-isp-red.

```
[edit]
protocols {
    mpls {
        interface t3-0/0/0.0;
    }
    bgp {
```

```
        group int {
            type internal;
            local-address 10.255.14.175;
            neighbor 10.255.14.179;
            neighbor 10.255.14.176;
            neighbor 10.255.14.177;
            neighbor 10.255.14.178;
            neighbor 10.255.14.181;
        }
        group to-isp-red {
            export internal;
            peer-as 10023;
            neighbor 192.168.197.13 {
                family inet {
                    labeled-unicast;
                }
            }
        }
    }
    ospf {
        area 0.0.0.0 {
            interface lo0.0 {
                passive;
            }
            interface fe-0/3/0.0;
            interface t3-0/0/0.0 {
                passive;
            }
        }
    }
}
policy options {
    policy-statement internal {
        term a {
            from protocol [ ospf direct ];
            then accept;
        }
        term b {
            then reject;
        }
    }
}
```

Configuration for Router E

This configuration sets up the inet-vpn IBGP session with Router H and the PE router portion of the VPN with Router D. Because Router D is required to send labels in this example, configure the BGP session with the labeled-unicast statement within the VPN routing and forwarding (VRF) table.

```
[edit]
protocols {
    mpls {
        interface t3-0/2/0.0;
        interface at-0/1/0.0;
```

```
    }
    bgp {
        group pe-pe {
            type internal;
            local-address 10.255.14.171;
            family inet-vpn {
                any;
            }
            neighbor 10.255.14.173;
        }
    }
    isis {
        interface at-0/1/0.0;
        interface lo0.0 {
            passive;
        }
    }
    ldp {
        interface at-0/1/0.0;
    }
}
routing-instances {
    vpn-isp1 {
        instance-type vrf;
        interface t3-0/2/0.0;
        route-distinguisher 10.255.14.171:21;
        vrf-import vpn-isp1-import;
        vrf-export vpn-isp1-export;
        protocols {
            bgp {
                group to-isp1 {
                    peer-as 21;
                    neighbor 192.168.197.14 {
                        family inet {
                            labeled-unicast;
                        }
                    }
                }
            }
        }
    }
}
policy-options {
    policy-statement vpn-isp1-import {
        term a {
            from {
                protocol bgp;
                community vpn-isp1-comm;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    policy-statement vpn-isp1-export {
```

```
        term a {
            from protocol bgp;
            then {
                community add vpn-isp1-comm;
                accept;
            }
        }
        term b {
            then reject;
        }
    }
    community vpn-isp1-comm members target:69:21;
}
```

Configuration for Router F

Configure Router F to act as a label-swapping router:

```
[edit]
protocols {
    isis {
        interface so-0/2/0.0;
        interface at-0/3/0.0;
        interface lo0.0 {
            passive;
        }
    }
    ldp {
        interface so-0/2/0.0;
        interface at-0/3/0.0;
    }
}
```

Configuration for Router G

Configure Router G to act as a label-swapping router:

```
[edit]
protocols {
    isis {
        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;
    }
}
```

Configuration for Router H

Router H acts as the PE router for AS 10023. The configuration that follows is similar to that for Router F:

```
[edit]
protocols {
    mpls {
        interface fe-1/1/0.0;
        interface so-1/0/0.0;
    }
    bgp {
        group pe-pe {
            type internal;
            local-address 10.255.14.173;
            family inet-vpn {
                any;
            }
            neighbor 10.255.14.171;
        }
    }
    isis {
        interface so-1/0/0.0;
        interface lo0.0 {
            passive;
        }
    }
    ldp {
        interface so-1/0/0.0;
    }
}
routing-instances {
    vpn-isp1 {
        instance-type vrf;
        interface fe-1/1/0.0;
        route-distinguisher 10.255.14.173:21;
        vrf-import vpn-isp1-import;
        vrf-export vpn-isp1-export;
        protocols {
            bgp {
                group to-isp1 {
                    peer-as 21;
                    neighbor 192.168.197.94 {
                        family inet {
                            labeled-unicast;
                        }
                    }
                }
            }
        }
    }
}
policy-options {
    policy-statement vpn-isp1-import {
        term a {
            from {
```

```
            protocol bgp;
            community vpn-isp1-comm;
          }
          then accept;
        }
        term b {
          then reject;
        }
      }
      policy-statement vpn-isp1-export {
        term a {
          from protocol bgp;
          then {
            community add vpn-isp1-comm;
            accept;
          }
        }
        term b {
          then reject;
        }
      }
      community vpn-isp1-comm members target:69:21;
}
```

Configuration for Router I

Configure Router I to connect to the basic Internet service customer (Router L):

```
[edit]
protocols {
    mpls {
        interface fe-1/0/1.0;
        interface fe-1/1/3.0;
    }
    bgp {
        group int {
            type internal;
            local-address 10.255.14.181;
            neighbor 10.255.14.177;
            neighbor 10.255.14.179;
            neighbor 10.255.14.175;
            neighbor 10.255.14.176;
            neighbor 10.255.14.178;
        }
        group to-vpn-green {
            peer-as 1;
            neighbor 192.168.197.198;
        }
    }
    ospf {
        area 0.0.0.0 {
            interface lo0.0 {
                passive;
            }
            interface fe-1/0/1.0 {
```

```
            passive;
        }
        interface fe-1/1/3.0;
    }
  }
}
```

Configuration for Router J

Configure Router J as a label-swapping router:

```
[edit]
protocols {
  bgp {
    group int {
      type internal;
      local-address 10.255.14.178;
      neighbor 10.255.14.177;
      neighbor 10.255.14.181;
      neighbor 10.255.14.175;
      neighbor 10.255.14.176;
      neighbor 10.255.14.179;
    }
  }
}
ospf {
  area 0.0.0.0 {
    interface lo0.0 {
      passive;
    }
    interface fe-1/0/2.0;
    interface fe-1/0/3.0;
  }
}
```

Configuration for Router K

Router K acts as the CE router at the end of the connection to the carrier provider. As in the configuration for Router D, include the labeled-unicast statement for the EBGP session:

```
[edit]
protocols {
  mpls {
    interface fe-1/1/2.0;
    interface fe-1/0/2.0;
  }
  bgp {
    group int {
      type internal;
      local-address 10.255.14.177;
      neighbor 10.255.14.181;
      neighbor 10.255.14.178;
      neighbor 10.255.14.175;
      neighbor 10.255.14.176;
```

```
        neighbor 10.255.14.179;
      }
      group to-isp-red {
        export internal;
        peer-as 10023;
        neighbor 192.168.197.93 {
          family inet {
            labeled-unicast;
          }
        }
      }
    }
    ospf {
      area 0.0.0.0 {
        interface lo0.0 {
          passive;
        }
        interface fe-1/0/2.0;
        interface fe-1/1/2.0 {
          passive;
        }
      }
    }
}
policy-options {
    policy-statement internal {
      term a {
        from protocol [ ospf direct ];
        then accept;
      }
      term b {
        then reject;
      }
    }
}
```

Configuration for Router L

Configure Router L to act as the end customer for the carrier-of-carriers VPN service:

```
[edit]
protocols {
    bgp {
      group to-routerl {
        export attached;
        peer-as 21;
        neighbor 192.168.197.197;
      }
    }
}
policy-options {
    policy-statement attached {
      from protocol direct;
      then accept;
    }
```

```
}
```

Carrier-of-Carriers VPN Example—Customer Provides VPN Service

In this example, the carrier customer *must* run some form of MPLS (Resource Reservation Protocol [RSVP] or LDP) on its network to provide VPN services to the end customer. In the example below, Router B and Router I act as PE routers, and a functioning MPLS path is required between these routers if they exchange VPN-IPv4 routes.

Configuration for Router A

In this example, Router A acts as the CE router for the end customer. Configure a default family inet BGP session on Router A:

```
[edit]
protocols {
   bgp {
      group to-routerB {
         export attached;
         peer-as 21;
         neighbor 192.168.197.169;
      }
   }
}
policy-options {
   policy-statement attached {
      from protocol direct;
      then accept;
   }
}
```

Configuration for Router B

Because Router B is the PE router for the end customer CE router (Router A), you need to configure a routing instance (vpna). Configure the labeled-unicast statement on the IBGP session to Router D, and configure family-inet-vpn for the IBGP session to the other side of the network (see Figure 53 on page 500) with Router I:

```
[edit]
protocols {
   mpls {
      interface fe-1/0/2.0;
      interface fe-1/0/3.0;
   }
   bgp {
      group int {
         type internal;
         local-address 10.255.14.179;
         neighbor 10.255.14.175 {
            family inet {
               labeled-unicast;
               resolve-vpn;
            }
```

```
        }
      }
      neighbor 10.255.14.181 {
        family inet-vpn {
          any;
        }
      }
    }
    ospf {
      area 0.0.0.0 {
        interface lo0.0 {
          passive;
        }
        interface fe-1/0/3.0;
      }
    }
    ldp {
      interface fe-1/0/3.0;
    }
}
routing-instances {
    vpna {
      instance-type vrf;
      interface fe-1/0/2.0;
      route-distinguisher 10.255.14.179:21;
      vrf-import vpna-import;
      vrf-export vpna-export;
      protocols {
        bgp {
          group vpna-06 {
            peer-as 1;
            neighbor 192.168.197.170;
          }
        }
      }
    }
}
policy-options {
    policy-statement vpna-import {
      term a {
        from {
          protocol bgp;
          community vpna-comm;
        }
        then accept;
      }
      term b {
        then reject;
      }
    }
    policy-statement vpna-export {
      term a {
        from protocol bgp;
        then {
          community add vpna-comm;
          accept;
```

```
            }
        }
        term b {
            then reject;
        }
    }
    community vpna-comm members target:100:1001;
}
```

Configuration for Router C

Configure Router C as a label-swapping router within the local AS:

```
[edit]
protocols {
    mpls {
        traffic-engineering bgp-igp;
    }
    ospf {
        area 0.0.0.0 {
            interface lo0.0 {
                passive;
            }
            interface fe-0/3/3.0;
            interface fe-0/3/0.0;
        }
    }
    ldp {
        interface fe-0/3/0.0;
        interface fe-0/3/3.0;
    }
}
```

Configuration for Router D

Router D acts as the CE router for the VPN services provided by the AS 10023 network. In the BGP group configuration for group int, which handles traffic to Router B (10.255.14.179), you include the labeled-unicast statement. You also need to configure the BGP group to-isp-red to send labeled internal routes to the PE router (Router E).

```
[edit]
protocols {
    mpls {
        traffic-engineering bgp-igp;
        interface fe-0/3/0.0;
        interface t3-0/0/0.0;
    }
    bgp {
        group int {
            type internal;
            local-address 10.255.14.175;
            neighbor 10.255.14.179 {
                family inet {
                    labeled-unicast;
```

```
            }
          }
        }
        group to-isp-red {
          export internal;
          peer-as 10023;
          neighbor 192.168.197.13 {
            family inet {
              labeled-unicast;
            }
          }
        }
      }
      ospf {
        area 0.0.0.0 {
          interface lo0.0 {
            passive;
          }
          interface fe-0/3/0.0;
        }
      }
      ldp {
        interface fe-0/3/0.0;
      }
    }
    policy-options {
      policy-statement internal {
        term a {
          from protocol [ ospf direct ];
          then accept;
        }
        term b {
          then reject;
        }
      }
    }
```

Configuration for Router E

Router E and Router H are PE routers. Configure a PE-router-to-PE-router BGP session to allow VPN-IPv4 routes to pass between these two PE routers. Configure the routing instance on Router E to send labeled routes to the CE router (Router D).

Configure Router E:

```
[edit]
protocols {
  mpls {
    interface t3-0/2/0.0;
    interface at-0/1/0.0;
  }
  bgp {
    group pe-pe {
      type internal;
      local-address 10.255.14.171;
```

```
            family inet-vpn {
                any;
            }
            neighbor 10.255.14.173;
        }
    }
    isis {
        interface at-0/1/0.0;
        interface lo0.0 {
            passive;
        }
    }
    ldp {
        interface at-0/1/0.0;
    }
}
policy-options {
    policy-statement vpn-isp1-import {
        term a {
            from {
                protocol bgp;
                community vpn-isp1-comm;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    policy-statement vpn-isp1-export {
        term a {
            from protocol bgp;
            then {
                community add vpn-isp1-comm;
                accept;
            }
        }
        term b {
            then reject;
        }
    }
    community vpn-isp1-comm members target:69:21;
}
routing-instances {
    vpn-isp1 {
        instance-type vrf;
        interface t3-0/2/0.0;
        route-distinguisher 10.255.14.171:21;
        vrf-import vpn-isp1-import;
        vrf-export vpn-isp1-export;
        protocols {
            bgp {
                group to-isp1 {
                    peer-as 21;
                    neighbor 192.168.197.14 {
                        family inet {
```

```
                labeled-unicast;
            }
        }
    }
  }
}
```

Configuration for Router F

Configure Router F to swap labels for routes running through its interfaces:

```
[edit]
protocols {
   isis {
      interface so-0/2/0.0;
      interface at-0/3/0.0;
      interface lo0.0 {
         passive;
      }
   }
   ldp {
      interface so-0/2/0.0;
      interface at-0/3/0.0;
   }
}
```

Configuration for Router G

Configure Router G:

```
[edit]
protocols {
   isis {
      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;
   }
}
```

Configuration for Router H

The configuration for Router H is similar to the configuration for Router E:

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

```
        interface fe-1/1/0.0;
        interface so-1/0/0.0;
    }
    bgp {
        group pe-pe {
            type internal;
            local-address 10.255.14.173;
            family inet-vpn {
                any;
            }
            neighbor 10.255.14.171;
        }
    }
    isis {
        interface so-1/0/0.0;
        interface lo0.0 {
            passive;
        }
    }
    ldp {
        interface so-1/0/0.0;
    }
}
routing-instances {
    vpn-isp1 {
        instance-type vrf;
        interface fe-1/1/0.0;
        route-distinguisher 10.255.14.173:21;
        vrf-import vpn-isp1-import;
        vrf-export vpn-isp1-export;
        protocols {
            bgp {
                group to-isp1 {
                    peer-as 21;
                    neighbor 192.168.197.94 {
                        family inet {
                            labeled-unicast;
                        }
                    }
                }
            }
        }
    }
}
policy-options {
    policy-statement vpn-isp1-import {
        term a {
            from {
                protocol bgp;
                community vpn-isp1-comm;
            }
            then accept;
        }
        term b {
            then reject;
        }
```

```
    }
    policy-statement vpn-isp1-export {
        term a {
            from protocol bgp;
            then {
                community add vpn-isp1-comm;
                accept;
            }
        }
        term b {
            then reject;
        }
    }
    community vpn-isp1-comm members target:69:21;
}
```

Configuration for Router I

Router I acts as the PE router for the end customer. The configuration that follows is similar to the configuration for Router B:

```
[edit]
protocols {
    mpls {
        interface fe-1/0/1.0;
        interface fe-1/1/3.0;
    }
    bgp {
        group int {
            type internal;
            local-address 10.255.14.181;
            neighbor 10.255.14.177 {
                family inet {
                    labeled-unicast {
                        resolve-vpn;
                    }
                }
            }
            neighbor 10.255.14.179 {
                family inet-vpn {
                    any;
                }
            }
        }
    }
    ospf {
        area 0.0.0.0 {
            interface lo0.0 {
                passive;
            }
            interface fe-1/1/3.0;
        }
    }
    ldp {
        interface fe-1/1/3.0;
```

```
    }
  }
  routing-instances {
    vpna {
      instance-type vrf;
      interface fe-1/0/1.0;
      route-distinguisher 10.255.14.181:21;
      vrf-import vpna-import;
      vrf-export vpna-export;
      protocols {
        bgp {
          group vpna-0 {
            peer-as 1;
            neighbor 192.168.197.198;
          }
        }
      }
    }
  }
  policy-options {
    policy-statement vpna-import {
      term a {
        from {
          protocol bgp;
          community vpna-comm;
        }
        then accept;
      }
      term b {
        then reject;
      }
    }
    policy-statement vpna-export {
      term a {
        from protocol bgp;
        then {
          community add vpna-comm;
          accept;
        }
      }
      term b {
        then reject;
      }
    }
    community vpna-comm members target:100:1001;
  }
```

Configuration for Router J

Configure Router J to swap labels for routes running through its interfaces:

```
[edit]
protocols {
  mpls {
    traffic-engineering bgp-igp;
```

```
    }
    ospf {
        area 0.0.0.0 {
            interface lo0.0 {
                passive;
            }
            interface fe-1/0/2.0;
            interface fe-1/0/3.0;
        }
    }
    ldp {
        interface fe-1/0/2.0;
        interface fe-1/0/3.0;
    }
}
```

Configuration for Router K

The configuration for Router K is similar to the configuration for Router D:

```
[edit]
protocols {
    mpls {
        traffic-engineering bgp-igp;
        interface fe-1/1/2.0;
        interface fe-1/0/2.0;
    }
    bgp {
        group int {
            type internal;
            local-address 10.255.14.177;
            neighbor 10.255.14.181 {
                family inet {
                    labeled-unicast;
                }
            }
        }
        group to-isp-red {
            export internal;
            peer-as 10023;
            neighbor 192.168.197.93 {
                family inet {
                    labeled-unicast;
                }
            }
        }
    }
    ospf {
        area 0.0.0.0 {
            interface lo0.0 {
                passive;
            }
            interface fe-1/0/2.0;
        }
    }
```

```
    ldp {
        interface fe-1/0/2.0;
    }
}
policy-options {
    policy-statement internal {
        term a {
            from protocol [ ospf direct ];
            then accept;
        }
        term b {
            then reject;
        }
    }
}
```

Configuration for Router L

In this example, Router L is the end customer's CE router. Configure Router L:

```
[edit]
protocols {
    bgp {
        group to-l {
            export attached;
            peer-as 21;
            neighbor 192.168.197.197;
        }
    }
}
policy-options {
    policy-statement attached {
        from protocol direct;
        then accept;
    }
}
```

Multiple Instances for LDP and Carrier-of-Carriers VPNs

By configuring multiple LDP routing instances, you can use LDP to advertise labels in a carrier-of-carriers VPN from a core provider PE router to a customer carrier CE router. Having LDP advertise labels in this manner is especially useful when the carrier customer is a basic 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 3 VPN or Layer 2 VPN services to its customers.

For an example of how to configure multiple LDP routing instances for carrier-of-carriers VPNs, see the *JUNOS Feature Guide* on the product documentation page of the Juniper Networks Web site, located at http://www.juniper.net/.

Chapter 24

Summary of the Interprovider and Carrier-of-Carriers VPNs Configuration Statements

The following section explains the configuration statements that apply specifically to hierarchical and recursive BGP and Multiprotocol Label Switching (MPLS) virtual private networks (VPNs).

labeled-unicast

Syntax

```
labeled-unicast {
    per-group-label;
    resolve-vpn;
    traffic-statistics {
        file file-name <world-readable | no-world-readable>;
        interval seconds;
    }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols bgp family inet],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* family inet],
[edit protocols bgp family inet],
[edit protocols bgp group *group-name* family inet]

Release Information

Statement introduced before JUNOS Release 7.4.

Description

Advertise labeled routes from the inet.0 VPN, and place labeled routes into the inet.0 VPN. When the labeled-unicast statement is used, the local router automatically performs a next hop to self on all routes advertised into the external BGP (EBGP) from the internal BGP (IBGP) and from IBGP to EBGP.

Options

resolve-vpn—(Optional) Store labeled routes in the inet.3 routing table to resolve routes for a provider edge (PE) router located in a different autonomous system (AS). For a PE router to install a route in the VPN routing and forwarding (VRF) table, the next hop must resolve to a route stored in the inet.3 routing table. This option is also used to configure inter-AS VPLS with MAC operations.

The other statements are explained separately.

Usage Guidelines

See “Configuring Interprovider and Carrier-of-Carriers VPNs” on page 465 and “Configuring Inter-AS VPLS with MAC Processing at the ASBR” on page 429.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

per-group-label

Syntax per-group-label;

Hierarchy Level [edit logical-systems *logical-system-name* protocols bgp family inet labeled-unicast],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* family inet labeled-unicast],
[edit protocols bgp family inet labeled-unicast],
[edit protocols bgp group *group-name* family inet labeled-unicast]

Release Information Statement introduced before JUNOS Release 7.4.

Description Account for traffic from each customer separately by advertising separate labels for the same prefix to the peer routers in the BGP groups.

Usage Guidelines See "Configuring BGP to Gather Interprovider and Carrier-of-Carriers VPNs Statistics" on page 483.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

traffic-statistics

Syntax
```
traffic-statistics {
    file filename <world-readable | no-world-readable>;
    interval seconds;
}
```

Hierarchy Level [edit logical-systems *logical-system-name* protocols bgp family inet labeled-unicast],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* family inet labeled-unicast],
[edit protocols bgp family inet labeled-unicast],
[edit protocols bgp group *group-name* family inet labeled-unicast]

Release Information Statement introduced before JUNOS Release 7.4.

Description Enable the collection of traffic statistics for interprovider or carrier-of-carriers VPNs.

Options file *filename*—Specify a filename for the BGP labeled–unicast traffic statistics file. If you do not specify a filename, statistics are still collected but can only be viewed by using the show bgp group traffic statistics *group-name* command.

interval *seconds*—Specify how often BGP labeled-unicast traffic statistics are collected.

Usage Guidelines See "Configuring BGP to Gather Interprovider and Carrier-of-Carriers VPNs Statistics" on page 483.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Part 7
Layer 2 Circuits

Chapter 25

Layer 2 Circuit Overview

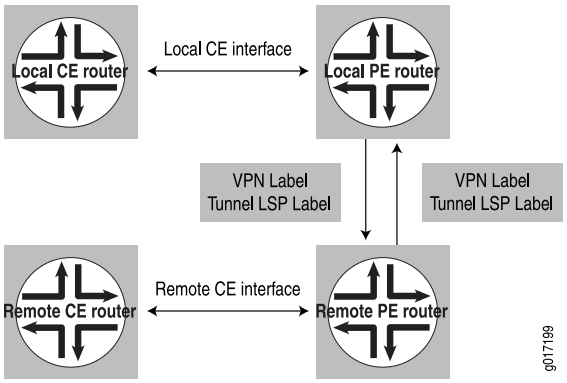
A Layer 2 circuit is a point-to-point Layer 2 connection transported by means of Multiprotocol Label Switching (MPLS) or another tunneling technology on the service provider's network. A Layer 2 circuit is similar to a circuit cross-connect (CCC), except that multiple Layer 2 circuits can be transported over a single label-switched path (LSP) tunnel between two provider edge (PE) routers. In contrast, each CCC requires a dedicated LSP.

This chapter discusses the following topics:

Layer 2 Circuit Overview

The JUNOS software implementation of Layer 2 circuits supports only the remote form of a Layer 2 circuit; that is, a connection from a local customer edge (CE) router to a remote CE router. Figure 54 on page 527 illustrates the components of a Layer 2 circuit.

Figure 54: Components of a Layer 2 Circuit



The interfaces shown in Figure 54 on page 527 are logical interfaces. Packets are sent to the remote CE router by means of an egress virtual private network (VPN) label advertised by the remote PE router. The VPN label transits over either a Resource Reservation Protocol (RSVP) or a Label Distribution Protocol (LDP) LSP (or other type) tunnel to the remote PE router connected to the remote CE router. If you configure RSVP for Layer 2 circuits, you must also configure LDP.

Return traffic sent from the remote CE router to the local CE router uses an ingress VPN label advertised by the local PE router, which again transits over an RSVP and LDP LSP to the local PE router from the remote PE router. LDP is the signaling protocol used for advertising VPN labels.

Layer 2 Circuit Standards

The JUNOS software substantially supports the following Layer 2 circuit standards:

- Internet draft draft-martini-l2circuit-encap-mpls-07.txt, *Encapsulation Methods for Transport of Layer 2 Frames Over IP and MPLS Networks*

The JUNOS software has the following exceptions:

- A packet with a sequence number of 0 is treated as out of sequence.
  - Any packet that does not have the next incremental sequence number is considered out of sequence.
  - When out-of-sequence packets arrive, the expected sequence number for the neighbor is set to the sequence number in the Layer 2 circuit control word.
- Internet draft draft-martini-l2circuit-trans-mpls-14.txt, *Transport of Layer 2 Frames Over MPLS*.

These drafts are available on the IETF Web site at http://www.ietf.org/.

Layer 2 Circuit Policy

You can configure JUNOS software routing policies to control the flow of packets over Layer 2 circuits. This capability allows you to provide different levels of service over a set of equal-cost Layer 2 circuits. For example, you can configure a circuit for high-priority traffic, a circuit for average-priority traffic, and a circuit for low-priority traffic. By configuring Layer 2 circuit policies, you can ensure that higher-value traffic has a greater likelihood of reaching its destination.

Layer 2 Circuit Bandwidth Accounting and Call Admission Control

The sections that follow discuss Layer 2 circuit bandwidth accounting and call admission control (CAC):

Bandwidth Accounting and Call Admission Control Overview

Some network environments require that a certain level of service be guaranteed across the entire length of a path transiting a service provider's network. For Layer 2 circuits transiting an MPLS core network, a customer requirement might be to assure that guarantees for bandwidth and class of service (CoS) be maintained across the core network. For example, an Asynchronous Transfer Mode (ATM) circuit can provide service guarantees for each traffic class. A Layer 2 circuit configured to transport that ATM circuit across the network could be expected to provide the same service guarantees.

Providing this type of service guarantee requires the following:

- The LSPs in the MPLS core network must be able to provide service guarantees for bandwidth, rerouting, and route failures. You accomplish these guarantees by configuring multiclass LSPs. For more information on multiclass LSPs, see the *JUNOS MPLS Applications Configuration Guide*.
- The service guarantee must be maintained across the entire length of the link as it transits the service provider's network. Different Layer 2 circuits could have different bandwidth requirements. However, many Layer 2 circuits could be transported over the same E-LSP in the MPLS core network.
- CAC ensures that the LSP has sufficient bandwidth to accommodate the Layer 2 circuit. If there is not enough bandwidth over a particular LSP, the Layer 2 circuit is prevented from using that LSP.

Selecting an LSP Based on the Bandwidth Constraint

CAC of Layer 2 circuits is based on the bandwidth constraint. You must configure this constraint for each Layer 2 circuit interface. If there is a bandwidth constraint configured for a Layer 2 circuit, CAC bases the final selection of which LSP-forwarding next hop to use on the following:

- If multiple LSPs meet the bandwidth requirements, the first LSP found that can satisfy the bandwidth requirements for the Layer 2 circuit is selected.
- If there is more than one next hop mapped to the same LSP, then all the next hops that map to that LSP and pass CAC constraints are installed. This allows the Layer 2 circuit routes to restore themselves quickly in case of failure.
- The available bandwidth on the selected LSP is decremented by the bandwidth required for each Layer 2 circuit. Similarly, when the Layer 2 circuit route is changed or deleted (for example, when the route is disassociated from that particular LSP), the bandwidth on the corresponding LSP is incremented.
- There are no priorities among different Layer 2 circuits competing for the same LSP next hop in the core network.
- When an LSP's bandwidth changes, the Layer 2 circuits using that LSP repeat the CAC process again.

  If the LSP bandwidth increases, some Layer 2 circuits that were not established might now successfully resolve over the LSP. Similarly, if the bandwidth of the

LSP decreases, some Layer 2 circuits that were previously up might now be declared down because of insufficient bandwidth on the LSP.

- When no LSP is found to meet the bandwidth requirements of the Layer 2 circuit, it is considered to be a CAC failure, and an error is reported.

LSP Path Protection and CAC

CAC can take into account LSPs that have been configured with an MPLS path protection feature, such as secondary paths, fast reroute, or node and link protection. CAC can consider the bandwidth available on these auxiliary links and can accept the backup connection as valid if the main connection fails. However, there are limitations on how the path protection feature must be configured to prevent CAC from taking down the Layer 2 circuit when the LSP it is using is switched to a backup route.

For more information on MPLS path protection features, see the *JUNOS MPLS Applications Configuration Guide*.

The sections that follow discuss the path protection features that can be used in conjunction with CAC and how they must be configured:

- Secondary Paths and CAC on page 530
- Fast Reroute and CAC on page 531
- Link and Node Protection and CAC on page 531

Secondary Paths and CAC

The following describes the ways in which secondary paths would interact with Layer 2 circuit CAC:

- If an LSP is configured with both primary and secondary paths, if the paths have the same bandwidth, and if this bandwidth is enough to accommodate the Layer 2 circuit, the Layer 2 circuit route installs both next hops in the forwarding table.

  CAC allows the Layer 2 circuit to be switched to the secondary path if the primary path fails.

- If the LSP has primary and secondary paths configured with different bandwidths, each path must run through CAC independently. If the active path for that LSP passes CAC constraints successfully, then that next hop is installed and the corresponding LSP is selected to transport the Layer 2 circuit traffic. The LSP's secondary paths are then checked for CAC, and installed if there is sufficient bandwidth.

  However, if the active path for the LSP fails to meet the CAC constraints, then that LSP is not selected and the system looks for a different LSP to transport the Layer 2 circuit.

  For example, an LSP has an active primary path with 30 megabits of bandwidth and a secondary path with 10 megabits of bandwidth. The Layer 2 circuit requires 15 megabits of bandwidth. The secondary path fails CAC, and only the next hop

corresponding to the primary path is installed for the Layer 2 circuit route. The path protection originally provided by the secondary path is no longer available.

Fast Reroute and CAC

No CAC is done for fast reroute detours. However, as long as the protected path satisfies the CAC bandwidth constraints, the detour next hop is also selected and installed.

Link and Node Protection and CAC

CAC cannot select or install the bypass route for a bandwidth-constrained Layer 2 circuit using an LSP that has link and node protection configured. Link and node protection is not available for these routes. If the protected LSP path goes down, even if the LSP switches to the bypass, CAC no longer uses that path for the Layer 2 circuit route. CAC reevaluates the Layer 2 circuit route and updates it either to use a different LSP or, if no LSP with sufficient bandwidth is found, the Layer 2 circuit is taken down and an error is reported.

Layer 2 Circuits Trunk Mode

Using Layer 2 circuit trunk mode, you can configure Layer 2 circuits to carry ATM trunks, providing a way to link ATM switches over an MPLS core network.

Layer 2 circuit trunk mode allows you to configure the following CoS features:

- CoS queues in Layer 2 circuit trunk mode—For ATM2 IQ interfaces, you can configure ATM CoS queues for Layer 2 circuit trunk mode.
- Layer 2 circuit trunk mode scheduling—For ATM2 IQ interfaces configured to use Layer 2 circuit trunk mode, you can share a scheduler among 32 trunks on an ATM port.
- Two early packet discard (EPD) thresholds per queue—For ATM2 IQ interfaces configured to use Layer 2 circuit trunk mode, you can set two EPD thresholds that depend on the packet-loss priorities (PLPs) of the packets.

For a detailed overview and configuration documentation, see the *JUNOS Network Interfaces Configuration Guide* and *JUNOS Class of Service Configuration Guide*.

Chapter 26

Layer 2 Circuit Configuration Guidelines

To configure a Layer 2 circuit, include the l2circuit statement:

```
l2circuit {
    local-switching {
        interface interface-name {
            description text;
            end-interface {
                interface interface-name;
                protect-interface interface-name;
            }
            ignore-mtu-mismatch;
            protect-interface interface-name;
        }
    }
    neighbor address {
        interface interface-name {
            community community-name;
            (control-word | no-control-word);
            description text;
            mtu mtu-number;
            protect-interface interface-name;
            psn-tunnel-endpoint address;
            virtual-circuit-id identifier;
        }
    }
    traceoptions {
        file filename <files number> <size size> <world-readable | no-world-readable>;
        flag flag <flag-modifier> <disable>;
    }
}
```

You can include the l2circuit statement at the following hierarchy levels:

- [edit protocols]
- [edit logical-systems *logical-system-name* protocols]

This chapter describes how to configure Layer 2 circuits, discussing the following topics:

Configuring Interfaces for Layer 2 Circuits

The following sections describe how to configure interfaces for Layer 2 circuits:

Configuring the Address for the Neighbor of the Layer 2 Circuit

All the Layer 2 circuits using a particular remote PE router designated for remote CE routers are listed under the neighbor statement ("neighbor" designates the PE router). Each neighbor is identified by its IP address and is usually the end-point destination for the label-switched path (LSP) tunnel transporting the Layer 2 circuit.

To configure a PE router as a neighbor for a Layer 2 circuit, specify the neighbor address using the neighbor statement:

```
neighbor address {
    ...
}
```

You can include the neighbor statement at the following hierarchy levels:

- [edit protocols l2circuit]
- [edit logical-systems *logical-system-name* protocols l2circuit]

Configuring the Neighbor Interface for the Layer 2 Circuit

Each Layer 2 circuit is represented by the logical interface connecting the local provider edge (PE) router to the local customer edge (CE) router. This interface is tied to the Layer 2 circuit neighbor configured in "Configuring the Address for the Neighbor of the Layer 2 Circuit" on page 534.

To configure the interface for a Layer 2 circuit neighbor, include the interface statement:

```
interface interface-name {
    bandwidth (bandwidth | ctnumber bandwidth);
    community community-name;
    (control-word | no-control-word);
    description text;
    ignore-encapsulation-mismatch;
    ignore-mtu-mismatch;
```

```
    mtu mtu-number;
    protect-interface interface-name;
    psn-tunnel-endpoint address;
    virtual-circuit-id identifier;
}
```

You can include the interface statement for the Layer 2 circuit neighbor at the following hierarchy levels:

- [edit protocols l2circuit neighbor *address*]
- [edit logical-systems *logical-system-name* protocols l2circuit neighbor *address*]

The following sections describe how to configure the interface for the Layer 2 circuit neighbor:

- Configuring a Community for the Layer 2 Circuit on page 535
- Configuring the Control Word for Layer 2 Circuits on page 535
- Configuring the MTU for the Layer 2 Circuit Neighbor Interface on page 537
- Configuring Layer 2 Circuits over Both RSVP and LDP LSPs on page 538
- Configuring the Protect Interface on page 539
- Configuring the Virtual Circuit ID on page 539

Configuring a Community for the Layer 2 Circuit

To configure a community for a Layer 2 circuit, include the community statement:

```
community community-name;
```

You can include this statement at the following hierarchy levels:

- [edit protocols l2circuit neighbor *address* interface *interface-name*]
- [edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*]

For information on how to configure a routing policy for a Layer 2 circuit, see "Configuring Layer 2 Circuit Policies" on page 542.

Configuring the Control Word for Layer 2 Circuits

To emulate the virtual circuit (VC) encapsulation for Layer 2 circuits, a 4-byte control word is added between the Layer 2 protocol data unit (PDU) being transported and the VC label that is used for demultiplexing. 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 following Layer 2 protocols map Layer 2 control information into special bit fields in the control word:

- Frame Relay—The 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 "Configuring the Control Word for Frame Relay Interfaces" on page 536.
- ATM AAL5 mode—The 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—The 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 IP and MPLS Networks*. The differences are as follows:

- A packet with a sequence number of 0 is considered as out of sequence.
- A packet that 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.

The following sections discuss how to configure the control word for Layer 2 circuits:

Configuring the Control Word for Frame Relay Interfaces

On interfaces with Frame Relay CCC encapsulation, you can configure Frame Relay control bit translation to support Frame Relay services over IP and Multiprotocol Label Switching (MPLS) backbones by using CCC, Layer 2 VPNs, and Layer 2 circuits. When you configure translation of Frame Relay control bits, the bits are mapped into the Layer 2 circuit control word and preserved across the IP or MPLS backbone.

For information on how to configure the control bits, see the *JUNOS Network Interfaces Configuration Guide* and the *JUNOS Feature Guide*.

Disabling the Control Word for Layer 2 Circuits

The JUNOS software can typically determine whether a neighboring router supports the control word. However, if you want to explicitly disable its use on a specific interface, include the no-control-word statement:

```
no-control-word;
```

For a list of hierarchy levels at which you can configure this statement, see the statement summary section for this statement.

Configuring the MTU for the Layer 2 Circuit Neighbor Interface

The following sections describe how to configure the MTU for the Layer 2 circuit neighbor interface:

- Enabling the Layer 2 Circuit When the Encapsulation Does Not Match on page 537
- Enabling the Layer 2 Circuit When the MTU Does Not Match on page 537
- Configuring the MTU Advertised for a Layer 2 Circuit on page 537

Enabling the Layer 2 Circuit When the Encapsulation Does Not Match

You can configure the JUNOS software to allow a Layer 2 circuit to be established even though the encapsulation configured on the CE device interface does not match the encapsulation configured on the Layer 2 circuit interface by including the ignore-encapsulation-mismatch statement:

```
ignore-encapsulation-mismatch;
```

You can include this statement at the following hierarchy levels:

- [edit protocols l2circuit neighbor *address* interface *interface-name*]
- [edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*]

Enabling the Layer 2 Circuit When the MTU Does Not Match

You can configure the JUNOS software to allow a Layer 2 circuit to be established even though the MTU configured on the PE router does not match the MTU configured on the remote PE router by including the ignore-mtu-mismatch statement:

```
ignore-mtu-mismatch;
```

You can configure this statement at the following hierarchy levels:

- [edit protocols l2circuit neighbor *address* interface *interface-name*]
- [edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*]

Configuring the MTU Advertised for a Layer 2 Circuit

By default, the MTU used to advertise a Layer 2 circuit is determined by taking the interface MTU for the associated physical interface and subtracting the encapsulation overhead for sending IP packets based on the encapsulation.

However, encapsulations that support multiple logical interfaces (and multiple Layer 2 circuits) rely on the same interface MTU (since they are all associated with the same physical interface). This can prove to be a limitation for VLAN Layer 2 circuits

using the same Ethernet interface or for Layer 2 circuit DLCIs using the same Frame Relay interface.

This can also affect multivendor environments. For example, if you have three PE devices supplied by different vendors and one of the devices only supports an MTU of 1500, even if the other devices support larger MTUs you must to configure the MTU as 1500 (the smallest MTU of the three PE devices).

You can explicitly configure which MTU is advertised for a Layer 2 circuit, even if the Layer 2 circuit is sharing a physical interface with other Layer 2 circuits. When you explicitly configure an MTU for a Layer 2 circuit, be aware of the following:

- An explicitly configured MTU is signaled to the remote PE device. The configured MTU is also compared to the MTU received from the remote PE device. If there is a conflict, the Layer 2 circuit is taken down.
- If you configure an MTU for an ATM cell relay interface on an ATM II PIC, the configured MTU is used to compute the cell bundle size advertised for that Layer 2 circuit, instead of the default interface MTU.
- A configured MTU is used only in the control plane. It is not enforced in the data plane. You need to ensure that the CE device for a given Layer 2 circuit uses the correct MTU for data transmission.

To configure the MTU for a Layer 2 circuit, include the mtu statement:

```
mtu mtu-number;
```

You can include this statement at the following hierarchy levels:

- [edit protocols l2circuit neighbor *address* interface *interface-name*]
- [edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*]

Configuring Layer 2 Circuits over Both RSVP and LDP LSPs

You can configure two Layer 2 circuits between the same two routers, and have one Layer 2 circuit traverse an RSVP LSP and the other traverse an LDP LSP. To accomplish this, you need to configure two loopback addresses on the local router. You configure one of the loopback address for the Layer 2 circuit traversing the RSVP LSP. You configure the other loopback address to handle the Layer 2 circuit traversing the LDP LSP. For information on how to configure multiple loop back interfaces, see “Configuring a Logical Unit on the Loopback Interface” on page 168.

You also need to configure a packet switched network (PSN) tunnel endpoint for one of the Layer 2 circuits. It can be either the Layer 2 circuit traversing the RSVP LSP or the one traversing the LDP LSP. The PSN tunnel endpoint address is the destination address for the LSP on the remote router.

To configure the address for the PSN tunnel endpoint, include the psn-tunnel-endpoint statement:

```
psn-tunnel-endpoint address;
```

You can configure this statement at the following hierarchy levels:

- [edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*]
- [edit protocols l2circuit neighbor *address* interface *interface-name*]

By default, the PSN tunnel endpoint for a Layer 2 circuit is identical to the neighbor address, which is also the same as the LDP neighbor address.

The tunnel endpoints on the remote router do not need to be loopback addresses.

Example: PSN Tunnel Endpoint

The following example illustrates how you might configure a PSN tunnel endpoint:

```
[edit protocols l2circuit]
neighbor 10.255.0.6 {
   interface t1-0/2/2.0 {
      psn-tunnel-endpoint 20.20.20.20;
      virtual-circuit-id 1;
   }
   interface t1-0/2/1.0 {
      virtual-circuit-id 10;
   }
}
```

The Layer 2 circuit configured for the t1-0/2/2.0 interface resolves in the inet3 routing table to 20.20.20.20. This could be either an RSVP route or a static route with an LSP next hop.

Configuring the Protect Interface

You can configure a protect interface for the logical interface linking a virtual circuit to its destination, whether the destination is remote or local. A protect interface provides a backup for the protected interface in case of failure. Network traffic uses the primary interface only so long as the primary interface functions. If the primary interface fails, traffic is switched to the protect interface. The protect interface is optional.

To configure the protect interface, include the protect-interface statement:

```
protect-interface interface-name;
```

For a list of hierarchy levels at which you can configure this statement, see the statement summary section for this statement.

For an example of how to configure a protect interface for a Layer 2 circuit, see “Layer 2 Circuits Example” on page 551.

Configuring the Virtual Circuit ID

You configure a virtual circuit ID on each interface. Each virtual circuit ID uniquely identifies the Layer 2 circuit among all the Layer 2 circuits to a specific neighbor.

The key to identifying a particular Layer 2 circuit on a PE router is the neighbor address and the virtual circuit ID. An LDP-FEC-to-label binding is associated with a Layer 2 circuit based on the virtual circuit ID in the FEC and the neighbor that sent this binding. The LDP-FEC-to-label binding enables the dissemination of the VPN label used for sending traffic on that Layer 2 circuit to the remote CE device.

You also configure a virtual circuit ID for each redundant pseudowire. A redundant pseudowire is identified by the backup neighbor address and the virtual circuit ID. For more information, see "Configuring Pseudowire Redundancy on the PE Router" on page 34.

To configure the virtual circuit ID, include the virtual-circuit-id statement:

```
virtual-circuit-id identifier;
```

For a list of hierarchy levels at which you can configure this statement, see the statement summary section for this statement.

Configuring the Interface Encapsulation Type for Layer 2 Circuits

The Layer 2 encapsulation type is carried in the Label Distribution Protocol (LDP) forwarding equivalence class (FEC). You can configure either circuit cross-connect (CCC) or translational cross-connect (TCC) encapsulation types for Layer 2 circuits. For more information, see the *JUNOS MPLS Applications Configuration Guide*.

To configure the interface encapsulation for a Layer 2 circuit, include the encapsulation-type statement:

```
encapsulation-type encapsulation-type;
```

You can include the encapsulation-type statement at the following hierarchy levels:

- [edit interfaces *interface-name*]
- [edit logical-systems *logical-system-name* interfaces *interface-name*]

Configuring ATM2 IQ Interfaces for Layer 2 Circuits

You can configure Asynchronous Transfer Mode 2 (ATM2) intelligent queuing (IQ) interfaces for Layer 2 circuits by using Layer 2 circuit ATM Adaptation Layer 5 (AAL5) transport mode, Layer 2 circuit ATM cell relay mode, and the Layer 2 circuit ATM trunk mode.

The configuration statements are as follows:

- atm-l2circuit-mode aal5
- atm-l2circuit-mode cell
- atm-l2circuit-mode trunk

For more information on these statements, see the *JUNOS System Basics Configuration Guide*. For more information on how to configure ATM2 IQ interfaces, see the *JUNOS Network Interfaces Configuration Guide*.

The JUNOS software implementation of sequence number processing for Layer 2 circuit ATM cell relay mode and Layer 2 circuit AAL5 mode differs from that described in the Internet draft draft-martini-l2circuit-encap-mpls-*version*.txt, *Frame Relay Encapsulation over Pseudo-Wires*.

The JUNOS software implementation has the following differences:

1. A packet with a sequence number of 0 is treated as out of sequence.
2. A packet that does not have the next incremental sequence number is considered out of sequence.

When out-of-sequence packets arrive, the expected sequence number for the neighbor is set to the sequence number in the Layer 2 circuit control word.

Configuring Local Interface Switching

You can configure a virtual circuit entirely on the local router, terminating the circuit on a local interface. Possible uses for this feature include being able to enable switching between Frame Relay DLCIs.

To configure a virtual circuit to terminate locally, include the local-switching statement:

```
local-switching {
    interface interface-name {
        description text;
        end-interface {
            interface interface-name;
            protect-interface interface-name;
        }
        ignore-mtu-mismatch;
        protect-interface interface-name;
    }
}
```

You can include the local-switching statement at the following hierarchy levels:

- [edit protocols l2circuit]
- [edit logical-systems *logical-system-name* protocols l2circuit]

The following sections describe how to configure local interface switching:

- Configuring the Interfaces for the Local Interface Switch on page 541
- Enabling Local Interface Switching When the MTU Does Not Match on page 542

Configuring the Interfaces for the Local Interface Switch

Local interface switching requires you to configure at least two interfaces:

- Starting interface—Configure using the interface statement at the [edit protocols l2circuit local-switching] hierarchy level.

- Ending interface—Configure using the end-interface statement at the [edit protocols l2circuit local-switching interface *interface-name*] hierarchy level.

You can also configure virtual circuit interface protection for each local interface:

- Protect interface for the starting interface—Configure using the protect-interface statement at the [edit protocols l2circuit local-switching interface *interface-name*] hierarchy level.
- Protect interface for the ending interface—Configure using the protect-interface statement at the [edit protocols l2circuit local-switching interface *interface-name* end-interface] hierarchy level.

For more information on how to configure protect interfaces, see "Configuring the Protect Interface" on page 539.

Enabling Local Interface Switching When the MTU Does Not Match

You can configure a local switching interface to ignore the MTU configuration set for the associated physical interface. This enables you to bring up a circuit between two logical interfaces that are defined on physical interfaces with different MTU values.

To configure the local switching interface to ignore the MTU configured for the physical interface, include the ignore-mtu-mismatch statement:

```
ignore-mtu-mismatch;
```

You can configure this statement at the following hierarchy levels:

- [edit protocols l2circuit local-switching interface *interface-name*]
- [edit logical-systems *logical-system-name* protocols l2circuit local-switching interface *interface-name*]

Configuring LDP for Layer 2 Circuits

Use LDP as the signaling protocol to advertise ingress labels to the remote PE routers. When configured, LDP examines the Layer 2 circuit configuration and initiates extended neighbor discovery for all the Layer 2 circuit neighbors (for example, remote PEs). This process is similar to how LDP works when tunneled over Resource Reservation Protocol (RSVP). You must run LDP on the lo0.0 interface for extended neighbor discovery to function correctly.

For detailed information about how to configure LDP, see the *JUNOS MPLS Applications Configuration Guide*.

Configuring Layer 2 Circuit Policies

You can configure JUNOS routing policies to control the flow of packets over Layer 2 circuits. This capability allows you to provide different level of service over a set of equal-cost Layer 2 circuits. For example, you can configure a circuit for high-priority traffic, a circuit for average-priority traffic, and a circuit for low-priority traffic. By

configuring Layer 2 circuit policies, you can ensure that higher-value traffic has a greater likelihood of reaching its destination.

The following sections explain how to configure Layer 2 circuit policies:

Configuring the Layer 2 Circuit Community

To configure a community for Layer 2 circuits, include the community statement.

```
community community-name {
    members [ community-ids ];
}
```

You can include the community statement at the following hierarchy levels:

- [edit policy-options]
- [edit logical-systems *logical-system-name* policy-options]

name identifies the community or communities.

community-ids identifies the type of community or extended community:

- A normal community uses the following community ID format:

  as-number:*community-value*

  as-number is the autonomous system (AS) number of the community member.

  community-value is the identifier of the community member. It can be a number from 0 through 65,535.

- An extended community uses the following community ID format:

  type:*administrator*:*assigned-number*

  type is the type of target community. The target community identifies the route's destination.

  administrator is either an AS number or an IP version 4 (IPv4) address prefix, depending on the type of community.

  assigned-number identifies the local provider.

  You also need to configure the community for the Layer 2 circuit interface; see "Configuring a Community for the Layer 2 Circuit" on page 535.

Configuring the Policy Statement for the Layer 2 Circuit Community

To configure a policy to send community traffic over a specific LSP, include the policy-statement statement:

```
policy-statement policy-name {
    term term-name {
        from community community-name;
        then {
            install-nexthop (except | lsp lsp-name | lsp-regex lsp-regular-expression);
            accept;
        }
    }
}
```

You can include the policy-statement statement at the following hierarchy levels:

- [edit policy-options]
- [edit logical-systems *logical-system-name* policy-options]

To prevent the installation of any matching next hops, include the install-nexthop statement with the except option:

```
install-nexthop except;
```

You can include the install-nexthop statement at the following hierarchy levels:

- [edit policy-options policy-statement *policy-name* term *term-name* then]
- [edit logical-systems *logical-system-name* policy-options policy-statement *policy-name* term *term-name* then]

To assign traffic from a community to a specific LSP, include the install-nexthop statement with the lsp *lsp-name* option and the accept statement:

```
install-nexthop lsp lsp-name;
accept;
```

You can include these statements at the following hierarchy levels:

- [edit policy-options policy-statement *policy-name* term *term-name* then]
- [edit logical-systems *logical-system-name* policy-options policy-statement *policy-name* term *term-name* then]

You can also use a regular expression to select an LSP from a set of similarly named LSPs for the install-nexthop statement. To configure a regular expression, include the install-nexthop statement with the lsp-regex option and the accept statement:

```
install-nexthop lsp-regex lsp-regular-expression;
accept;
```

You can include these statements at the following hierarchy levels:

- [edit policy-options policy-statement *policy-name* term *term-name* then]

- [edit logical-systems *logical-system-name* policy-options policy-statement *policy-name* term *term-name* then]

Example: Configuring a Policy for a Layer 2 Circuit Community

The following example illustrates how you might configure a regular expression in a Layer 2 circuit policy. You create three LSPs to handle gold-tier traffic from a Layer 2 circuit. The LSPs are named alpha-gold, beta-gold, and delta-gold. You then include the install-nexthop statement with the lsp-regex option with the LSP regular expression .*-gold at the [edit policy-options policy-statement *policy-name* term *term-name* then] hierarchy level:

```
[edit policy-options]
policy-statement gold-traffic {
   term to-gold-LSPs {
      from community gold;
      then {
         install-nexthop lsp-regex .*-gold;
         accept;
      }
   }
}
```

The community gold Layer 2 circuits can now use any of the -gold LSPs. Given equal utilization across the three -gold LSPs, LSP selection is made at random.

You need to apply the policy to the forwarding table. To apply a policy to the forwarding table, configure the export statement at the [edit routing-options forwarding-table] hierarchy level:

```
[edit routing-options forwarding-table]
export policy-name;
```

Verifying the Layer 2 Circuit Policy Configuration

To verify that you have configured a policy for the Layer 2 circuit, issue the show route table mpls detail command. It should display the community for ingress routes that corresponds to the Layer 2 circuits, as shown by the following example:

```
user@host> show route table mpls detail
so-1/0/1.0 (1 entry, 1 announced)
*L2VPN  Preference: 7
Next hop: via so-1/0/0.0 weight 1, selected
Label-switched-path to-community-gold
Label operation: Push 100000 Offset: -4
Next hop: via so-1/0/0.0 weight 1
Label-switched-path to-community-silver
Label operation: Push 100000 Offset: -4
Protocol next hop: 10.255.245.45
Push 100000 Offset: -4
Indirect next hop: 85333f0 314
State: <Active Int>
Local AS:   100
Age: 22
```

```
Task: Common L2 VC
Announcement bits (2): 0-KRT 1-Common L2 VC
AS path: I
Communities: 100:1
```

For more information on how to configure routing policies, see the *JUNOS Policy Framework Configuration Guide*.

Configuring ATM Trunking on Layer 2 Circuits

You can configure Layer 2 circuits to transport ATM traffic from directly connected ATM switches across an MPLS core network. Traffic from an ATM switch is received on the local PE router. The ATM cells are given an MPLS label and then sent across the MPLS network to the remote PE router. The receiving router removes the MPLS label from the ATM cell and then forwards the cell the receiving ATM switch.

NOTE: ATM trunking on Layer 2 circuits is supported only on T-series and M320 routers and ATM2 IQ PICs.

Figure 55: ATM Trunking on Layer 2 Circuits

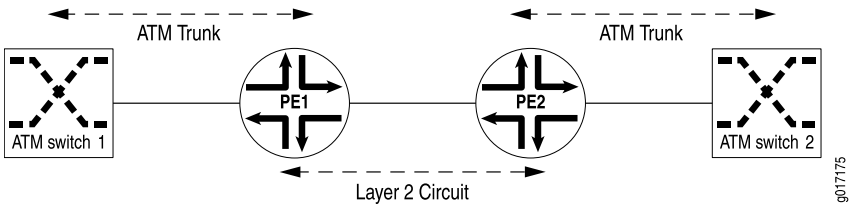


Figure 55 on page 546 illustrates how ATM switches could be linked together by a Layer 2 circuit. The PE1 Router is configured to receive ATM trunk traffic from ATM Switch 1. As each ATM cell is received on the PE1 Router, it is classified by means of the class-of-service (CoS) information in the cell header and then encapsulated as a labeled packet. The CoS information and cell loss priority (CLP) of the ATM cell are copied into the experimental (EXP) bits of the MPLS label. The labeled packet is then transported across the service provider network to the PE2 Router by means of a Layer 2 circuit.

On the PE2 Router, the label is removed and the plain ATM cell is forwarded to ATM Switch 2. The CoS and CLP are extracted from the EXP bits and are then used to select the correct output queue and determine whether the ATM cell should be dropped.

The ATM physical port on the router can support 32 logical trunks when network-to-network interface (NNI) is used and 8 logical trunks when user-to-network interface (UNI) is used. A trunk can carry traffic on 32 virtual path identifiers (VPIs), numbered 0 through 31. Each ATM trunk is associated with an MPLS label and a logical interface. On the ingress router, one or more of these trunks are mapped to a Layer 2 circuit.

The configuration for the Layer 2 circuit between PE routers is conventional. Follow the procedures outlined in this chapter for configuring the circuit. However, there is some specific configuration you need to complete for the Layer 2 circuit to carry traffic from an ATM trunk.

First, enable ATM trunking for Layer 2 circuits. To enable ATM trunking for Layer 2 circuits, specify the trunk option for the atm-l2circuit-mode statement:

```
atm-l2circuit-mode trunk (uni | nni);
```

You can include the atm-l2circuit-mode statement at the [edit chassis fpc *number* pic *number*] hierarchy level.

Specify the uni option for UNI trunks and the nni option for NNI trunks. The default option is uni.

You also need to configure each ATM trunk for a specific logical interface. Each ATM trunk has a trunk identifier in the range from 0 to 31. This configuration step is in addition to the typical configuration steps you follow related to configuring interfaces for Layer 2 circuits, as described in "Configuring Interfaces for Layer 2 Circuits" on page 534.

To associate a specific trunk identifier with a logical interface, include the trunk-id statement:

```
trunk-id number;
```

You can include the trunk-id statement at the following hierarchy levels:

- [edit interfaces *interface-name* unit *number*]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *number*]

Since ATM trunking is supported on ATM2 IQ PICs only, the only value you can configure for the pic-type statement is atm2. If you do not configure the pic-type statement but you do configure the trunk option for the atm-l2circuit-mode statement (at the [chassis fpc *number* pic *number*] hierarchy level), the pic-type statement defaults to atm2.

Configuring Bandwidth Allocation and Call Admission Control

You can configure bandwidth allocation and call admission control (CAC) on Layer 2 circuits. This feature is available for RSVP-signaled LSPs traversing an MPLS network.

When you enable bandwidth allocation on a Layer 2 circuit, attempts to establish an RSVP-signaled LSP are preceded by a check of the available bandwidth on the network. This check is the CAC. The available bandwidth is compared to the bandwidth requested by the LSP. If there is insufficient bandwidth, the Layer 2 circuit is not established and an error message is generated. To apply CAC to a Layer 2 circuit, a bandwidth constraint must be configured.

You can specify the bandwidth for a Layer 2 circuit without configuring a bandwidth for each class type (queue). To specify the bandwidth allocation for a Layer 2 circuit, include the bandwidth statement:

```
bandwidth bandwidth;
```

Specify the bandwidth in bits per second.

You can include the bandwidth statement at the following hierarchy levels:

- [edit protocols l2circuit neighbor *address* interface *interface-name*]
- [edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*]

Alternatively, you can configure the bandwidth for each class type on a Layer 2 circuit. If you use this type of configuration, you cannot simultaneously configure the nonclass type of bandwidth configuration for the Layer 2 circuit (the commit operation fails).

To configure the bandwidth for each class type on an Layer 2 circuit, include the bandwidth statement:

```
bandwidth {
    ct0 bandwidth;
    ct1 bandwidth;
    ct2 bandwidth;
    ct3 bandwidth;
}
```

You can include the bandwidth statement at the following hierarchy levels:

- [edit protocols l2circuit neighbor *address* interface *interface-name*]
- [edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*]

Specify the bandwidth for each class type in bits per second. It is not necessary to specify a bandwidth for all four class types.

Tracing Layer 2 Circuit Creation and Changes

To trace the creation of and changes to Layer 2 circuits, you can specify options in the traceoptions statement:

```
traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
}
```

You can include the traceoptions statement at the following hierarchy levels:

- [edit protocols l2circuit]
- [edit logical-systems *logical-system-name* protocols l2circuit]

The following tracing flags display the operations associated with Layer 2 circuits:

- connections—Layer 2 circuit connections (events and state changes)
- error—Error conditions
- FEC—Layer 2 circuit advertisements received or sent using LDP
- topology—Layer 2 circuit topology changes caused by reconfiguration or advertisements received from other PE routers

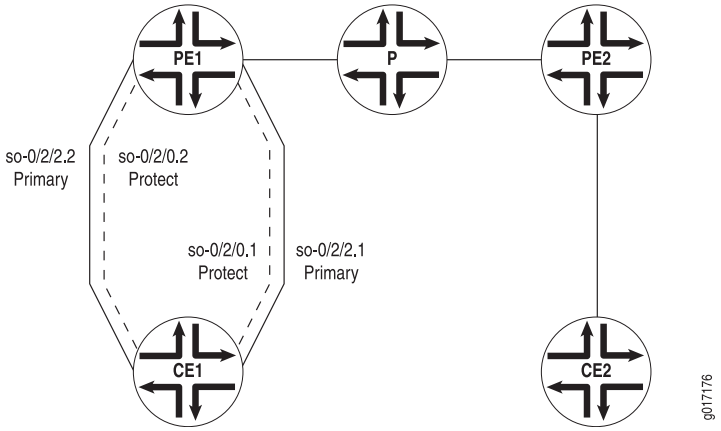
Chapter 27

Layer 2 Circuits Example

The example presented in this chapter illustrates how you might configure a Layer 2 circuit with protect interfaces. Protect interfaces act as backups for their associated interfaces. The primary interface has priority over the protect interface and carries network traffic as long as it is functional. If the primary interface fails, the protect interface is activated. These interfaces can also share the same virtual path identifier (VPI) or virtual circuit identifier (VCI).

For more examples on how to configure Layer 2 circuits, see the *JUNOS Feature Guide*.

Figure 56 on page 551 shows the network topology used in this example.



The following sections describe how to configure a Layer 2 circuit to use a protect interface:

Configuring Router PE1

Configure an interface for traffic to Router CE1 from Router PE1 at the [edit interfaces] hierarchy level:

```
[edit interfaces]
```

```
so-0/2/2 {
    description "Router CE1 so-0/2/2";
    no-keepalives;
    encapsulation frame-relay-ccc;
    unit 1 {
        encapsulation frame-relay-ccc;
        point-to-point;
        dlci 600;
    }
    unit 2 {
        encapsulation frame-relay-ccc;
        point-to-point;
        dlci 602;
    }
}
```

Configure an interface for traffic to Router CE1 from Router PE1 at the [edit interfaces] hierarchy level. Logical interface so-0/2/0.2 acts as the protect interface for so-0/2/2.2, and logical interface so-0/2/0.1 acts as the protect interface for so-0/2/2.1:

```
[edit interfaces]
so-0/2/0 {
    description "to Router CE1 so-0/3/0";
    no-keepalives;
    encapsulation frame-relay-ccc;
    unit 1 {
        encapsulation frame-relay-ccc;
        dlci 600;
    }
    unit 2 {
        encapsulation frame-relay-ccc;
        dlci 602;
    }
}
```

Configure an interface for traffic to Router PE2 from Router PE1 at the [edit interfaces] hierarchy level:

```
[edit interfaces]
so-0/2/1 {
    description "to Router PE2 so-1/0/1";
    unit 0 {
        family inet {
            address 100.100.40.22/32 {
                destination 100.100.40.23;
            }
        }
        family iso;
        family mpls;
    }
}
```

Configure an interface for traffic to Router PE2 from Router PE1 at the [edit interfaces] hierarchy level:

```
[edit interfaces]
so-0/2/3 {
   description "Router PE2 so-1/0/3";
   unit 0 {
      family inet;
      family iso;
      family mpls;
   }
   lo0 {
      unit 0 {
         family inet {
            address 127.0.0.1/32;
            address 10.100.40.200/32;
         }
         family iso {
            address 47.0005.80ff.f800.0000.0108.0001.1921.6800.4213.00;
         }
      }
   }
}
```

Configure the Layer 2 circuit by including the l2circuit statement at the [edit protocols] hierarchy level. The logical interfaces for the Layer 2 circuits and their corresponding protect interfaces are included here:

```
[edit protocols]
l2circuit {
   neighbor 10.100.40.210 {
      interface so-0/2/2.2 {
         protect-interface so-0/2/0.2;
         virtual-circuit-id 2;
         no-control-word;
      }
      interface so-0/2/2.1 {
         protect-interface so-0/2/0.1;
         virtual-circuit-id 1;
         no-control-word;
      }
   }
}
```

Configuring Router PE2

Configure an interface for traffic to Router CE2 from Router PE2:

```
[edit interfaces]
so-1/0/0 {
   description "to Router CE2 so-0/2/0";
   no-keepalives;
   encapsulation frame-relay-ccc;
   unit 1 {
      encapsulation frame-relay-ccc;
      point-to-point;
      dlci 700;
   }
```

```
    unit 2 {
        encapsulation frame-relay-ccc;
        point-to-point;
        dlci 702;
    }
}
```

Configure an interface for traffic to Router PE1 from Router PE2:

```
[edit interfaces]
so-1/0/1 {
    description "to Router PE1 so-0/2/1";
    unit 0 {
        family inet {
            address 100.100.40.23/32 {
                destination 100.100.40.22;
            }
        }
        family iso;
        family mpls;
    }
}
```

Configure an interface for traffic to Router PE1 from Router PE2:

```
[edit interfaces]
so-1/0/3 {
    description "to Router PE1 so-0/2/3";
    unit 0 {
        family inet;
        family iso;
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 127.0.0.1/32;
            address 10.100.40.210/32;
        }
        family iso {
            address 47.0005.80ff.f800.0000.0108.0001.1921.6800.4216.00;
        }
    }
}
```

Configure the Layer 2 circuit at the [edit protocols] hierarchy level:

```
[edit protocols]
l2circuit {
    neighbor 10.100.40.200 {
        interface so-1/0/0.1 {
            virtual-circuit-id 1;
            no-control-word;
        }
        interface so-1/0/0.2 {
```

```
        virtual-circuit-id 2;
        no-control-word;
      }
    }
}
```

Configuring Router CE1

Configure an interface for traffic to Router PE1 from Router CE1:

```
[edit interfaces]
so-0/3/0 {
    description "to Router PE1 so-0/2/0";
    no-keepalives;
    encapsulation frame-relay;
    unit 1 {
        dlci 601;
        family inet {
            address 12.12.12.1/24;
        }
    }
}
```

Configure an interface for traffic to Router PE1 from Router CE1:

```
[edit interfaces]
so-0/3/1 {
    description "Router PE1 so-0/2/2";
    no-keepalives;
    encapsulation frame-relay;
    unit 0 {
        dlci 600;
        family inet {
            address 10.10.10.1/24;
            address 11.1.1.1/24;
        }
        family iso;
        family mpls;
    }
    unit 2 {
        dlci 602;
        family inet {
            address 13.13.13.1/24;
        }
    }
}
```

Configuring Router CE2

Configure an interface for traffic to Router PE2 from Router CE2:

```
[edit interfaces]
so-0/2/0 {
    description "to Router PE2 so-1/0/0";
```

```
    no-keepalives;
    encapsulation frame-relay;
    unit 1 {
        dlci 700;
        family inet {
            address 10.10.10.2/24;
            address 11.1.1.2/24;
            address 12.12.12.2/24;
        }
    }
    unit 2 {
        dlci 702;
        family inet {
            address 13.13.13.2/24;
        }
    }
}
```

Chapter 28

Summary of Layer 2 Circuit Configuration Statements

The following sections explain the major protocol configuration statements that apply specifically to Layer 2 circuits. The statements are organized alphabetically. Protocols and the statements at the [edit protocols] hierarchy level are explained in the *JUNOS Routing Protocols Configuration Guide*.

bandwidth

Syntax bandwidth (*bandwidth* | ct*number bandwidth*);

Hierarchy Level [edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*],
[edit protocols l2circuit neighbor *address* interface *interface-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Specify bandwidth allocation for a Layer 2 circuit or for the class types of a Layer 2 circuit.

Options *bandwidth*—Configure the bandwidth in bits per second for the Layer 2 circuit. You cannot configure the bandwidth for the Layer 2 circuit and for the class types at the same time.

ct*number bandwidth*—Configure the bandwidth in bits per second for a class type on the Layer 2 circuit. You can configure bandwidth for up to 4 class types (ct0, ct1, ct2, ct3) per Layer 2 circuit. If you configure the class types, you must configure them in order, starting with class type ct0.

Usage Guidelines See "Configuring Bandwidth Allocation and Call Admission Control" on page 547.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

community

Syntax

```
community community-name {
    invert-match;
    members community-members;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* policy-options],
[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*],
[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name* backup-neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls neighbor *address* backup-neighbor *address*],
[edit policy-options],
[edit protocols l2circuit neighbor *address* interface *interface-name*],
[edit protocols l2circuit neighbor *address* interface *interface-name* backup-neighbor *address*],
[edit routing-instances *routing-instance-name* protocols vpls neighbor *address* backup-neighbor *address*]

Release Information

Statement introduced before JUNOS Release 7.4. Hierarchy levels associated with the backup-neighbor statement (pseudowire redundancy) added in JUNOS Release 9.2.

Description

Specify the community for the Layer 2 circuit.

Options

invert-match—Invert the results of the community expression match.

members *community-members*—Specify the members of the community.

Usage Guidelines

See "Configuring the Layer 2 Circuit Community" on page 543 and "Configuring Pseudowire Redundancy on the PE Router" on page 34.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

control-word

Syntax (control-word | no-control-word);

Hierarchy Level [edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*],
[edit protocols l2circuit neighbor *address* interface *interface-name*],

Release Information Statement introduced before JUNOS Release 7.4.

Description Specify the control word. The control word is 4 bytes long and is inserted between the Layer 2 protocol data unit (PDU) being transported and the virtual circuit (VC) label that is used for demultiplexing.

Options control-word—Enable the use of the control word.
Default: A null control word is enabled by default. You can also configure the control word explicitly using the control-word statement.

no-control-word—Disable the use of the control word.

Usage Guidelines See "Configuring the Control Word for Frame Relay Interfaces" on page 536.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

description

Syntax description *text*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*],
[edit protocols l2circuit neighbor *address* interface *interface-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Provide a text description for the Layer 2 circuit. If the text includes one or more spaces, enclose the entire text string in quotation marks (" ").

Usage Guidelines See "Configuring the Description" on page 19.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

end-interface

Syntax
```
end-interface {
    interface interface-name;
    protect-interface interface-name;
}
```

Hierarchy Level
```
[edit logical-systems logical-system-name protocols l2circuit local-switching interface
    interface-name],
[edit protocols l2circuit local-switching interface interface-name]
```

Release Information Statement introduced before JUNOS Release 7.4.

Description Specify the end interface for a local interface switch.

The remaining statements are explained separately

Usage Guidelines See "Configuring Local Interface Switching" on page 541.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration

ignore-encapsulation-mismatch

Syntax
```
ignore-encapsulation-mismatch;
```

Hierarchy Level
```
[edit logical-systems logical-system-name protocols l2circuit local-switching interface
    interface-name neighbor address],
[edit protocols l2circuit local-switching interface interface-name neighbor address]
```

Release Information Statement introduced in JUNOS Release 9.2.

Description Allow a Layer 2 circuit to be established even though the encapsulation configured on the CE device interface does not match the encapsulation configured on the Layer 2 circuit interface.

Usage Guidelines See "Enabling the Layer 2 Circuit When the Encapsulation Does Not Match" on page 537.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration

ignore-mtu-mismatch

Syntax ignore-mtu-mismatch;

Hierarchy Level [edit logical-systems *logical-system-name* protocols l2circuit local-switching interface *interface-name*],
[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*],
[edit protocols l2circuit local-switching interface *interface-name*],
[edit protocols l2circuit neighbor *address* interface *interface-name*]

Release Information Statement introduced in JUNOS Release 8.5. Support for remote PE routers added in JUNOS Release 9.2.

Description Ignore the MTU configuration set for the physical interface associated with the local switching interface or with the remote PE router. This allows a Layer 2 circuit to be brought up between two logical interfaces that are defined on physical interfaces with different MTU values.

Usage Guidelines See "Enabling the Layer 2 Circuit When the MTU Does Not Match" on page 537 and "Enabling Local Interface Switching When the MTU Does Not Match" on page 542.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration

install-nexthop

Syntax install-nexthop (except | lsp *lsp-name* | lsp-regex *lsp-regular-expression*);

Hierarchy Level [edit logical-systems *logical-system-name* policy-options policy-statement *policy-name* term *term-name* then],
[edit policy-options policy-statement *policy-name* term *term-name* then]

Release Information Statement introduced before JUNOS Release 7.4.

Description Select a specific label-switched path (LSP), or select an LSP from a set of similarly named LSPs as the traffic destination for the configured community. Also can prevent the installation of any matching next hops.

Options except—Prevent the installation of any matching next hops.

lsp *lsp-name*—Configure a specific LSP.

lsp-regex *lsp-regular-expression*—Configure a range of similarly named LSPs. You can use the following wildcard characters when configuring an LSP regular expression:

- Asterisk (*)—Match any characters.
- Period (.)—Match any single digit.

Usage Guidelines See "Configuring the Policy Statement for the Layer 2 Circuit Community" on page 544.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

interface

Syntax

```
interface interface-name {
    bandwidth (bandwidth | ctnumber bandwidth);
    community community-name;
    (control-word | no-control-word);
    description text;
    ignore-encapsulation-mismatch;
    ignore-mtu-mismatch;
    mtu mtu-number;
    protect-interface interface-name;
    psn-tunnel-endpoint address;
    virtual-circuit-id identifier;
}
```

Hierarchy Level [edit logical-systems *logical-system-name* protocols l2circuit local-switching],
[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address*],
[edit protocols l2circuit local-switching],
[edit protocols l2circuit neighbor *address*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Interface over which Layer 2 circuit traffic travels.

Options *interface-name*—Name of the interface to configure.

The remaining statements are explained separately.

Usage Guidelines See "Configuring the Neighbor Interface for the Layer 2 Circuit" on page 534.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

l2circuit

Syntax

```
l2circuit {
    local-switching {
        interface interface-name {
            description text;
            end-interface {
                interface interface-name;
                protect-interface interface-name;
            }
            ignore-mtu-mismatch;
            protect-interface interface-name;
        }
    }
    neighbor address {
        interface interface-name {
            bandwidth (bandwidth | ctnumber bandwidth);
            community community-name;
            (control-word | no-control-word);
            description text;
            ignore-encapsulation-mismatch;
            ignore-mtu-mismatch;
            mtu mtu-number;
            protect-interface interface-name;
            psn-tunnel-endpoint address;
            virtual-circuit-id identifier;
        }
    }
    traceoptions {
        file filename <files number> <size size> <world-readable | no-world-readable>;
        flag flag <flag-modifier> <disable>;
    }
}
```

Hierarchy Level [edit logical-systems *logical-system-name* protocols],
[edit protocols]

Release Information Statement introduced before JUNOS Release 7.4.

Description Enables a Layer 2 circuit.

The remaining statements are explained separately.

Usage Guidelines See “Layer 2 Circuit Configuration Guidelines” on page 533.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

local-switching

Syntax

```
local-switching {
    interface interface-name {
        description text;
        end-interface {
            interface interface-name;
            protect-interface interface-name;
        }
        ignore-mtu-mismatch;
        protect-interface interface-name;
    }
}
```

Hierarchy Level [edit logical-systems *logical-system-name* protocols l2circuit],
[edit protocols l2circuit]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure a local switching interface. A local switching interface allows you to terminate a virtual circuit on the local router.

The remaining statements are explained separately.

Usage Guidelines See "Configuring Local Interface Switching" on page 541.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

mtu

Syntax mtu *mtu-number*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*],
[edit protocols l2circuit neighbor *address* interface *interface-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure the MTU to be advertised for the Layer 2 circuit.

Options *mtu-number*—MTU number to be advertised for the Layer 2 circuit.

Usage Guidelines See "Configuring the MTU Advertised for a Layer 2 Circuit" on page 537.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

neighbor

Syntax

```
neighbor address {
    interface interface-name {
        bandwidth (bandwidth | ctnumber bandwidth);
        community community-name;
        (control-word | no-control-word);
        description text;
        ignore-encapsulation-mismatch;
        ignore-mtu-mismatch;
        mtu mtu-number;
        protect-interface interface-name;
        psn-tunnel-endpoint address;
        virtual-circuit-id identifier;
    }
}
```

Hierarchy Level [edit logical-systems *logical-system-name* protocols l2circuit],
[edit protocols l2circuit]

Release Information Statement introduced before JUNOS Release 7.4.

Description Each Layer 2 circuit is represented by the logical interface connecting the local provider edge (PE) router to the local customer edge (CE) router. All the Layer 2 circuits using a particular remote PE router designated for remote CE routers are listed under the neighbor statement (neighbor designates the PE router). Each neighbor is identified by its IP address and is usually the end-point destination for the LSP tunnel (transporting the Layer 2 circuit).

Options *address*—IP address of a neighboring router.

The remaining statements are explained separately.

Usage Guidelines See "Configuring the Neighbor Interface for the Layer 2 Circuit" on page 534.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

no-control-word

See control-word

protect-interface

Syntax protect-interface *interface-name*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*],
[edit logical-systems *logical-system-name* protocols l2circuit local-switching interface *interface-name*],
[edit logical-systems *logical-system-name* protocols l2circuit local-switching interface *interface-name* end-interface],
[edit protocols l2circuit local-switching interface *interface-name*],
[edit protocols l2circuit neighbor *address* interface *interface-name*],
[edit protocols l2circuit local-switching interface *interface-name* end-interface]

Release Information Statement introduced before JUNOS Release 7.4.

Description Provide a backup for the protected interface in case of failure. Network traffic uses the primary interface only, as long as the primary interface functions.

Options *interface-name*—Name of the protect interface to configure.

Usage Guidelines See "Configuring the Protect Interface" on page 539.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

psn-tunnel-endpoint

Syntax psn-tunnel-endpoint *address*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*],
[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name* backup-neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls neighbor *address* backup-neighbor *address*],
[edit protocols l2circuit neighbor *address* interface *interface-name*],
[edit protocols l2circuit neighbor *address* interface *interface-name* backup-neighbor *address*],
[edit routing-instances *routing-instance-name* protocols vpls neighbor *address* backup-neighbor *address*]

Release Information Statement introduced before JUNOS Release 7.4. Hierarchy levels associated with the backup-neighbor statement added in JUNOS Release 9.2.

Description Specify the endpoint of the packet switched network (PSN) tunnel on the remote PE router.

Options *address*—Address for the tunnel endpoint.

Usage Guidelines See “Configuring Layer 2 Circuits over Both RSVP and LDP LSPs” on page 538 and “Configuring Pseudowire Redundancy on the PE Router” on page 34.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

traceoptions

Syntax

```
traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
}
```

Hierarchy Level [edit logical-systems *logical-system-name* protocols l2circuit],
[edit protocols l2circuit]

Release Information Statement introduced before JUNOS Release 7.4.

Description Trace traffic flowing through a Layer 2 circuit.

Options disable—(Optional) Disable the tracing operation. You can use this option to disable a single operation when you have defined a broad group of tracing operations, such as all.

file *filename*—Name of the file to receive the output of the tracing operation. Enclose the name in quotation marks (" ").

files *number*—(Optional) Maximum number of trace files. When a trace file named *trace-file* reaches its maximum size, it is renamed *trace-file*.0, then *trace-file*.1, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum number of files, you also must specify a maximum file size with the size option.
Range: 2 through 1000 files
Default: 2 files

flag *flag*—Tracing operation to perform. To specify more than one tracing operation, include multiple flag statements.

- connections—Layer 2 circuit connections (events and state changes)
- error—Error conditions
- fec—Layer 2 circuit advertisements received or sent by means of the Label Distribution Protocol (LDP)
- topology—Layer 2 circuit topology changes caused by reconfiguration or advertisements received from other PE routers

flag-modifier—(Optional) Modifier for the tracing flag. You can specify the detail modifier if you want to provide detailed trace information.

no-world-readable—(Optional) Do not allow any user to read the log file.

size *size*—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named *trace-file* reaches this size, it is

renamed *trace-file*.0. When the *trace-file* again reaches its maximum size, *trace-file*.0 is renamed *trace-file*.1 and *trace-file* is renamed *trace-file*.0. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum file size, you also must specify a maximum number of trace files with the files option.
Syntax: *x*k to specify kilobytes, *x*m to specify megabytes, or *x*g to specify gigabytes
Range: 10 KB through the maximum file size supported on your system
Default: 1 MB

world-readable—(Optional) Allow any user to read the log file.

Usage Guidelines See "Tracing Layer 2 Circuit Creation and Changes" on page 548.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

virtual-circuit-id

Syntax virtual-circuit-id *identifier*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*],
[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name* backup-neighbor *address*],
[edit protocols l2circuit neighbor *address* interface *interface-name*],
[edit protocols l2circuit neighbor *address* interface *interface-name* backup-neighbor *address*]

Release Information Statement introduced before JUNOS Release 7.4. Hierarchy levels for backup-neighbor (pseudowire redundancy) added in JUNOS Release 9.2.

Description Uniquely identify a Layer 2 circuit for either a regular pseudowire or a redundant pseudowire.

Options *identifier*—1 through 4,294,967,295

Usage Guidelines See "Configuring the Virtual Circuit ID" on page 539 and "Configuring Pseudowire Redundancy on the PE Router" on page 34.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Part 8
Indexes

Index

E

F

G

I

L

M

N

O

P

R

S

T

U

V

Index of Statements and Commands

N

P

R

S

T

U

V