



Junos[®] OS

MPLS Configuration Guide for Security Devices

Release
12.1



Published: 2012-03-07

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

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

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

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

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

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

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

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

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

Junos OS MPLS Configuration Guide for Security Devices

Release 12.1

Copyright © 2012, Juniper Networks, Inc.

All rights reserved.

Revision History

March 2012—R1 Junos OS 12.1

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

YEAR 2000 NOTICE

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

SOFTWARE LICENSE

The terms and conditions for using this software are described in the software license contained in the acknowledgment to your purchase order or, to the extent applicable, to any reseller agreement or end-user purchase agreement executed between you and Juniper Networks. By using this software, you indicate that you understand and agree to be bound by those terms and conditions. Generally speaking, the software license restricts the manner in which you are permitted to use the software and may contain prohibitions against certain uses. The software license may state conditions under which the license is automatically terminated. You should consult the license for further details. For complete product documentation, please see the Juniper Networks website at www.juniper.net/techpubs.

END USER LICENSE AGREEMENT

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

Abbreviated Table of Contents

	About This Guide	xi
Part 1	MPLS	
Chapter 1	MPLS	3
Chapter 2	Traffic Engineering	13
Chapter 3	MPLS VPNs	47
Chapter 4	CLNS VPNs	65
Chapter 5	VPLS	81
Part 2	Index	
	Index	135

Table of Contents

	About This Guide	xi
	J Series and SRX Series Documentation and Release Notes	xi
	Objectives	xii
	Audience	xii
	Supported Routing Platforms	xii
	Document Conventions	xii
	Documentation Feedback	xiv
	Requesting Technical Support	xiv
	Self-Help Online Tools and Resources	xiv
	Opening a Case with JTAC	xv
Part 1	MPLS	
Chapter 1	MPLS	3
	MPLS Overview	3
	Label Switching	4
	Label-Switched Paths	4
	Label-Switching Routers	5
	Labels	6
	Label Operations	6
	Penultimate Hop Popping	7
	LSP Establishment	7
	Static LSPs	7
	Dynamic LSPs	7
	MPLS Configuration Overview	8
	Example: Deleting Security Services	9
	Example: Enabling MPLS	10
Chapter 2	Traffic Engineering	13
	MPLS Traffic Engineering and Signaling Protocols Overview	13
	LDP Signaling Protocol	14
	Understanding the LDP Signaling Protocol	14
	Example: Configuring LDP-Signaled LSPs	15
	RSVP Signaling Protocol	18
	Understanding the RSVP Signaling Protocol	19
	RSVP Fundamentals	19
	Bandwidth Reservation Requirement	19
	Explicit Route Objects	20
	Constrained Shortest Path First	21
	Link Coloring	21
	Example: Configuring RSVP-Signaled LSPs	22

	Example: Configuring Point-to-Multipoint LSPs with Static Routes	26
	Understanding Point-to-Multipoint LSPs	26
	Point-to-Multipoint LSP Configuration Overview	27
	Example: Configuring an RSVP-Signaled Point-to-Multipoint LSP	27
Chapter 3	MPLS VPNs	47
	MPLS VPN Overview	47
	MPLS VPN Topology	47
	MPLS VPN Routing	48
	VRF Instances	48
	Route Distinguishers	49
	Layer 2 VPNs	50
	Understanding MPLS Layer 2 VPNs	50
	MPLS Layer 2 VPN Configuration Overview	50
	Configuring MPLS for Layer 2 VPNs (CLI Procedure)	52
	Configuring MPLS for Layer 2 VPNs (CLI Procedure)	53
	Configuring a BGP Session for MPLS VPNs (CLI Procedure)	53
	Configuring an IGP and the LDP Signaling Protocol (CLI Procedure)	54
	Configuring an IGP and the RSVP Signaling Protocol (CLI Procedure)	55
	Configuring Routing Options for MPLS VPNs (CLI Procedure)	55
	Configuring a Routing Instance for MPLS VPNs (CLI Procedure)	55
	Configuring a Routing Policy for MPLS Layer 2 VPNs (CLI Procedure)	56
	Verifying an MPLS Layer 2 VPN Configuration	57
	Layer 3 VPNs	57
	Understanding MPLS Layer 3 VPNs	58
	MPLS Layer 3 VPN Configuration Overview	58
	Configuring a Routing Policy for MPLS Layer 3 VPNs (CLI Procedure)	60
	Verifying an MPLS Layer 3 VPN Configuration	60
	Layer 2 Circuits	60
	Understanding MPLS Layer 2 Circuits	60
	MPLS Layer 2 Circuit Configuration Overview	61
	Configuring an MPLS Layer 2 Circuit (CLI Procedure)	62
	Verifying an MPLS Layer 2 Circuit Configuration	62
Chapter 4	CLNS VPNs	65
	CLNS Overview	65
	CLNS Configuration Overview	66
	Example: Configuring a VPN Routing Instance for CLNS	67
	ES-IS for CLNS	69
	Understanding ES-IS for CLNS	69
	Example: Configuring ES-IS for CLNS	70
	IS-IS for CLNS	71
	Understanding IS-IS for CLNS	71
	Example: Configuring IS-IS for CLNS	72
	Example: Configuring Static Routes for CLNS	74
	Understanding Static Routes for CLNS	74
	Example: Configuring Static Routes for CLNS	74

	BGP for CLNS VPNs	76
	Understanding BGP for CLNS VPNs	76
	Example: Configuring BGP for CLNS VPNs	77
	Verifying a CLNS VPN Configuration	78
Chapter 5	VPLS	81
	VPLS Overview	81
	Sample VPLS Topology	82
	VPLS on PE Routers	82
	Using an Ethernet Switch as the VPLS CE Device	84
	VPLS Exceptions on J Series and SRX Series Devices	84
	VPLS Configuration Overview	85
	VPLS Interfaces	86
	Understanding VPLS Interfaces	86
	Interface Name	87
	Encapsulation Type	87
	Flexible VLAN Tagging	87
	VLAN Rewrite	88
	Example: Configuring Routing Interfaces on the VPLS PE Router	88
	Example: Configuring the Interface to the VPLS CE Device	89
	VPLS Routing Instances	90
	Understanding VPLS Routing Instances	90
	BGP Signaling	91
	VPLS Routing Table	91
	Trace Options	92
	Example: Configuring the VPLS Routing Instance	93
	VPLS VLAN Encapsulation	95
	Understanding VPLS VLAN Encapsulation	95
	Example: Configuring VPLS VLAN Encapsulation on Gigabit Ethernet Interfaces	96
	Understanding VPLS VLAN Encapsulation on a Logical Interface	97
	Example: Configuring VPLS VLAN Encapsulation	97
	Example: Configuring Extended VLAN VPLS Encapsulation	100
	VPLS Filters and Policers	102
	VPLS Filters and Policers Overview	102
	Example: Configuring VPLS Policers	102
	Example: Configuring VPLS Filters	104
	Example: Configuring OSPF on the VPLS PE Router	106
	Example: Configuring RSVP on the VPLS PE Router	107
	Example: Configuring MPLS on the VPLS PE Router	108
	Example: Configuring LDP on the VPLS PE Router	110
	Example: Configuring Routing Options on the VPLS PE Router	111
	Example: Configuring BGP on the VPLS PE Router	112
	Example: Configuring VPLS over GRE with IPSec VPNs	113
Part 2	Index	
	Index	135

About This Guide

This preface provides the following guidelines for using the *Junos OS MPLS Configuration Guide for Security Devices*:

- [J Series and SRX Series Documentation and Release Notes on page xi](#)
- [Objectives on page xii](#)
- [Audience on page xii](#)
- [Supported Routing Platforms on page xii](#)
- [Document Conventions on page xii](#)
- [Documentation Feedback on page xiv](#)
- [Requesting Technical Support on page xiv](#)

J Series and SRX Series Documentation and Release Notes

For a list of related J Series documentation, see
<http://www.juniper.net/techpubs/software/junos-jseries/index-main.html> .

For a list of related SRX Series documentation, see
<http://www.juniper.net/techpubs/hardware/srx-series-main.html> .

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

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

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

Objectives

This guide describes how to use and configure key security features on J Series Services Routers and SRX Series Services Gateways running Junos OS. It provides conceptual information, suggested workflows, and examples where applicable.

Audience

This manual is designed for anyone who installs, sets up, configures, monitors, or administers a J Series Services Router or an SRX Series Services Gateway running Junos OS. The manual is intended for the following audiences:

- Customers with technical knowledge of and experience with networks and network security, the Internet, and Internet routing protocols
- Network administrators who install, configure, and manage Internet routers

Supported Routing Platforms

This manual describes features supported on J Series Services Routers and SRX Series Services Gateways running Junos OS.

Document Conventions

Table 1 on page xii defines the 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 xiii 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
<i>Italic text like this</i>	<ul style="list-style-type: none"> Introduces important new terms. Identifies book names. Identifies RFC and Internet draft titles. 	<ul style="list-style-type: none"> A policy <i>term</i> is a named structure that defines match conditions and actions. <i>Junos OS System Basics Configuration Guide</i> RFC 1997, <i>BGP Communities Attribute</i>
<i>Italic text like this</i>	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 <i>domain-name</i>
Text like this	Represents names of configuration statements, commands, files, and directories; interface names; configuration hierarchy levels; or labels on routing platform components.	<ul style="list-style-type: none"> 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 <i>metric</i> >;
(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 (<i>string1</i> <i>string2</i> <i>string3</i>)
# (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 [<i>community-ids</i>]
Indentation and braces ({ })	Identify a level in the configuration hierarchy.	[edit] routing-options { static { route default { nexthop <i>address</i> ; retain; } } }
;(semicolon)	Identifies a leaf statement at a configuration hierarchy level.	

J-Web GUI Conventions

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

Convention	Description	Examples
Bold text like this	Represents J-Web graphical user interface (GUI) items you click or select.	<ul style="list-style-type: none"> In the Logical Interfaces box, select All Interfaces. To cancel the configuration, click Cancel.
> (bold right angle bracket)	Separates levels in a hierarchy of J-Web selections.	In the configuration editor hierarchy, select Protocols>Ospf .

Documentation Feedback

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

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

Requesting Technical Support

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

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the JTAC User Guide located at <http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.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/>
- Find product documentation: <http://www.juniper.net/techpubs/>

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

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

Opening a Case with JTAC

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

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

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

PART 1

MPLS

- [MPLS on page 3](#)
- [Traffic Engineering on page 13](#)
- [MPLS VPNs on page 47](#)
- [CLNS VPNs on page 65](#)
- [VPLS on page 81](#)

CHAPTER 1

MPLS

- [MPLS Overview on page 3](#)
- [MPLS Configuration Overview on page 8](#)
- [Example: Deleting Security Services on page 9](#)
- [Example: Enabling MPLS on page 10](#)

MPLS Overview

Multiprotocol Label Switching (MPLS) is a method for engineering traffic patterns by assigning short labels to network packets that describe how to forward them through the network. MPLS is independent of routing tables or any routing protocol and can be used for unicast packets.

The MPLS framework supports traffic engineering and the creation of virtual private networks (VPNs). Traffic is engineered (controlled) primarily by the use of signaling protocols to establish label-switched paths (LSPs). VPN support includes Layer 2 and Layer 3 VPNs and Layer 2 circuits.

When you enable your device to allow MPLS traffic, the device performs packet-based processing and functions as a standard Junos router.



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

<http://www.juniper.net/us/en/local/pdf/app-notes/3500192-en.pdf>

This overview contains the following topics:

- [Label Switching on page 4](#)
- [Label-Switched Paths on page 4](#)
- [Label-Switching Routers on page 5](#)

- [Labels on page 6](#)
- [Label Operations on page 6](#)
- [Penultimate Hop Popping on page 7](#)
- [LSP Establishment on page 7](#)

Label Switching

In a traditional IP network, packets are transmitted with an IP header that includes a source and destination address. When a router receives such a packet, it examines its forwarding tables for the next-hop address associated with the packet's destination address and forwards the packet to the next-hop location.

In an MPLS network, each packet is encapsulated with an MPLS header. When a router receives the packet, it copies the header as an index into a separate MPLS forwarding table. The MPLS forwarding table consists of pairs of inbound interfaces and path information. Each pair includes forwarding information that the router uses to forward the traffic and modify, when necessary, the MPLS header.

Because the MPLS forwarding table has far fewer entries than the more general forwarding table, the lookup consumes less processing time and processing power. The resultant savings in time and processing are a significant benefit for traffic that uses the network to transit between outside destinations only.

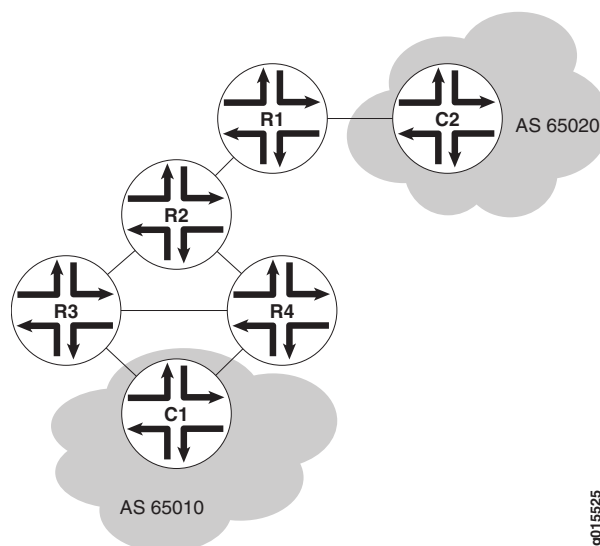
Label-Switched Paths

Label-switched paths (LSPs) are unidirectional routes through a network or autonomous system (AS). In normal IP routing, the packet has no predetermined path. Instead, each router forwards a packet to the next-hop address stored in its forwarding table, based only on the packet's destination address. Each subsequent router then forwards the packet using its own forwarding table.

In contrast, MPLS routers within an AS determine paths through a network through the exchange of MPLS traffic engineering information. Using these paths, the routers direct traffic through the network along an established route. Rather than selecting the next hop along the path as in IP routing, each router is responsible for forwarding the packet to a predetermined next-hop address.

[Figure 1 on page 5](#) shows a typical LSP topology.

Figure 1: Typical LSP Topology



In the topology shown in [Figure 1 on page 5](#), traffic is forwarded from Host C1 to the transit network with standard IP forwarding. When the traffic enters the transit network, it is switched across a preestablished LSP through the network. In this example, an LSP might switch the traffic from Router R4 to Router R2 to Router R1. When the traffic exits the network, it is forwarded to its destination by IP routing protocols.

Label-Switching Routers

Routers that are part of the LSP are label-switching routers (LSRs). Each LSR must be configured with MPLS so that it can interpret MPLS headers and perform the MPLS operations required to pass traffic through the network. An LSP can include four types of LSRs:

- Inbound router—The only entry point for traffic into MPLS. Native IPv4 packets are encapsulated into the MPLS protocol by the inbound router. Each LSP can have only one inbound router. Inbound routers are also known as ingress routers.
- Transit router—Any router in the middle of an LSP. An individual LSP can contain between 0 and 253 transit routers. Transit routers forward MPLS traffic along the LSP, using only the MPLS header to determine how the packet is routed.
- Penultimate router—The second-to-last router in the LSP. The penultimate router in an LSP is responsible for stripping the MPLS header from the packet before forwarding it to the outbound router.
- Outbound router—The endpoint for the LSP. The outbound router receives MPLS packets from the penultimate router and performs an IP route lookup. The router then forwards the packet to the next hop of the route. Each LSP can have only one outbound router. Outbound routers are also known as egress routers.

Labels

To forward traffic through an MPLS network, MPLS routers encapsulate packets and assign and manage headers known as *labels*. A label is a 20-bit unsigned integer in the range 0 through 1,048,575. The routers use the labels to index the MPLS forwarding tables that determine how packets are routed through the network.

When a network's inbound router receives traffic, it inserts an MPLS label between the IP packet and the appropriate Layer 2 header for the physical link. The label contains an index value that identifies a next-hop address for the particular LSP. When the next-hop transit router receives the packet, it uses the index in the MPLS label to determine the next-hop address for the packet and forwards the packet to the next router in the LSP.

As each packet travels through the transit network, every router along the way performs a lookup on the MPLS label and forwards the packet accordingly. When the outbound router receives a packet, it examines the header to determine that it is the final router in the LSP. The outbound router then removes the MPLS header, performs a regular IP route lookup, and forwards the packet with its IP header to the next-hop address.

Label Operations

Each LSR along an LSP is responsible for examining the MPLS label, determining the LSP next hop, and performing the required label operations. LSRs can perform five label operations:

- **Push**—Adds a new label to the top of the packet. For IPv4 packets arriving at the inbound router, the new label is the first label in the label stack. For MPLS packets with an existing label, this operation adds a label to the stack and sets the stacking bit to 0, indicating that more MPLS labels follow the first.

When it receives the packet, the inbound router performs an IP route lookup on the packet. Because the route lookup yields an LSP next hop, the inbound router performs a label push on the packet, and then forwards the packet to the LSP next hop.

- **Swap**—Replaces the label at the top of the label stack with a new label.

When a transit router receives the packet, it performs an MPLS forwarding table lookup. The lookup yields the LSP next hop and the path index of the link between the transit router and the next router in the LSP.

- **Pop**—Removes the label from the top of the label stack. For IPv4 packets arriving at the penultimate router, the entire MPLS label is removed from the label stack. For MPLS packets with an existing label, this operation removes the top label from the label stack and modifies the stacking bit as necessary—sets it to 1, for example, if only a single label remains in the stack.

If multiple LSPs terminate at the same outbound router, the router performs MPLS label operations for all outbound traffic on the LSPs. To share the operations among multiple routers, most LSPs use penultimate hop popping (PHP).

- **Multiple push**—Adds multiple labels to the top of the label stack. This action is equivalent to performing multiple push operations.

The multiple push operation is used with label stacking, which is beyond the scope of this topic.

- Swap and push—Replaces the top label with a new label and then pushes a new label to the top of the stack.

The swap and push operation is used with label stacking, which is beyond the scope of this topic.

Penultimate Hop Popping

Multiple LSPs terminating at a single outbound router load the router with MPLS label operations for all their outbound traffic. Penultimate hop popping (PHP) transfers the operation from the outbound router to penultimate routers.

With PHP, the penultimate router is responsible for popping the MPLS label and forwarding the traffic to the outbound router. The outbound router then performs an IP route lookup and forwards the traffic. For example, if four LSPs terminate at the same outbound router and each has a different penultimate router, label operations are shared across four routers.

LSP Establishment

An MPLS LSP is established by one of two methods: static LSPs and dynamic LSPs.

Static LSPs

Like a static route, a static LSP requires each router along the path to be configured explicitly. You must manually configure the path and its associated label values. Static LSPs require less processing by the LSRs because no signaling protocol is used. However, because paths are statically configured, they cannot adapt to network conditions. Topology changes and network outages can create black holes in the LSP that exist until you manually reconfigure the LSP.

Dynamic LSPs

Dynamic LSPs use signaling protocols to establish themselves and propagate LSP information to other LSRs in the network. You configure the inbound router with LSP information that is transmitted throughout the network when you enable the signaling protocols across the LSRs. Because the LSRs must exchange and process signaling packets and instructions, dynamic LSPs consume more resources than static LSPs. However, dynamic LSPs can avoid the network black holes of static LSPs by detecting topology changes and outages and propagating them throughout the network.

Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [MPLS Configuration Overview on page 8](#)
- [Example: Deleting Security Services on page 9](#)
- [Example: Enabling MPLS on page 10](#)
- [MPLS Traffic Engineering and Signaling Protocols Overview on page 13](#)
- [MPLS VPN Overview on page 47](#)

MPLS Configuration Overview

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

To enable MPLS:

1. Delete all configured security services from the device. If you do not complete this step, you will get a commit failure. See [“Example: Deleting Security Services” on page 9](#).
2. Enable MPLS on the device. See [“Example: Enabling MPLS” on page 10](#).
3. Commit the configuration.
4. Reboot the device.
5. Configure MPLS features such as traffic engineering, VPNs, and VPLS. See:
 - [MPLS Traffic Engineering and Signaling Protocols Overview on page 13](#)
 - [MPLS VPN Overview on page 47](#)
 - [CLNS Overview on page 65](#)
 - [VPLS Overview on page 81](#)



CAUTION: When packet forwarding mode is changed to MPLS, all flow-based security features are deactivated, and the device performs packet-based processing only. Flow-based services such as security policies, zones, NAT, ALGs, chassis clustering, screens, firewall authentication, and IPsec VPNs are unavailable on the device. However, MPLS can be enabled in flow-based packet forwarding mode for selected traffic using firewall filters. For more information, please see the following application note:
<http://www.juniper.net/us/en/local/pdf/app-notes/3500192-en.pdf>

Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Junos OS MPLS Applications Configuration Guide](#)
- [Junos OS VPNs Configuration Guide](#)
- [MPLS Overview on page 3](#)
- [Example: Deleting Security Services on page 9](#)
- [Example: Enabling MPLS on page 10](#)

Example: Deleting Security Services

This example shows how to delete configured services in the security level of the configuration hierarchy.

- [Requirements on page 9](#)
- [Overview on page 9](#)
- [Configuration on page 9](#)
- [Verification on page 9](#)

Requirements

Before you begin, save your current configuration to a temporary file. Do this prior to removing all configurations from the security level of the configuration hierarchy and deleting the inherited configurations.

Overview

In this example, you save your current configuration in the `var/tmp/` directory with an appropriate filename and `.cfg` extension—for example, `curfeb08.cfg`. Then you remove all configurations from the **security** level of the configuration hierarchy, and delete all global groups and inherited configurations.

Configuration

Step-by-Step Procedure

To delete the configured services in the security level of the configuration hierarchy:

1. Save your current configuration.

```
[edit]
user@host# save /var/tmp/curfeb08.cfg
```
2. Remove all configurations in the **security** level of the configuration hierarchy.

```
[edit]
user@host# delete security
```
3. Remove all inherited configurations in the security level of the configuration hierarchy.

```
[edit]
user@host# delete groups global security
```



CAUTION: Do not commit after deleting the security configurations. A commit without any security configurations leaves the router unreachable through the management port.

Verification

To verify the configuration is working properly, enter the **show groups global security** command.

Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [MPLS Overview on page 3](#)
- [MPLS Configuration Overview on page 8](#)
- [Example: Enabling MPLS on page 10](#)

Example: Enabling MPLS

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

Requirements

Before you begin, delete all configured security services. See [“Example: Deleting Security Services” on page 9](#).

Overview

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

Configuration

CLI Quick Configuration

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

```
set security forwarding-options family mpls mode packet-based
set interfaces ge-1/0/0 unit 0 family mpls
set protocols mpls ge-1/0/0 unit 0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see [Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*](#).

To enable MPLS:

1. Enable MPLS for packet-based processing.

```
[edit security forwarding-options]
user@host# set family mpls mode packet-based
```
2. Enable the MPLS family on each transit interface that you want to include in the MPLS network.

```
[edit interfaces]
user@host# set interfaces ge-1/0/0 unit 0 family mpls
```
3. Enable the MPLS process on all of the transit interfaces in the MPLS network.

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

Results From configuration mode, confirm your configuration by entering the **show security forwarding-options** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.



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

warning: Reboot may required when try reset flow inet mode

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

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



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

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

For information about configuring zones, see [Junos OS Security Configuration Guide](#).



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

```
[edit]
user@host# show security forwarding-options
family {
  mpls {
    mode packet-based;
  }
}
```

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

Verification

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

- [Verifying MPLS Is Enabled at the Protocols Level on page 12](#)
- [Verifying MPLS Is Enabled at the Interfaces Level on page 12](#)

Verifying MPLS Is Enabled at the Protocols Level

Purpose Verify that MPLS is enabled at the protocols level.

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

Verifying MPLS Is Enabled at the Interfaces Level

Purpose Verify that MPLS is enabled at the interfaces level.

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

Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [MPLS Overview on page 3](#)
- [MPLS Configuration Overview on page 8](#)
- [Example: Deleting Security Services on page 9](#)

CHAPTER 2

Traffic Engineering

- [MPLS Traffic Engineering and Signaling Protocols Overview on page 13](#)
- [LDP Signaling Protocol on page 14](#)
- [RSVP Signaling Protocol on page 18](#)
- [Example: Configuring Point-to-Multipoint LSPs with Static Routes on page 26](#)

MPLS Traffic Engineering and Signaling Protocols Overview

Traffic engineering facilitates efficient and reliable network operations while simultaneously optimizing network resources and traffic performance. Traffic engineering provides the ability to move traffic flow away from the shortest path selected by the interior gateway protocol (IGP) to a potentially less congested physical path across a network. To support traffic engineering, besides source routing, the network must do the following:

- Compute a path at the source by taking into account all the constraints, such as bandwidth and administrative requirements.
- Distribute the information about network topology and link attributes throughout the network once the path is computed.
- Reserve network resources and modify link attributes.

When transit traffic is routed through an IP network, MPLS is often used to engineer its passage. Although the exact path through the transit network is of little importance to either the sender or the receiver of the traffic, network administrators often want to route traffic more efficiently between certain source and destination address pairs. By adding a short label with specific routing instructions to each packet, MPLS switches packets from router to router through the network rather than forwarding packets based on next-hop lookups. The resulting routes are called *label-switched paths (LSPs)*. LSPs control the passage of traffic through the network and speed traffic forwarding.

You can create LSPs manually, or through the use of signaling protocols. Signaling protocols are used within an MPLS environment to establish LSPs for traffic across a transit network. Junos OS supports two signaling protocols—LDP and the Resource Reservation Protocol (RSVP).

MPLS traffic engineering uses the following components:

- MPLS LSPs for packet forwarding
- IGP extensions for distributing information about the network topology and link attributes
- Constrained Shortest Path First (CSPF) for path computation and path selection
- RSVP extensions to establish the forwarding state along the path and to reserve resources along the path

Junos OS also supports traffic engineering across different OSPF regions.

**Related
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Junos OS MPLS Applications Configuration Guide](#)
- [Understanding the LDP Signaling Protocol on page 14](#)
- [Understanding the RSVP Signaling Protocol on page 19](#)
- [Understanding Point-to-Multipoint LSPs on page 26](#)

LDP Signaling Protocol

- [Understanding the LDP Signaling Protocol on page 14](#)
- [Example: Configuring LDP-Signaled LSPs on page 15](#)

Understanding the LDP Signaling Protocol

LDP is a signaling protocol that runs on a device configured for MPLS support. The successful configuration of both MPLS and LDP initiates the exchange of TCP packets across the LDP interfaces. The packets establish TCP-based LDP sessions for the exchange of MPLS information within the network. Enabling both MPLS and LDP on the appropriate interfaces is sufficient to establish LSPs.

LDP is a simple, fast-acting signaling protocol that automatically establishes LSP adjacencies within an MPLS network. Routers then share LSP updates such as hello packets and LSP advertisements across the adjacencies. Because LDP runs on top of an IGP such as IS-IS or OSPF, you must configure LDP and the IGP on the same set of interfaces. After both are configured, LDP begins transmitting and receiving LDP messages through all LDP-enabled interfaces. Because of LDP's simplicity, it cannot perform the true traffic engineering which RSVP can perform. LDP does not support bandwidth reservation or traffic constraints.

When you configure LDP on a label-switching router (LSR), the router begins sending LDP discovery messages out all LDP-enabled interfaces. When an adjacent LSR receives LDP discovery messages, it establishes an underlying TCP session. An LDP session is then created on top of the TCP session. The TCP three-way handshake ensures that the LDP session has bidirectional connectivity. After they establish the LDP session, the LDP neighbors maintain, and terminate, the session by exchanging messages. LDP advertisement messages allow LSRs to exchange label information to determine the

next hops within a particular LSP. Any topology changes, such as a router failure, generate LDP notifications that can terminate the LDP session or generate additional LDP advertisements to propagate an LSP change.

Example: Configuring LDP-Signaled LSPs

This example shows how to create and configure LDP instances within an MPLS network.

- [Requirements on page 15](#)
- [Overview on page 15](#)
- [Configuration on page 15](#)
- [Verification on page 16](#)

Requirements

Before you begin:

- Configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
- Configure an IGP across your network. (The LDP configuration is added to the existing IGP configuration and included in the MPLS configuration.)
- Configure a network to use LDP for LSP establishment by enabling MPLS on all transit interfaces in the MPLS network.



NOTE: Because LDP runs on top of an IGP such as IS-IS or OSPF, you must configure LDP and the IGP on the same set of interfaces.

Overview

To configure LDP-signaled RSPs, you must enable the MPLS family on all transit interfaces in the MPLS network, enable the MPLS process on all router interfaces in the MPLS network, and enable an LDP instance on each router. In this example, you enable the MPLS family and create an LDP instance on the ge-0/0/0 interface. Additionally, you enable the MPLS process on all router interfaces in the MPLS network.

Configuration

CLI Quick Configuration

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

```
set interfaces ge-0/0/0 unit 0 family mpls
set protocols mpls ge-0/0/0 unit 0
set protocols ldp interface ge-0/0/0.0 unit 0
```

Step-by-Step Procedure

To enable LDP instances within an MPLS network:

1. Enable the MPLS family on the transit interface.

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

2. Enable the MPLS process on the transit interface.

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

3. Create the LDP instance on the transit interface.

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

Results Confirm your configuration by entering the **show** command from configuration mode. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show** output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
user@host# show
...
interfaces {
  ge-0/0/0 {
    unit 0 {
      family inet {
        address 10.100.37.20/24;
      }
      family mpls;
    }
  }
}
...
protocols {
  mpls {
    interface all;
  }
  ldp {
    interface ge-0/0/0.0;
  }
}
```

If you are done configuring the device, enter the **commit** command from the configuration mode to activate the configuration.

Verification

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

- [Verifying LDP Neighbors on page 17](#)
- [Verifying LDP Sessions on page 17](#)
- [Verifying the Presence of LDP-Signaled LSPs on page 18](#)
- [Verifying Traffic Forwarding over the LDP-Signaled LSP on page 18](#)

Verifying LDP Neighbors

Purpose Verify that each router shows the appropriate LDP neighbors.

Action From the CLI, enter the **show ldp neighbor** command.

```
user@r5> show ldp neighbor
Address      Interface      Label space ID  Hold time
10.0.8.5     ge-0/0/0.0     10.0.9.6:0      14
10.0.8.10    ge-0/0/1.0     10.0.9.7:0      11
```

The output shows the IP addresses of the neighboring interfaces along with the interface through which the neighbor adjacency is established. Verify the following information:

- Each interface on which LDP is enabled is listed.
- Each neighboring LDP interface address is listed with the appropriate corresponding LDP interface.
- Under **Label space ID**, the appropriate loopback address for each neighbor appears.

Verifying LDP Sessions

Purpose Verify that a TCP-based LDP session has been established between all LDP neighbors. Also, verify that the modified keepalive value is active.

Action From the CLI, enter the **show ldp session detail** command.

```
user@r5> show ldp session detail
Address: 10.0.9.7, State: Operational, Connection: Open, Hold time: 28
Session ID: 10.0.3.5:0--10.0.9.7:0
Next keepalive in 3 seconds
Passive, Maximum PDU: 4096, Hold time: 30, Neighbor count: 1
Keepalive interval: 10, Connect retry interval: 1
Local - Restart: disabled, Helper mode: enabled
Remote - Restart: disabled, Helper mode: disabled
Local maximum recovery time: 240000 msec
Next-hop addresses received:
  10.0.8.10
  10.0.2.17
```

The output shows the detailed information, including session IDs, keepalive interval, and next-hop addresses, for each established LDP session. Verify the following information:

- Each LDP neighbor address has an entry, listed by loopback address.
- The state for each session is **Operational**, and the connection for each session is **Open**. A state of **Nonexistent** or a connection of **Closed** indicates a problem with one of the following:
 - LDP configuration
 - Passage of traffic between the two devices
 - Physical link between the two routers
- For **Keepalive interval**, the appropriate value, **10**, appears.

Verifying the Presence of LDP-Signaled LSPs

Purpose Verify that each Juniper Networks device's **inet.3** routing table has an LSP for the loopback address on each of the other routers.

Action From the CLI, enter the **show route table inet.3** command.

```
user@r5> show route table inet.3
inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.9.6/32          *[LDP/9/0] 00:05:29, metric 1
> to 10.0.8.5 via ge-0/0/0.0
10.0.9.7/32          *[LDP/9/0] 00:05:37, metric 1
> to 10.0.8.10 via ge-0/0/1.0
```

The output shows the LDP routes that exist in the **inet.3** routing table. Verify that an LDP-signaled LSP is associated with the loopback addresses of the other routers in the MPLS network.

Verifying Traffic Forwarding over the LDP-Signaled LSP

Purpose Verify that traffic between hosts is forwarded over the LDP-signaled LSP. Because traffic uses any configured gateway address by default, you must explicitly specify that the gateway address is to be bypassed.

Action From the CLI, enter the **traceroute 220.220.0.0 source 200.200.0.1 bypass-routing gateway 172.16.0.1** command.

```
user@c1> traceroute 220.220.0.0 source 200.200.0.1 bypass-routing gateway 172.16.0.1
traceroute to 220.220.0.1 (172.16.0.1) from 200.200.0.1, 30 hops max, 40 byte
packets
 1  172.16.0.1 (172.16.0.1)  0.661 ms  0.538 ms  0.449 ms
 2  10.0.8.9 (10.0.8.9)  0.511 ms  0.479 ms  0.468 ms
    MPLS Label=100004 CoS=0 TTL=1 S=1
 3  10.0.8.5 (10.0.8.5)  0.476 ms  0.512 ms  0.441 ms
 4  220.220.0.1 (220.220.0.1)  0.436 ms  0.420 ms  0.416 ms
```

The output shows the route that traffic travels between hosts without using the default gateway. In this example, verify that traffic sent from Host C1 to Host C2 travels through Router R7. The **10.0.8.9** address is the interface address for Router R5.

Related Documentation

- [MPLS Traffic Engineering and Signaling Protocols Overview on page 13](#)
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Junos OS Routing Protocols and Policies Configuration Guide for Security Devices](#)
- [Junos OS Routing Protocols and Policies Command Reference](#)

RSVP Signaling Protocol

- [Understanding the RSVP Signaling Protocol on page 19](#)
- [Example: Configuring RSVP-Signaled LSPs on page 22](#)

Understanding the RSVP Signaling Protocol

RSVP is a signaling protocol that handles bandwidth allocation and true traffic engineering across an MPLS network. Like LDP, RSVP uses discovery messages and advertisements to exchange LSP path information between all hosts. However, RSVP also includes a set of features that control the flow of traffic through an MPLS network. Whereas LDP is restricted to using the configured IGP's shortest path as the transit path through the network, RSVP uses a combination of the Constrained Shortest Path First (CSPF) algorithm and Explicit Route Objects (EROs) to determine how traffic is routed through the network.

Basic RSVP sessions are established in exactly the same way that LDP sessions are established. By configuring both MPLS and RSVP on the appropriate transit interfaces, you enable the exchange of RSVP packets and the establishment of LSPs. However, RSVP also lets you configure link authentication, explicit LSP paths, and link coloring.

This topic contains the following sections:

- [RSVP Fundamentals on page 19](#)
- [Bandwidth Reservation Requirement on page 19](#)
- [Explicit Route Objects on page 20](#)
- [Constrained Shortest Path First on page 21](#)
- [Link Coloring on page 21](#)

RSVP Fundamentals

RSVP uses unidirectional and simplex flows through the network to perform its function. The inbound router initiates an RSVP path message and sends it downstream to the outbound router. The path message contains information about the resources needed for the connection. Each router along the path begins to maintain information about a reservation.

When the path message reaches the outbound router, resource reservation begins. The outbound router sends a reservation message upstream to the inbound router. Each router along the path receives the reservation message and sends it upstream, following the path of the original path message. When the inbound router receives the reservation message, the unidirectional network path is established.

The established path remains open as long as the RSVP session is active. The session is maintained by the transmission of additional path and reservation messages that report the session state every 30 seconds. If a router does not receive the maintenance messages for 3 minutes, it terminates the RSVP session and reroutes the LSP through another active router.

Bandwidth Reservation Requirement

When a bandwidth reservation is configured, reservation messages propagate the bandwidth value throughout the LSP. Routers must reserve the bandwidth specified across the link for the particular LSP. If the total bandwidth reservation exceeds the available bandwidth for a particular LSP segment, the LSP is rerouted through another

LSR. If no segments can support the bandwidth reservation, LSP setup fails and the RSVP session is not established.

Explicit Route Objects

Explicit Route Objects (EROs) limit LSP routing to a specified list of LSRs. By default, RSVP messages follow a path that is determined by the network IGP's shortest path. However, in the presence of a configured ERO, the RSVP messages follow the path specified.

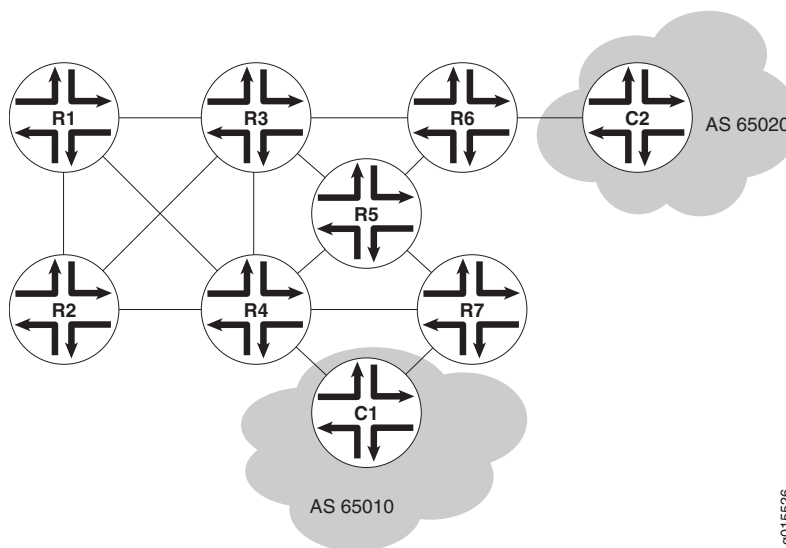
EROs consist of two types of instructions: loose hops and strict hops. When a loose hop is configured, it identifies one or more transit LSRs through which the LSP must be routed. The network IGP determines the exact route from the inbound router to the first loose hop, or from one loose hop to the next. The loose hop specifies only that a particular LSR be included in the LSP.

When a strict hop is configured, it identifies an exact path through which the LSP must be routed. Strict-hop EROs specify the exact order of the routers through which the RSVP messages are sent.

You can configure loose-hop and strict-hop EROs simultaneously. In this case, the IGP determines the route between loose hops, and the strict-hop configuration specifies the exact path for particular LSP path segments.

Figure 2 on page 20 shows a typical RSVP-signaled LSP that uses EROs.

Figure 2: Typical RSVP-Signaled LSP with EROs



In the topology shown in Figure 2 on page 20, traffic is routed from Host C1 to Host C2. The LSP can pass through Routers R4 or Router R7. To force the LSP to use R4, you can set up either a loose-hop or strict-hop ERO that specifies R4 as a hop in the LSP. To force a specific path through Router R4, R3, and R6, configure a strict-hop ERO through the three LSRs.

Constrained Shortest Path First

Whereas IGPs use the Shortest Path First (SPF) algorithm to determine how traffic is routed within a network, RSVP uses the Constrained Shortest Path First (CSPF) algorithm to calculate traffic paths that are subject to the following constraints:

- LSP attributes—Administrative groups such as link coloring, bandwidth requirements, and EROs
- Link attributes—Colors on a particular link and available bandwidth

These constraints are maintained in the traffic engineering database (TED). The database provides CSPF with up-to-date topology information, the current reservable bandwidth of links, and the link colors.

In determining which path to select, CSPF follows these rules:

- Computes LSPs one at a time, beginning with the highest priority LSP—the one with the lowest setup priority value. Among LSPs of equal priority, CSPF starts with those that have the highest bandwidth requirement.
- Prunes the traffic engineering database of links that are not full duplex and do not have sufficient reservable bandwidth.
- If the LSP configuration includes the **include** statement, prunes all links that do not share any included colors.
- If the LSP configuration includes the **exclude** statement, prunes all links that contain excluded colors. If a link does not have a color, it is accepted.
- Finds the shortest path toward the LSP's outbound router, taking into account any EROs. For example, if the path must pass through Router A, two separate SPF algorithms are computed: one from the inbound router to Router A and one from Router A to the outbound router.
- If several paths have equal cost, chooses the one with a last-hop address the same as the LSP's destination.
- If several equal-cost paths remain, selects the path with the least number of hops.
- If several equal-cost paths remain, applies CSPF load-balancing rules configured on the LSP.

Link Coloring

RSVP allows you to configure administrative groups for CSPF path selection. An administrative group is typically named with a color, assigned a numeric value, and applied to the RSVP interface for the appropriate link. Lower numbers indicate higher priority.

After configuring the administrative group, you can either exclude, include, or ignore links of that color in the TED:

- If you exclude a particular color, all segments with an administrative group of that color are excluded from CSPF path selection.

- If you include a particular color, only those segments with the appropriate color are selected.
- If you neither exclude nor include the color, the metrics associated with the administrative groups and applied on the particular segments determine the path cost for that segment.

The LSP with the lowest total path cost is selected into the TED.

Example: Configuring RSVP-Signaled LSPs

This example shows how to create an LSP between routers in an IP network using RSVP as the signaling protocol.

- [Requirements on page 22](#)
- [Overview and Topology on page 22](#)
- [Configuration on page 23](#)
- [Verification on page 24](#)

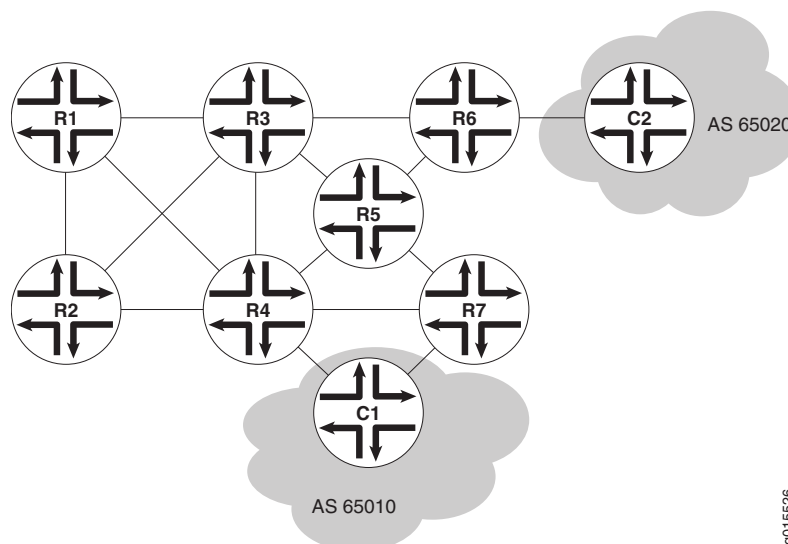
Requirements

Before you begin, delete security services from the device. See [“Example: Deleting Security Services” on page 9](#).

Overview and Topology

Using RSVP as a signaling protocol, you can create LSPs between routers in an IP network. In this example, you configure a sample network as shown in [Figure 3 on page 22](#).

Figure 3: Typical RSVP-Signaled LSP



To establish an LSP between routers, you must individually enable the MPLS family and configure RSVP on each of the transit interfaces in the MPLS network. This example shows how to enable MPLS and configure RSVP on the ge-0/0/0 transit interface.

Additionally, you must enable the MPLS process on all of the MPLS interfaces in the network.

This example shows how to define an LSP from R1 to R7 on the ingress router (R1) using R7's loopback address (10.0.9.7). The configuration reserves 10 Mbps of bandwidth. Additionally, the configuration disables the CSPF algorithm, ensuring that Hosts C1 and C2 use the RSVP-signaled LSP that correspond to the network IGP's shortest path.

Configuration

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

```
set interfaces ge-0/0/0 unit 0 family mpls
set protocols rsvp interface ge-0/0/0.0
set protocols mpls label-switched-path r1-r7 to 10.0.9.7
set protocols mpls label-switched-path r1-r7 bandwidth 10m
set protocols mpls interface all
```

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

To configure RSVP:

1. Enable the MPLS family on all transit interfaces in the MPLS network.

```
[edit]
user@host# set interfaces ge-0/0/0 unit 0 family mpls
```
2. Enable RSVP on each transit interface in the MPLS network.

```
[edit]
user @host# set protocols rsvp interface ge-0/0/0
```
3. Enable the MPLS process on the transit interface in the MPLS network.

```
[edit]
user@host# set protocols mpls interface ge-0/0/0
```
4. Define the LSP on the ingress router.

```
[edit protocols mpls]
user@host# set label-switched-path r1-r7 to 10.0.9.7
```
5. Reserve 10 Mbps of bandwidth on the LSP.

```
[edit protocols mpls]
user @host# set label-switched-path r1-r7 bandwidth 10m
```

Results Confirm your configuration by entering the **show** command from configuration mode. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show** command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
user@host# show
...
interfaces {
  ge-0/0/0 {
    family mpls;
  }
}
...
protocols {
  rsvp {
    interface ge-0/0/0.0;
  }
  mpls {
    label-switched-path r1-r7 {
      to 10.0.9.7;
      bandwidth 10m;
    }
    interface all;
  }
}
...
```

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

Verification

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

- [Verifying RSVP Neighbors on page 24](#)
- [Verifying RSVP Sessions on page 25](#)
- [Verifying the Presence of RSVP-Signaled LSPs on page 25](#)

Verifying RSVP Neighbors

Purpose Verify that each device shows the appropriate RSVP neighbors—for example, that Router R1 in [Figure 3 on page 22](#) lists both Router R3 and Router R2 as RSVP neighbors.

Action From the CLI, enter the **show rsvp neighbor** command.

```
user@r1> show rsvp neighbor
RSVP neighbor: 2 learned
Address           Idle Up/Dn LastChange HelloInt HelloTx/Rx
10.0.6.2           0 3/2      13:01      3    366/349
10.0.3.3           0 1/0      22:49      3    448/448
```

The output shows the IP addresses of the neighboring routers. Verify that each neighboring RSVP router loopback address is listed.

Verifying RSVP Sessions

Purpose Verify that an RSVP session has been established between all RSVP neighbors. Also, verify that the bandwidth reservation value is active.

Action From the CLI, enter the **show rsvp session detail** command.

```
user@r1> show rsvp session detail
Ingress RSVP: 1 sessions

10.0.9.7
  From: 10.0.6.1, LSPstate: Up, ActiveRoute: 0
  LSPname: r1-r7, LSPpath: Primary
  Bidirectional, Upstream label in: -, Upstream label out: -
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 100000
  Resv style: 1 FF, Label in: -, Label out: 100000
  Time left: -, Since: Thu Jan 26 17:57:45 2002
  Tspec: rate 10Mbps size 10Mbps peak Infbps m 20 M 1500
  Port number: sender 3 receiver 17 protocol 0
  PATH rcvfrom: localclient
  PATH sentto: 10.0.4.13 (ge-0/0/1.0) 1467 pkts
  RESV rcvfrom: 10.0.4.13 (ge-0/0/1.0) 1467 pkts
  Record route: <self> 10.0.4.13 10.0.2.1 10.0.8.10
```

The output shows the detailed information, including session IDs, bandwidth reservation, and next-hop addresses, for each established RSVP session. Verify the following information:

- Each RSVP neighbor address has an entry for each neighbor, listed by loopback address.
- The state for each LSP session is **Up**.
- For **Tspec**, the appropriate bandwidth value, **10Mbps**, appears.

Verifying the Presence of RSVP-Signaled LSPs

Purpose Verify that the routing table of the entry (ingress) router has a configured LSP to the loopback address of the other router. For example, verify that the **inet.3** routing table of the R1 entry router in [Figure 3 on page 22](#) has a configured LSP to the loopback address of Router R7.

Action From the CLI, enter the **show route table inet.3** command.

```
user@r1> show route table inet.3
inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.9.7/32          *[RSVP/7] 00:05:29, metric 10
                    > to 10.0.4.17 via ge-0/0/0.0, label-switched-path r1-r7
```

The output shows the RSVP routes that exist in the **inet.3** routing table. Verify that an RSVP-signaled LSP is associated with the loopback address of the exit (egress) router, R7, in the MPLS network.

Example: Configuring Point-to-Multipoint LSPs with Static Routes

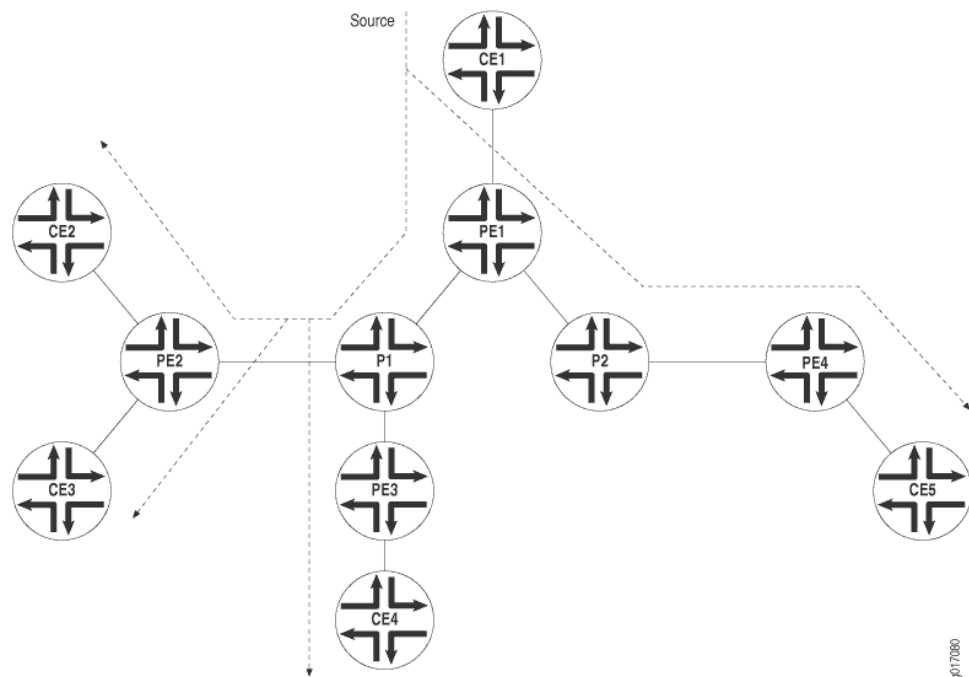
- [Understanding Point-to-Multipoint LSPs on page 26](#)
- [Point-to-Multipoint LSP Configuration Overview on page 27](#)
- [Example: Configuring an RSVP-Signaled Point-to-Multipoint LSP on page 27](#)

Understanding Point-to-Multipoint LSPs

A point-to-multipoint MPLS label-switched path (LSP) is an LDP-signaled or RSVP-signaled LSP with a single source and multiple destinations. By taking advantage of the MPLS packet replication capability of the network, point-to-multipoint LSPs avoid unnecessary packet replication at the inbound (ingress) router. Packet replication takes place only when packets are forwarded to two or more different destinations requiring different network paths.

This process is illustrated in [Figure 4 on page 26](#). Device PE1 is configured with a point-to-multipoint LSP to Routers PE2, PE3, and PE4. When Device PE1 sends a packet on the point-to-multipoint LSP to Routers P1 and P2, Device P1 replicates the packet and forwards it to Routers PE2 and PE3. Device P2 sends the packet to Device PE4.

Figure 4: Point-to-Multipoint LSPs



Following are some of the properties of point-to-multipoint LSPs:

- A point-to-multipoint LSP allows you to use MPLS for point-to-multipoint data distribution. This functionality is similar to that provided by IP multicast.
- You can add and remove branch LSPs from a main point-to-multipoint LSP without disrupting traffic. The unaffected parts of the point-to-multipoint LSP continue to function normally.
- You can configure a node to be both a transit and an outbound (egress) router for different branch LSPs of the same point-to-multipoint LSP.
- You can enable link protection on a point-to-multipoint LSP. Link protection can provide a bypass LSP for each of the branch LSPs that make up the point-to-multipoint LSP. If any primary paths fail, traffic can be quickly switched to the bypass.
- You can configure subpaths either statically or dynamically.
- You can enable graceful restart on point-to-multipoint LSPs.

Point-to-Multipoint LSP Configuration Overview

To set up a point-to-multipoint LSP:

1. Configure the primary LSP from the ingress router and the branch LSPs that carry traffic to the egress routers.
2. Specify a pathname on the primary LSP and this same path name on each branch LSP.



NOTE: By default, the branch LSPs are dynamically signaled by means of Constrained Shortest Path First (CSPF) and require no configuration. You can alternatively configure the branch LSPs as static paths.

Example: Configuring an RSVP-Signaled Point-to-Multipoint LSP

This example shows how to configure a collection of paths to create an RSVP-signaled point-to-multipoint label-switched path (LSP).

- [Requirements on page 27](#)
- [Overview on page 27](#)
- [Configuration on page 28](#)
- [Verification on page 44](#)

Requirements

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

Overview

In this example, multiple routing devices serve as the transit, branch, and leaf nodes of a single point-to-multipoint LSP. On the provider edge (PE), Device PE1 is the ingress

node. The branches go from PE1 to PE2, PE1 to PE3, and PE1 to PE4. Static unicast routes on the ingress node (PE1) point to the egress nodes.

This example also demonstrates static routes with a next hop that is a point-to-multipoint LSP, using the `p2mp-lsp-next-hop` statement. This is useful when implementing filter-based forwarding.

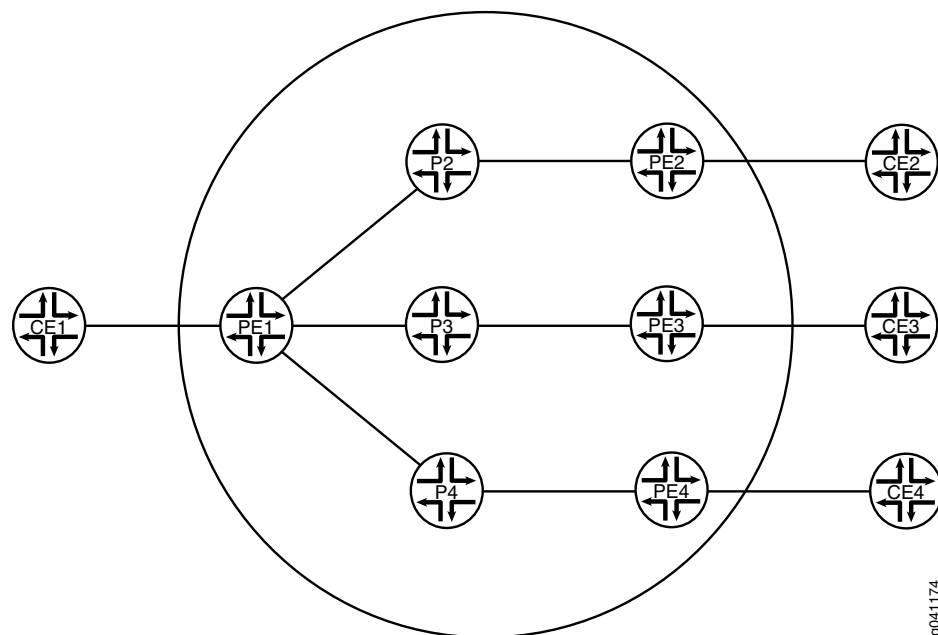


NOTE: Another option is to use the `lsp-next-hop` statement to configure a regular point-to-point LSP to be the next hop. Though not shown in this example, you can optionally assign an independent preference and metric to the next hop.

Topology Diagram

Figure 5 on page 28 shows the topology used in this example.

Figure 5: RSVP-Signaled Point-to-Multipoint LSP



g041174

Configuration

CLI Quick Configuration

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

Device PE1

```
set interfaces ge-2/0/2 unit 0 description PE1-to-CE1
set interfaces ge-2/0/2 unit 0 family inet address 10.0.244.10/30
set interfaces fe-2/0/10 unit 1 description PE1-to-P2
set interfaces fe-2/0/10 unit 1 family inet address 2.2.2.1/24
set interfaces fe-2/0/10 unit 1 family mpls
set interfaces fe-2/0/9 unit 8 description PE1-to-P3
```

```

set interfaces fe-2/0/9 unit 8 family inet address 6.6.6.1/24
set interfaces fe-2/0/9 unit 8 family mpls
set interfaces fe-2/0/8 unit 9 description PE1-to-P4
set interfaces fe-2/0/8 unit 9 family inet address 3.3.3.1/24
set interfaces fe-2/0/8 unit 9 family mpls
set interfaces lo0 unit 1 family inet address 100.10.10.10/32
set protocols rsvp interface fe-2/0/10.1
set protocols rsvp interface fe-2/0/9.8
set protocols rsvp interface fe-2/0/8.9
set protocols rsvp interface lo0.1
set protocols mpls traffic-engineering bgp-igp
set protocols mpls label-switched-path PE1-PE2 to 100.50.50.50
set protocols mpls label-switched-path PE1-PE2 link-protection
set protocols mpls label-switched-path PE1-PE2 p2mp p2mp1
set protocols mpls label-switched-path PE1-PE3 to 100.70.70.70
set protocols mpls label-switched-path PE1-PE3 link-protection
set protocols mpls label-switched-path PE1-PE3 p2mp p2mp1
set protocols mpls label-switched-path PE1-PE4 to 100.40.40.40
set protocols mpls label-switched-path PE1-PE4 link-protection
set protocols mpls label-switched-path PE1-PE4 p2mp p2mp1
set protocols mpls interface fe-2/0/10.1
set protocols mpls interface fe-2/0/9.8
set protocols mpls interface fe-2/0/8.9
set protocols mpls interface lo0.1
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-2/0/2.0
set protocols ospf area 0.0.0.0 interface fe-2/0/10.1
set protocols ospf area 0.0.0.0 interface fe-2/0/9.8
set protocols ospf area 0.0.0.0 interface fe-2/0/8.9
set protocols ospf area 0.0.0.0 interface lo0.1
set routing-options static route 5.5.5.0/24 p2mp-lsp-next-hop p2mp1
set routing-options static route 7.7.7.0/24 p2mp-lsp-next-hop p2mp1
set routing-options static route 4.4.4.0/24 p2mp-lsp-next-hop p2mp1
set routing-options router-id 100.10.10.10

```

Device CE1	<pre> set interfaces ge-1/3/2 unit 0 family inet address 10.0.244.9/30 set interfaces ge-1/3/2 unit 0 description CE1-to-PE1 set routing-options static route 10.0.104.8/30 next-hop 10.0.244.10 set routing-options static route 10.0.134.8/30 next-hop 10.0.244.10 set routing-options static route 10.0.224.8/30 next-hop 10.0.244.10 </pre>
Device CE2	<pre> set interfaces ge-1/3/3 unit 0 family inet address 10.0.224.9/30 set interfaces ge-1/3/3 unit 0 description CE2-to-PE2 set routing-options static route 10.0.244.8/30 next-hop 10.0.224.10 </pre>
Device CE3	<pre> set interfaces ge-2/0/1 unit 0 family inet address 10.0.134.9/30 set interfaces ge-2/0/1 unit 0 description CE3-to-PE3 set routing-options static route 10.0.244.8/30 next-hop 10.0.134.10 </pre>
Device CE4	<pre> set interfaces ge-3/1/3 unit 0 family inet address 10.0.104.10/30 set interfaces ge-3/1/3 unit 0 description CE4-to-PE4 set routing-options static route 10.0.244.8/30 next-hop 10.0.104.9 </pre>

*Configuring the Ingress Label-Switched Router (LSR) (Device PE1)***Step-by-Step
Procedure**

To configure Device PE1:

1. Configure the interfaces, interface encapsulation, and protocol families.

```
[edit interfaces]
user@PE1# set ge-2/0/2 unit 0 description PE1-to-CE1
user@PE1# set ge-2/0/2 unit 0 family inet address 10.0.244.10/30
user@PE1# set fe-2/0/10 unit 1 description PE1-to-P2
user@PE1# set fe-2/0/10 unit 1 family inet address 2.2.2.1/24
user@PE1# set fe-2/0/10 unit 1 family mpls
user@PE1# set fe-2/0/9 unit 8 description PE1-to-P3
user@PE1# set fe-2/0/9 unit 8 family inet address 6.6.6.1/24
user@PE1# set fe-2/0/9 unit 8 family mpls
user@PE1# set fe-2/0/8 unit 9 description PE1-to-P4
user@PE1# set fe-2/0/8 unit 9 family inet address 3.3.3.1/24
user@PE1# set fe-2/0/8 unit 9 family mpls
user@PE1# set lo0 unit 1 family inet address 100.10.10.10/32
```

2. Enable RSVP, MPLS, and OSPF on the interfaces.

```
[edit protocols]
user@PE1# set rsvp interface fe-2/0/10.1
user@PE1# set rsvp interface fe-2/0/9.8
user@PE1# set rsvp interface fe-2/0/8.9
user@PE1# set rsvp interface lo0.1
user@PE1# set mpls interface fe-2/0/10.1
user@PE1# set mpls interface fe-2/0/9.8
user@PE1# set mpls interface fe-2/0/8.9
user@PE1# set mpls interface lo0.1
user@PE1# set ospf area 0.0.0.0 interface ge-2/0/2.0
user@PE1# set ospf area 0.0.0.0 interface fe-2/0/10.1
user@PE1# set ospf area 0.0.0.0 interface fe-2/0/9.8
user@PE1# set ospf area 0.0.0.0 interface fe-2/0/8.9
user@PE1# set ospf area 0.0.0.0 interface lo0.1
```

3. Configure the MPLS point-to-multipoint LSPs.

```
[edit protocols]
user@PE1# set mpls label-switched-path PE1-PE2 to 100.50.50.50
user@PE1# set mpls label-switched-path PE1-PE2 p2mp p2mp1
user@PE1# set mpls label-switched-path PE1-PE3 to 100.70.70.70
user@PE1# set mpls label-switched-path PE1-PE3 p2mp p2mp1
user@PE1# set mpls label-switched-path PE1-PE4 to 100.40.40.40
user@PE1# set mpls label-switched-path PE1-PE4 p2mp p2mp1
```

4. (Optional) Enable link protection on the LSPs.

Link protection helps to ensure that traffic sent over a specific interface to a neighboring router can continue to reach the router if that interface fails.

```
[edit protocols]
user@PE1# set mpls label-switched-path PE1-PE2 link-protection
user@PE1# set mpls label-switched-path PE1-PE3 link-protection
user@PE1# set mpls label-switched-path PE1-PE4 link-protection
```

5. Enable MPLS to perform traffic engineering for OSPF.

```
[edit protocols]
user@PE1# set mpls traffic-engineering bgp-igp
```

This causes the ingress routes to be installed in the `inet.0` routing table. By default, MPLS performs traffic engineering for BGP only. You need to enable MPLS traffic engineering on the ingress LSR only.

6. Enable traffic engineering for OSPF.

```
[edit protocols]
user@PE1# set ospf traffic-engineering
```

This causes the shortest-path first (SPF) algorithm to take into account the LSPs configured under MPLS.

7. Configure the router ID.

```
[edit routing-options]
user@PE1# set router-id 100.10.10.10
```

8. Configure static IP unicast routes with the point-to-multipoint LSP name as the next hop for each route.

```
[edit routing-options]
user@PE1# set static route 5.5.5.0/24 p2mp-lsp-next-hop p2mp1
user@PE1# set static route 7.7.7.0/24 p2mp-lsp-next-hop p2mp1
user@PE1# set static route 4.4.4.0/24 p2mp-lsp-next-hop p2mp1
```

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

```
[edit]
user@PE1# commit
```

Configuring the Transit and Egress LSRs (Devices P2, P3, P4, PE2, PE3, and PE4)

Step-by-Step Procedure

To configure the transit and egress LSRs:

1. Configure the interfaces, interface encapsulation, and protocol families.

```
[edit]
user@P2# set interfaces fe-2/0/10 unit 2 description P2-to-PE1
user@P2# set interfaces fe-2/0/10 unit 2 family inet address 2.2.2.2/24
user@P2# set interfaces fe-2/0/10 unit 2 family mpls
user@P2# set interfaces fe-2/0/9 unit 10 description P2-to-PE2
user@P2# set interfaces fe-2/0/9 unit 10 family inet address 5.5.5.1/24
user@P2# set interfaces fe-2/0/9 unit 10 family mpls
user@P2# set interfaces lo0 unit 2 family inet address 100.20.20.20/32

user@PE2# set interfaces ge-2/0/3 unit 0 description PE2-to-CE2
user@PE2# set interfaces ge-2/0/3 unit 0 family inet address 10.0.224.10/30
user@PE2# set interfaces fe-2/0/10 unit 5 description PE2-to-P2
user@PE2# set interfaces fe-2/0/10 unit 5 family inet address 5.5.5.2/24
user@PE2# set interfaces fe-2/0/10 unit 5 family mpls
user@PE2# set interfaces lo0 unit 5 family inet address 100.50.50.50/32

user@P3# set interfaces fe-2/0/10 unit 6 description P3-to-PE1
```

```
user@P3# set interfaces fe-2/0/10 unit 6 family inet address 6.6.6.2/24
user@P3# set interfaces fe-2/0/10 unit 6 family mpls
user@P3# set interfaces fe-2/0/9 unit 11 description P3-to-PE3
user@P3# set interfaces fe-2/0/9 unit 11 family inet address 7.7.7.1/24
user@P3# set interfaces fe-2/0/9 unit 11 family mpls
user@P3# set interfaces lo0 unit 6 family inet address 100.60.60.60/32
```

```
user@PE3# set interfaces ge-2/0/1 unit 0 description PE3-to-CE3
user@PE3# set interfaces ge-2/0/1 unit 0 family inet address 10.0.134.10/30
user@PE3# set interfaces fe-2/0/10 unit 7 description PE3-to-P3
user@PE3# set interfaces fe-2/0/10 unit 7 family inet address 7.7.7.2/24
user@PE3# set interfaces fe-2/0/10 unit 7 family mpls
user@PE3# set interfaces lo0 unit 7 family inet address 100.70.70.70/32
```

```
user@P4# set interfaces fe-2/0/10 unit 3 description P4-to-PE1
user@P4# set interfaces fe-2/0/10 unit 3 family inet address 3.3.3.2/24
user@P4# set interfaces fe-2/0/10 unit 3 family mpls
user@P4# set interfaces fe-2/0/9 unit 12 description P4-to-PE4
user@P4# set interfaces fe-2/0/9 unit 12 family inet address 4.4.4.1/24
user@P4# set interfaces fe-2/0/9 unit 12 family mpls
user@P4# set interfaces lo0 unit 3 family inet address 100.30.30.30/32
```

```
user@PE4# set interfaces ge-2/0/0 unit 0 description PE4-to-CE4
user@PE4# set interfaces ge-2/0/0 unit 0 family inet address 10.0.104.9/30
user@PE4# set interfaces fe-2/0/10 unit 4 description PE4-to-P4
user@PE4# set interfaces fe-2/0/10 unit 4 family inet address 4.4.4.2/24
user@PE4# set interfaces fe-2/0/10 unit 4 family mpls
user@PE4# set interfaces lo0 unit 4 family inet address 100.40.40.40/32
```

2. Enable RSVP, MPLS, and OSPF on the interfaces.

```
[edit]
user@P2# set protocols rsvp interface fe-2/0/10.2
user@P2# set protocols rsvp interface fe-2/0/9.10
user@P2# set protocols rsvp interface lo0.2
user@P2# set protocols mpls interface fe-2/0/10.2
user@P2# set protocols mpls interface fe-2/0/9.10
user@P2# set protocols mpls interface lo0.2
user@P2# set protocols ospf area 0.0.0.0 interface fe-2/0/10.2
user@P2# set protocols ospf area 0.0.0.0 interface fe-2/0/9.10
user@P2# set protocols ospf area 0.0.0.0 interface lo0.2

user@PE2# set protocols rsvp interface fe-2/0/10.5
user@PE2# set protocols rsvp interface lo0.5
user@PE2# set protocols mpls interface fe-2/0/10.5
user@PE2# set protocols mpls interface lo0.5
user@PE2# set protocols ospf area 0.0.0.0 interface ge-2/0/3.0
user@PE2# set protocols ospf area 0.0.0.0 interface fe-2/0/10.5
user@PE2# set protocols ospf area 0.0.0.0 interface lo0.5

user@P3# set protocols rsvp interface fe-2/0/10.6
user@P3# set protocols rsvp interface fe-2/0/9.11
user@P3# set protocols rsvp interface lo0.6
user@P3# set protocols mpls interface fe-2/0/10.6
```



```

user@P3# set protocols mpls interface fe-2/0/9.11
user@P3# set protocols mpls interface lo0.6
user@P3# set protocols ospf area 0.0.0.0 interface fe-2/0/10.6
user@P3# set protocols ospf area 0.0.0.0 interface fe-2/0/9.11
user@P3# set protocols ospf area 0.0.0.0 interface lo0.6

```

```

user@PE3# set protocols rsvp interface fe-2/0/10.7
user@PE3# set protocols rsvp interface lo0.7
user@PE3# set protocols mpls interface fe-2/0/10.7
user@PE3# set protocols mpls interface lo0.7
user@PE3# set protocols ospf area 0.0.0.0 interface ge-2/0/1.0
user@PE3# set protocols ospf area 0.0.0.0 interface fe-2/0/10.7
user@PE3# set protocols ospf area 0.0.0.0 interface lo0.7

```

```

user@P4# set protocols rsvp interface fe-2/0/10.3
user@P4# set protocols rsvp interface fe-2/0/9.12
user@P4# set protocols rsvp interface lo0.3
user@P4# set protocols mpls interface fe-2/0/10.3
user@P4# set protocols mpls interface fe-2/0/9.12
user@P4# set protocols mpls interface lo0.3
user@P4# set protocols ospf area 0.0.0.0 interface fe-2/0/10.3
user@P4# set protocols ospf area 0.0.0.0 interface fe-2/0/9.12
user@P4# set protocols ospf area 0.0.0.0 interface lo0.3

```

```

user@PE4# set protocols rsvp interface fe-2/0/10.4
user@PE4# set protocols rsvp interface lo0.4
user@PE4# set protocols mpls interface fe-2/0/10.4
user@PE4# set protocols mpls interface lo0.4
user@PE4# set protocols ospf area 0.0.0.0 interface ge-2/0/0.0
user@PE4# set protocols ospf area 0.0.0.0 interface fe-2/0/10.4
user@PE4# set protocols ospf area 0.0.0.0 interface lo0.4

```

3. Enable traffic engineering for OSPF.

```

[edit]
user@P2# set protocols ospf traffic-engineering

```

```

user@P2# set protocols ospf traffic-engineering

```

```

user@P3# set protocols ospf traffic-engineering

```

```

user@PE2# set protocols ospf traffic-engineering

```

```

user@PE3# set protocols ospf traffic-engineering

```

```

user@PE4# set protocols ospf traffic-engineering

```

This causes the shortest-path first (SPF) algorithm to take into account the LSPs configured under MPLS.

4. Configure the router IDs.

```

[edit]
user@P2# set routing-options router-id 100.20.20.20

```

```
user@P3# set routing-options router-id 100.60.60.60
```

```
user@P4# set routing-options router-id 100.30.30.30
```

```
user@PE2# set routing-options router-id 100.50.50.50
```

```
user@PE3# set routing-options router-id 100.70.70.70
```

```
user@PE4# set routing-options router-id 100.40.40.40
```

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

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

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
Device PE1 user@PE1# show interfaces  
ge-2/0/2 {  
  unit 0 {  
    description R1-to-CE1;  
    family inet {  
      address 10.0.244.10/30;  
    }  
  }  
}  
fe-2/0/10 {  
  unit 1 {  
    description PE1-to-P2;  
    family inet {  
      address 2.2.2.1/24;  
    }  
    family mpls;  
  }  
}  
fe-2/0/9 {  
  unit 8 {  
    description PE1-to-P2;  
    family inet {  
      address 6.6.6.1/24;  
    }  
    family mpls;  
  }  
}  
fe-2/0/8 {  
  unit 9 {  
    description PE1-to-P3;  
    family inet {  
      address 3.3.3.1/24;  
    }  
    family mpls;  
  }  
}
```

```
}
lo0 {
  unit 1 {
    family inet {
      address 100.10.10.10/32;
    }
  }
}

user@PE1# show protocols
rsvp {
  interface fe-2/0/10.1;
  interface fe-2/0/9.8;
  interface fe-2/0/8.9;
  interface lo0.1;
}
mpls {
  traffic-engineering bgp-igp;
  label-switched-path PE1-to-PE2 {
    to 100.50.50.50;
    link-protection;
    p2mp p2mp1;
  }
  label-switched-path PE1-to-PE3 {
    to 100.70.70.70;
    link-protection;
    p2mp p2mp1;
  }
  label-switched-path PE1-to-PE4 {
    to 100.40.40.40;
    link-protection;
    p2mp p2mp1;
  }
  interface fe-2/0/10.1;
  interface fe-2/0/9.8;
  interface fe-2/0/8.9;
  interface lo0.1;
}
ospf {
  traffic-engineering;
  area 0.0.0.0 {
    interface ge-2/0/2.0;
    interface fe-2/0/10.1;
    interface fe-2/0/9.8;
    interface fe-2/0/8.9;
    interface lo0.1;
  }
}

user@PE1# show routing-options
static {
  route 5.5.5.0/24 {
    p2mp-lsp-next-hop p2mp1;
  }
  route 7.7.7.0/24 {
    p2mp-lsp-next-hop p2mp1;
  }
}
```

```
    }
    route 4.4.4.0/24 {
        p2mp-lsp-next-hop p2mpl;
    }
}
router-id 100.10.10.10;

Device P2 user@P2# show interfaces
fe-2/0/10 {
    unit 2 {
        description P2-to-PE1;
        family inet {
            address 2.2.2.2/24;
        }
        family mpls;
    }
fe-2/0/9 {
    unit 10 {
        description P2-to-PE2;
        family inet {
            address 5.5.5.1/24;
        }
        family mpls;
    }
}
lo0 {
    unit 2 {
        family inet {
            address 100.20.20.20/32;
        }
    }
}

user@P2# show protocols
rsvp {
    interface fe-2/0/10.2;
    interface fe-2/0/9.10;
    interface lo0.2;
}
mpls {
    interface fe-2/0/10.2;
    interface fe-2/0/9.10;
    interface lo0.2;
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface fe-2/0/10.2;
        interface fe-2/0/9.10;
        interface lo0.2;
    }
}

user@P2# show routing-options
router-id 100.20.20.20;
```

```
Device P3 user@P3# show interfaces
```

```

fe-2/0/10 {
  unit 6 {
    description P3-to-PE1;
    family inet {
      address 6.6.6.2/24;
    }
    family mpls;
  }
}
fe-2/0/9 {
  unit 11 {
    description P3-to-PE3;
    family inet {
      address 7.7.7.1/24;
    }
    family mpls;
  }
}
lo0 {
  unit 6 {
    family inet {
      address 100.60.60.60/32;
    }
  }
}

```

user@P3# show protocols

```

rsvp {
  interface fe-2/0/10.6;
  interface fe-2/0/9.11;
  interface lo0.6;
}
mpls {
  interface fe-2/0/10.6;
  interface fe-2/0/9.11;
  interface lo0.6;
}
ospf {
  traffic-engineering;
  area 0.0.0.0 {
    interface fe-2/0/10.6;
    interface fe-2/0/9.11;
    interface lo0.6;
  }
}

```

user@P2# show routing-options
router-id 100.60.60.60;

Device P4

```

user@P4# show interfaces
fe-2/0/10 {
  unit 3 {
    description P4-to-PE1;
    family inet {
      address 3.3.3.2/24;
    }
  }
}

```

```
        family mpls;
    }
}
fe-2/0/9 {
    unit 12 {
        description P4-to-PE4;
        family inet {
            address 4.4.4.1/24;
        }
        family mpls;
    }
}
lo0 {
    unit 3 {
        family inet {
            address 100.30.30.30/32;
        }
    }
}
```

user@P4# show protocols

```
rsvp {
    interface fe-2/0/10.3;
    interface fe-2/0/9.12;
    interface lo0.3;
}
mpls {
    interface fe-2/0/10.3;
    interface fe-2/0/9.12;
    interface lo0.3;
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface fe-2/0/10.3;
        interface fe-2/0/9.12;
        interface lo0.3;
    }
}
```

user@P3# show routing-options

```
router-id 100.30.30.30;
```

Device PE2

user@PE2# show interfaces

```
ge-2/0/3 {
    unit 0 {
        description PE2-to-CE2;
        family inet {
            address 10.0.224.10/30;
        }
    }
}
fe-2/0/10 {
    unit 5 {
        description PE2-to-P2;
        family inet {
```

```

        address 5.5.5.2/24;
    }
    family mpls;
}
}
lo0 {
    unit 5 {
        family inet {
            address 100.50.50.50/32;
        }
    }
}
}
}

```

```
user@PE2# show protocols
```

```

rsvp {
    interface fe-2/0/10.5;
    interface lo0.5;
}
mpls {
    interface fe-2/0/10.5;
    interface lo0.5;
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface ge-2/0/3.0;
        interface fe-2/0/10.5;
        interface lo0.5;
    }
}
}

```

```
user@PE2# show routing-options
router-id 100.50.50.50;
```

Device PE3

```
user@PE3# show interfaces
```

```

ge-2/0/1 {
    unit 0 {
        description PE3-to-CE3;
        family inet {
            address 10.0.134.10/30;
        }
    }
}
fe-2/0/10 {
    unit 7 {
        description PE3-to-P3;
        family inet {
            address 7.7.7.2/24;
        }
        family mpls;
    }
}
lo0 {
    unit 7 {
        family inet {

```

```
        address 100.70.70.70/32;
    }
}
}
```

```
user@PE3# show protocols
rsvp {
    interface fe-2/0/10.7;
    interface lo0.7;
}
mpls {
    interface fe-2/0/10.7;
    interface lo0.7;
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface ge-2/0/1.0;
        interface fe-2/0/10.7;
        interface lo0.7;
    }
}

user@PE3# show routing-options
router-id 100.70.70.70;
```

Device PE4

```
user@PE4# show interfaces
ge-2/0/0 {
    unit 0 {
        description PE4-to-CE4;
        family inet {
            address 10.0.104.9/30;
        }
    }
}
fe-2/0/10 {
    unit 4 {
        description PE4-to-P4;
        family inet {
            address 4.4.4.2/24;
        }
        family mpls;
    }
}
lo0 {
    unit 4 {
        family inet {
            address 100.40.40.40/32;
        }
    }
}

user@PE4# show protocols
rsvp {
```



```

interface fe-2/0/10.4;
interface lo0.4;
}
mpls {
interface fe-2/0/10.4;
interface lo0.4;
}
ospf {
traffic-engineering;
area 0.0.0.0 {
interface ge-2/0/0.0;
interface fe-2/0/10.4;
interface lo0.4;
}
}
}

user@PE4# show routing-options
router-id 100.40.40.40;

```

Configuring Device CE1

Step-by-Step Procedure

To configure Device CE1:

1. Configure an interface to Device PE1.

```

[edit interfaces]
user@CE1# set ge-1/3/2 unit 0 family inet address 10.0.244.9/30
user@CE1# set ge-1/3/2 unit 0 description CE1-to-PE1

```

2. Configure static routes from Device CE1 to the three other customer networks, with Device PE1 as the next hop.

```

[edit routing-options]
user@CE1# set static route 10.0.104.8/30 next-hop 10.0.244.10
user@CE1# set static route 10.0.134.8/30 next-hop 10.0.244.10
user@CE1# set static route 10.0.224.8/30 next-hop 10.0.244.10

```

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

```

[edit]
user@CE1# commit

```

Results From configuration mode, confirm your configuration by entering the **show interfaces** and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```

user@CE1# show interfaces
ge-1/3/2 {
unit 0 {
family inet {
address 10.0.244.9/30;
description CE1-to-PE1;
}
}
}

user@CE1# show routing-options

```

```
static {  
  route 10.0.104.8/30 next-hop 10.0.244.10;  
  route 10.0.134.8/30 next-hop 10.0.244.10;  
  route 10.0.224.8/30 next-hop 10.0.244.10;  
}
```

Configuring Device CE2

Step-by-Step Procedure

To configure Device CE2:

1. Configure an interface to Device PE2.

```
[edit interfaces]  
user@CE2# set ge-1/3/3 unit 0 family inet address 10.0.224.9/30  
user@CE2# set ge-1/3/3 unit 0 description CE2-to-PE2
```

2. Configure a static route from Device CE2 to CE1, with Device PE2 as the next hop.

```
[edit routing-options]  
user@CE2# set static route 10.0.244.8/30 next-hop 10.0.224.10
```

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

```
[edit]  
user@CE2# commit
```

Results From configuration mode, confirm your configuration by entering the **show interfaces** and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@CE2# show interfaces  
ge-1/3/3 {  
  unit 0 {  
    family inet {  
      address 10.0.224.9/30;  
      description CE2-to-PE2;  
    }  
  }  
}  
  
user@CE2# show routing-options  
static {  
  route 10.0.244.8/30 next-hop 10.0.224.10;  
}
```

Configuring Device CE3

Step-by-Step Procedure

To configure Device CE3:

1. Configure an interface to Device PE3.

```
[edit interfaces]  
user@CE3# set ge-2/0/1 unit 0 family inet address 10.0.134.9/30  
user@CE3# set ge-2/0/1 unit 0 description CE3-to-PE3
```

2. Configure a static route from Device CE3 to CE1, with Device PE3 as the next hop.

```
[edit routing-options]
```

```
user@CE3# set static route 10.0.244.8/30 next-hop 10.0.134.10
```

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

```
[edit]
user@CE3# commit
```

Results From configuration mode, confirm your configuration by entering the **show interfaces** and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@CE3# show interfaces
ge-2/0/1 {
  unit 0 {
    family inet {
      address 10.0.134.9/30;
      description CE3-to-PE3;
    }
  }
}

user@CE3# show routing-options
static {
  route 10.0.244.8/30 next-hop 10.0.134.10;
}
```

Configuring Device CE4

Step-by-Step Procedure To configure Device CE4:

1. Configure an interface to Device PE4.

```
[edit interfaces]
user@CE4# set ge-3/1/3 unit 0 family inet address 10.0.104.10/30
user@CE4# set ge-3/1/3 unit 0 description CE4-to-PE4
```

2. Configure a static route from Device CE4 to CE1, with Device PE4 as the next hop.

```
[edit routing-options]
user@CE4# set static route 10.0.244.8/30 next-hop 10.0.104.9
```

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

```
[edit]
user@CE4# commit
```

Results From configuration mode, confirm your configuration by entering the **show interfaces** and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@CE4# show interfaces
ge-3/1/3 {
  unit 0 {
    family inet {
      address 10.0.104.10/30;
      description CE4-to-PE4;
    }
  }
}
```

```
    }  
  }  
  
user@CE4# show routing-options  
static {  
    route 10.0.244.8/30 next-hop 10.0.104.9;  
}
```

Verification

Confirm that the configuration is working properly.

- [Verifying Connectivity on page 44](#)
- [Verifying the State of the Point-to-Multipoint LSP on page 45](#)
- [Checking the Forwarding Table on page 45](#)

Verifying Connectivity

Purpose Make sure that the devices can ping each other.

Action Run the **ping** command from CE1 to the interface on CE2 connecting to PE2.

```
user@CE1> ping 10.0.224.9  
PING 10.0.224.9 (10.0.224.9): 56 data bytes  
64 bytes from 10.0.224.9: icmp_seq=0 ttl=61 time=1.387 ms  
64 bytes from 10.0.224.9: icmp_seq=1 ttl=61 time=1.394 ms  
64 bytes from 10.0.224.9: icmp_seq=2 ttl=61 time=1.506 ms  
^C  
--- 10.0.224.9 ping statistics ---  
3 packets transmitted, 3 packets received, 0% packet loss  
round-trip min/avg/max/stddev = 1.387/1.429/1.506/0.055 ms
```

Run the **ping** command from CE1 to the interface on CE3 connecting to PE3.

```
user@CE1> ping 10.0.134.9  
PING 10.0.134.9 (10.0.134.9): 56 data bytes  
64 bytes from 10.0.134.9: icmp_seq=0 ttl=61 time=1.068 ms  
64 bytes from 10.0.134.9: icmp_seq=1 ttl=61 time=1.062 ms  
64 bytes from 10.0.134.9: icmp_seq=2 ttl=61 time=1.053 ms  
^C  
--- 10.0.134.9 ping statistics ---  
3 packets transmitted, 3 packets received, 0% packet loss  
round-trip min/avg/max/stddev = 1.053/1.061/1.068/0.006 ms
```

Run the **ping** command from CE1 to the interface on CE4 connecting to PE4.

```
user@CE1> ping 10.0.104.10  
PING 10.0.104.10 (10.0.104.10): 56 data bytes  
64 bytes from 10.0.104.10: icmp_seq=0 ttl=61 time=1.079 ms  
64 bytes from 10.0.104.10: icmp_seq=1 ttl=61 time=1.048 ms  
64 bytes from 10.0.104.10: icmp_seq=2 ttl=61 time=1.070 ms  
^C  
--- 10.0.104.10 ping statistics ---  
3 packets transmitted, 3 packets received, 0% packet loss  
round-trip min/avg/max/stddev = 1.048/1.066/1.079/0.013 ms
```

Verifying the State of the Point-to-Multipoint LSP

Purpose Make sure that the ingress, transit, and egress LSRs are in the Up state.

Action Run the `show mpls lsp p2mp` command on all of the LSRs. Only the ingress LSR is shown here.

```
user@PE1> show mpls lsp p2mp
Ingress LSP: 1 sessions
P2MP name: p2mp1, P2MP branch count: 3
To          From          State Rt P    ActivePath    LSPname
100.40.40.40 100.10.10.10 Up    0 *           PE1-PE4
100.70.70.70 100.10.10.10 Up    0 *           PE1-PE3
100.50.50.50 100.10.10.10 Up    0 *           PE1-PE2
Total 3 displayed, Up 3, Down 0
...
```

Checking the Forwarding Table

Purpose Make sure that the routes are set up as expected by running the `show route forwarding-table` command. Only the routes to the remote customer networks are shown here.

```
user@PE1> show route forwarding-table
Routing table: default.inet
Internet:
Destination          Type RtRef Next hop          Type Index NhRef Netif
...
10.0.104.8/30         user    0 3.3.3.2          ucst  1006   6 fe-2/0/8.9
10.0.134.8/30         user    0 6.6.6.2          ucst  1010   6 fe-2/0/9.8
10.0.224.8/30         user    0 2.2.2.2          ucst  1008   6 fe-2/0/10.1
...
```

Related Documentation

- [RSVP Signaling Protocol on page 18](#)
- [Example: Configuring RSVP-Signaled Point-to-Multipoint LSPs on Logical Systems](#)

CHAPTER 3

MPLS VPNs

- [MPLS VPN Overview on page 47](#)
- [Layer 2 VPNs on page 50](#)
- [Layer 3 VPNs on page 57](#)
- [Layer 2 Circuits on page 60](#)

MPLS VPN Overview

Virtual private networks (VPNs) are private networks that use a public network to connect two or more remote sites. Instead of dedicated connections between networks, VPNs use virtual connections routed (tunneled) through public networks that are typically service provider networks. VPNs are a cost-effective alternative to expensive dedicated lines. The type of VPN is determined by the connections it uses and whether the customer network or the provider network performs the virtual tunneling.

You can configure a routers running Junos OS to participate in several types of VPNs. This topic discusses MPLS VPNs.

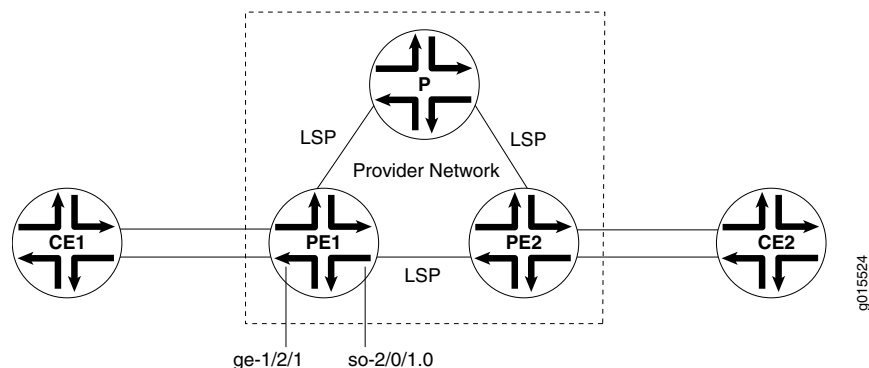
This topic contains the following sections:

- [MPLS VPN Topology on page 47](#)
- [MPLS VPN Routing on page 48](#)
- [VRF Instances on page 48](#)
- [Route Distinguishers on page 49](#)

MPLS VPN Topology

There are many ways to set up an MPLS VPN and direct traffic through it. [Figure 6 on page 48](#) shows a typical MPLS VPN topology.

Figure 6: Typical VPN Topology



There are three primary types of MPLS VPNs: Layer 2 VPNs, Layer 2 circuits, and Layer 3 VPNs. All types of MPLS VPNs share certain components:

- The *provider edge (PE) routers* in the provider's network connect to the customer edge (CE) routers located at customer sites. PE routers support VPN and MPLS label functionality. Within a single VPN, pairs of PE routers are connected through a virtual tunnel, typically a label-switched path (LSP).
- *Provider routers* within the core of the provider's network are not connected to any routers at a customer site but are part of the tunnel between pairs of PE routers. Provider routers support LSP functionality as part of the tunnel support, but do not support VPN functionality.
- *CE routers* are the routers or switches located at the customer site that connect to the provider's network. CE routers are typically IP routers, but they can also be Asynchronous Transfer Mode (ATM), Frame Relay, or Ethernet switches.

All VPN functions are performed by the PE routers. Neither CE routers nor provider routers are required to perform any VPN functions.

MPLS VPN Routing

VPNs tunnel traffic as follows from one customer site to another customer site, using a public network as a transit network, when certain requirements are met:

1. Traffic is forwarded by standard IP forwarding from the CE routers to the PE routers.
2. The PE routers establish an LSP through the provider network.
3. The inbound PE router receives traffic, and it performs a route lookup. The lookup yields an LSP next hop, and the traffic is forwarded along the LSP.
4. The traffic reaches the outbound PE router, and the PE router pops the MPLS label and forwards the traffic with standard IP routing.

VRF Instances

A *routing instance* is a collection of routing tables, interfaces, and routing protocol parameters. The interfaces belong to the routing tables, and the routing protocol

parameters control the information in the routing tables. In the case of MPLS VPNs, each VPN has a VPN routing and forwarding (VRF) instance.

A *VRF instance* consists of one or more routing tables, a derived forwarding table, the interfaces that use the forwarding table, and the policies and routing protocols that determine what goes into the forwarding table. Because each instance is configured for a particular VPN, each VPN has separate tables, rules, and policies that control its operation.

A separate VRF table is created for each VPN that has a connection to a CE router. The VRF table is populated with routes received from directly connected CE sites associated with the VRF instance, and with routes received from other PE routers in the same VPN.

Route Distinguishers

Because a typical transit network is configured to handle more than one VPN, the provider routers are likely to have multiple VRF instances configured. As a result, depending on the origin of the traffic and any filtering rules applied to the traffic, the BGP routing tables can contain multiple routes for a particular destination address. Because BGP requires that exactly one BGP route per destination be imported into the forwarding table, BGP must have a way to distinguish between potentially identical network layer reachability information (NLRI) messages received from different VPNs.

A *route distinguisher* is a locally unique number that identifies all route information for a particular VPN. Unique numeric identifiers allow BGP to distinguish between routes that are otherwise identical.

Each routing instance that you configure on a PE router must have a unique route distinguisher. There are two possible formats:

- ***as-number:number***, where ***as-number*** is an autonomous system (AS) number (a 2-byte value) in the range 1 through 65,535, and ***number*** is any 4-byte value. We recommend that you use an Internet Assigned Numbers Authority (IANA)-assigned, nonprivate AS number, preferably the ISP or the customer 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 public IP address in your assigned prefix range.

The *route target* defines which route is part of a VPN. A unique route target helps distinguish between different VPN services on the same router. Each VPN also has a policy that defines how routes are imported into the VRF table on the router. A Layer 2 VPN is configured with import and export policies. A Layer 3 VPN uses a unique route target to distinguish between VPN routes.

The PE router then exports the route in IBGP sessions to the other provider routers. Route export is governed by any routing policy that has been applied to the particular VRF table. To propagate the routes through the provider network, the PE router must also convert the route to VPN format, which includes the route distinguisher.

When the outbound PE router receives the route, it strips off the route distinguisher and advertises the route to the connected CE router, typically through standard BGP IPv4 route advertisements.

**Related
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Junos OS VPNs Configuration Guide](#)
- [Understanding MPLS Layer 2 VPNs on page 50](#)
- [Understanding MPLS Layer 3 VPNs on page 58](#)
- [Understanding MPLS Layer 2 Circuits on page 60](#)

Layer 2 VPNs

- [Understanding MPLS Layer 2 VPNs on page 50](#)
- [MPLS Layer 2 VPN Configuration Overview on page 50](#)
- [Configuring MPLS for Layer 2 VPNs \(CLI Procedure\) on page 52](#)
- [Configuring MPLS for Layer 2 VPNs \(CLI Procedure\) on page 53](#)
- [Configuring a BGP Session for MPLS VPNs \(CLI Procedure\) on page 53](#)
- [Configuring an IGP and the LDP Signaling Protocol \(CLI Procedure\) on page 54](#)
- [Configuring an IGP and the RSVP Signaling Protocol \(CLI Procedure\) on page 55](#)
- [Configuring Routing Options for MPLS VPNs \(CLI Procedure\) on page 55](#)
- [Configuring a Routing Instance for MPLS VPNs \(CLI Procedure\) on page 55](#)
- [Configuring a Routing Policy for MPLS Layer 2 VPNs \(CLI Procedure\) on page 56](#)
- [Verifying an MPLS Layer 2 VPN Configuration on page 57](#)

Understanding MPLS Layer 2 VPNs

In an MPLS Layer 2 VPN, traffic is forwarded to the provider edge (PE) router in Layer 2 format, carried by MPLS through an label-switched path (LSP) over the service provider network, and then converted back to Layer 2 format at the receiving customer edge (CE) router.

Routing occurs on the customer 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 network to the PE router on the outbound side. The PE routers need no information about the customer's routes or routing topology, and need only to determine the virtual tunnel through which to send the traffic.

Implementing a Layer 2 VPN on the router is similar to implementing a VPN using a Layer 2 technology such as Asynchronous Transfer Mode (ATM) or Frame Relay.

MPLS Layer 2 VPN Configuration Overview

To configure MPLS Layer 2 VPN functionality on a router running Junos OS, you must enable support on the provider edge (PE) router and configure the PE router to distribute

routing information to other routers in the VPN, as explained in the following steps. However, because the tunnel information is maintained at both PE routers, neither the provider core routers nor the customer edge (CE) routers need to maintain any VPN information in their configuration databases.

To configure an MPLS Layer 2 VPN:

1. Determine all of the routers that you want to participate in the VPN, and then complete the initial configuration of their interfaces. See the [Junos OS Interfaces Configuration Guide for Security Devices](#).
2. For all of the routers in the VPN configuration, update the interface configurations to enable participation in the Layer 2 VPN. As part of the interface configuration, you must configure the MPLS address family for each interface that uses LDP or RSVP. See [Configuring Interfaces for Layer 2 VPNs \(CLI Procedure\)](#).
3. For all of the routers in the VPN configuration, configure the appropriate protocols.
 - a. MPLS—For PE routers and provider routers, use MPLS to advertise the Layer 2 VPN interfaces that communicate with other PE routers and provider routers. See [“Configuring MPLS for Layer 2 VPNs \(CLI Procedure\)” on page 52](#).
 - b. BGP and internal BGP (IBGP)—For PE routers, configure a BGP session to enable the routers to exchange information about routes originating and terminating in the VPN. (The PE routers use this information to determine which labels to use for traffic destined to the remote sites. The IBGP session for the VPN runs through the loopback address.) In addition, CE routers require a BGP connection to the PE routers. See [“Configuring a BGP Session for MPLS VPNs \(CLI Procedure\)” on page 53](#).
 - c. IGP and a signaling protocol—For PE routers, configure a signaling protocol (either LDP or RSVP) to dynamically set up label-switched paths (LSPs) through the provider network. (LDP routes traffic using IGP metrics. RSVP has traffic engineering that lets you override IGP metrics as needed.) You must use LDP or RSVP between PE routers and provider routers, but you cannot use them for interfaces between PE routers and CE routers.

In addition, configure an IGP such as OSPF or static routes for PE routers to enable exchanges of routing information between the PE routers and provider routers. Each PE router's loopback address must appear as a separate route. Do not configure any summarization of the PE router's loopback addresses at the area boundary. Configure the provider network to run OSPF or IS-IS as an IGP, as well as IBGP sessions through either a full mesh or route reflector.

See [“Configuring an IGP and the LDP Signaling Protocol \(CLI Procedure\)” on page 54](#) and [“Configuring an IGP and the RSVP Signaling Protocol \(CLI Procedure\)” on page 55](#).

4. For all of the routers in the VPN configuration, configure routing options. The only required routing option for VPNs is the AS number. You must specify it on each router involved in the VPN. See [“Configuring Routing Options for MPLS VPNs \(CLI Procedure\)” on page 55](#).

5. For each PE router in the VPN configuration, configure a routing instance for each VPN. The routing instance should have the same name on each PE router. Each routing instance must have a unique route distinguisher associated with it. (VPN routing instances need a route distinguisher to help BGP distinguish between potentially identical network layer reachable information [NLR] messages received from different VPNs.) See [“Configuring a Routing Instance for MPLS VPNs \(CLI Procedure\)”](#) on page 55.
6. For each PE router in the VPN configuration, configure a VPN routing policy if you are not using a route target. Within the policy, describe which packets are sent and received across the VPN and specify how routes are imported into and exported from the router's VRF table. Each advertisement must have an associated route target that uniquely identifies the VPN for which the advertisement is valid. If the routing instance uses a policy for accepting and rejecting packets instead of a route target, you must specify the import and export routing policies and the community on each PE router. See [“Configuring a Routing Policy for MPLS Layer 2 VPNs \(CLI Procedure\)”](#) on page 56.

Configuring MPLS for Layer 2 VPNs (CLI Procedure)

To configure MPLS for VPNs:

1. Specify the interfaces used for communication between PE routers or between PE routers and provider routers.

```
[edit]
user@host# edit protocols mpls interface interface-name
```

2. For RSVP only, configure the PE router interface communicating with another PE router or PE router with another provider router.

- a. Configure an MPLS LSP to the destination point on the PE router. The label switched path name is defined on the source router only and is unique between two routers.

```
[edit protocols mpls]
user@host# set label-switched-path-name path-name
```

- b. Specify the IP address of the LSP destination point, which is an address on the remote PE router.

```
[edit edit protocols mpls label-switched-path path-name ]
user@host# set to ip-address
```

- c. Enable protocol RSVP on the PE router interface communicating with another PE router or provider router.

```
[edit edit protocols mpls label-switched-path path-name ]
user@host# set protocols rsvp interface interface-name
```

3. Commit the configuration if you are finished configuring the device.

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

Configuring MPLS for Layer 2 VPNs (CLI Procedure)

To configure MPLS for VPNs:

1. Specify the interfaces used for communication between PE routers or between PE routers and provider routers.

```
[edit]
user@host# edit protocols mpls interface interface-name
```

2. For RSVP only, configure the PE router interface communicating with another PE router or PE router with another provider router.
 - a. Configure an MPLS LSP to the destination point on the PE router. The label switched path name is defined on the source router only and is unique between two routers.

```
[edit protocols mpls]
user@host# set label-switched-path-name path-name
```

- b. Specify the IP address of the LSP destination point, which is an address on the remote PE router.

```
[edit edit protocols mpls label-switched-path path-name ]
user@host# set to ip-address
```

- c. Enable protocol RSVP on the PE router interface communicating with another PE router or provider router.

```
[edit edit protocols mpls label-switched-path path-name ]
user@host# set protocols rsvp interface interface-name
```

3. Commit the configuration if you are finished configuring the device.

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

Configuring a BGP Session for MPLS VPNs (CLI Procedure)



NOTE: This section is valid for Layer 2 VPNs and Layer 3 VPNs, but not Layer 2 circuits.

To configure an IBGP session, perform the following steps on each PE router:

1. Configure BGP.

```
[edit]
user@host# edit protocols bgp group group-name
```

2. Set the BGP type to internal.

```
[edit protocols bgp group group-name]
user@host# set type internal
```

3. Specify the loopback interface.

```
[edit protocols bgp group group-name]
```

```
user@host# set local-address loopback-interface-ip-address
```

4. Set the Layer 2 or Layer 3 VPN family type to unicast.

```
[edit protocols bgp group group-name]  
user@host# set family family-type unicast
```

Replace *family-type* with *l2vpn* for a Layer 2 VPN or *inet-vpn* for a Layer 3 VPN.

5. Enter the loopback address of the neighboring PE router.

```
[edit protocols bgp]  
user@host# set neighbor ip-address
```

6. Commit the configuration if you are finished configuring the device.

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

Configuring an IGP and the LDP Signaling Protocol (CLI Procedure)

The following instructions show how to configure LDP and OSPF on PE routers and provider routers. Within the task, you specify which interfaces to enable for LDP. Perform this step on each PE router interface and provider router interface that communicates with other PE routers and provider routers. For OSPF, you configure at least one area on at least one of the router's interfaces. (An AS can be divided into multiple areas.) These instructions use the backbone area 0.0.0.0 and show how to enable traffic engineering for Layer 2 VPN circuits.

To configure LDP and OSPF:

1. Enable the ldp protocol.

```
[edit]  
user@host# edit protocols ldp
```



NOTE: You must configure the IGP at the [protocols] level of the configuration hierarchy, not within the routing instance at the [routing-instances] level of the configuration hierarchy.

2. Specify which interfaces to enable for LDP.

```
[edit protocols ldp]  
user@host# edit interface interface-name
```

3. Configure OSPF for each interface that uses LDP.

```
[edit]  
user@host# edit protocols ospf area 0.0.0.0 interface interface-name
```

4. (Layer 2 VPN circuits only) Enable traffic engineering.

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

5. Commit the configuration if you are finished configuring the device.

```
[edit]
```

```
user@host# commit
```

Configuring an IGP and the RSVP Signaling Protocol (CLI Procedure)

To configure RSVP and OSPF:

1. Configure OSPF with traffic engineering support on the PE routers.

```
[edit]
user@host# edit protocols ospf traffic-engineering shortcuts
```



NOTE: You must configure the IGP at the [edit protocols] level, not within the routing instance at the [edit routing-instances] level.

2. Enable RSVP on interfaces that participate in the LSP. For PE routers, enable interfaces on the source and destination points. For provider routers, enable interfaces that connect the LSP between the PE routers.

```
[edit]
user@host# edit protocols rsvp interface interface-name
```

3. Commit the configuration if you are finished configuring the device.

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

Configuring Routing Options for MPLS VPNs (CLI Procedure)

To configure routing options for a VPN:

1. Configure the AS number.

```
[edit]
user@host# set routing-options autonomous-system as-number
```

2. Commit the configuration if you are finished configuring the device.

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

Configuring a Routing Instance for MPLS VPNs (CLI Procedure)

To configure a VPN routing instance on each PE router:

1. Create the routing instance.

```
[edit]
user@host# edit routing-instances routing-instance-name
```

2. Create a routing instance description. (This text appears in the output of the **show route instance detail** command.)

```
[edit routing-instances routing-instance-name]
user@host# set description "text"
```

3. Specify the instance type, either **l2vpn** for Layer 2 VPNs or **vrf** for Layer 3 VPNs.

```
[edit routing-instances routing-instance-name]  
user@host# set instance-type instance-type
```

- Specify the interface of the remote PE router.

```
[edit routing-instances routing-instance-name]  
user@host# set interface interface-name
```

- Specify the route distinguisher using one of the following commands:

```
[edit routing-instances routing-instance-name]  
user@host# set route-distinguisher-as-number: number  
user@host# set route-distinguisher ip-address: number
```

- Specify the policy for the Layer 2 VRF table.

```
[edit routing-instances routing-instance-name]  
user@host# set vrf-import import-policy-name vrf-export export-policy-name
```

- Specify the policy for the Layer 3 VRF table.

```
[edit routing-instances routing-instance-name]  
user@host# set vrf-target target: community-id
```

Where *community-id* is either *as-number: number* or *ip-address: number*.

- Commit the configuration if you are finished configuring the device.

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

Configuring a Routing Policy for MPLS Layer 2 VPNs (CLI Procedure)

These instructions show how to configure a Layer 2 VPN routing policy on the PE routers in the VPN.

After configuring an import routing policy for a Layer 2 VPN, configure an export routing policy for the Layer 2 VPN. Configure this export policy on the PE routers in the VPN. The export routing policy defines how routes are exported from the PE router routing table. An export policy is applied to routes sent to other PE routers in the VPN. The export policy must also evaluate all routes received over the routing protocol session with the CE router. The export policy must also contain a second term for rejecting all other routes.

To configure a Layer 2 VPN routing policy on a PE router:

- Configure the import routing policy.

```
[edit]  
user@host# edit policy-options policy-statement import-policy-name
```

- Define the import policy's term for accepting packets.

```
[edit edit policy-options policy-statement import-policy-name]  
user@host# set term term-name-accept from protocol bgp community  
                  community-name  
user@host# set term term-name-accept then accept
```

- Define the import policy's term for rejecting packets.

```
[edit edit policy-options policy-statement import-policy-name]  
user@host# set term term-name-reject then reject
```


4. Configure the export routing policy.

```
[edit]
user@host# edit policy-options policy-statement export-policy-name
```

5. Define the export policy's term for accepting packets.

```
[edit policy-options policy-statement export-policy-name]
user@host# set term term-name-accept from community add community-name
user@host# set term term-name-accept then accept
```

6. Define the export policy's term for rejecting packets.

```
[edit policy-options policy-statement export-policy-name]
user@host# set term term-name-reject from community add community-name
user@host# set term term-name-reject then reject
```

7. Define the export policy's community using one of the following commands.

```
[edit policy-options policy-statement export-policy-name]
user@host# community community-name target: as-number
user@host# community community-name target: ip-address:number
```

8. Commit the configuration if you are finished configuring the device.

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

Verifying an MPLS Layer 2 VPN Configuration

Purpose Verify the connectivity of MPLS Layer 2 VPNs using the **ping mpls** command. This command helps to verify that a VPN has been enabled by testing the integrity of the VPN connection between the PE routers. It does not test the connection between a PE router and CE router.

Action

- To ping an interface configured for the Layer 2 VPN on the PE router, use the following command:

```
ping mpls l2vpn interface interface-name
```

- To 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 identifiers) between the two PE routers, use the following command:

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

Layer 3 VPNs

- [Understanding MPLS Layer 3 VPNs on page 58](#)
- [MPLS Layer 3 VPN Configuration Overview on page 58](#)
- [Configuring a Routing Policy for MPLS Layer 3 VPNs \(CLI Procedure\) on page 60](#)
- [Verifying an MPLS Layer 3 VPN Configuration on page 60](#)

Understanding MPLS Layer 3 VPNs

An MPLS Layer 3 VPN operates at the Layer 3 level of the OSI model, the Network layer. The VPN is composed of a set of sites that are connected over a service provider's existing public Internet backbone. The sites share common routing information and the connectivity of the sites is controlled by a collection of policies.

In an MPLS Layer 3 VPN, routing occurs on the service provider's routers. The provider routers route and forward VPN traffic at the entry and exit points of the transit network. The service provider network must learn the IP addresses of devices sending traffic across the VPN and the routes must be advertised and filtered throughout the provider network. As a result, Layer 3 VPNs require information about customer routes and a more extensive VPN routing and forwarding (VRF) policy configuration than a Layer 2 VPN. This information is used to share and filter routes that originate or terminate in the VPN.

The MPLS Layer 3 VPN requires more processing power on the provider edge (PE) routers than a Layer 2 VPN, because the Layer 3 VPN has larger routing tables for managing network traffic on the customer sites. Route advertisements originate at the customer edge (PE) routers and are shared with the inbound PE routers through standard IP routing protocols, typically BGP. Based on the source address, the PE router filters route advertisements and imports them into the appropriate VRF table.

The provider router uses OSPF and LDP to communicate with the PE routers. For OSPF, the provider router interfaces that communicate with the PE routers are specified, as well as the loopback interface. For the PE routers, the loopback interface is in passive mode, meaning it does not send OSPF packets to perform the control function.

MPLS Layer 3 VPN Configuration Overview

To configure MPLS Layer 3 VPN functionality on a router running Junos OS, you must enable support on the provider edge (PE) router and configure the PE router to distribute routing information to other routers in the VPN, as explained in the following steps. However, because the tunnel information is maintained at both PE routers, neither the provider core routers nor the customer edge (CE) routers need to maintain any VPN information in their configuration databases.

To configure an MPLS Layer 3 VPN:

1. Determine all of the routers that you want to participate in the VPN, and then complete the initial configuration of their interfaces. See the [Junos OS Interfaces Configuration Guide for Security Devices](#).
2. For all of the routers in the VPN configuration, update the interface configurations to enable participation in the Layer 3 VPN. As part of the interface configuration, you must configure the MPLS address family for each interface that uses LDP or RSVP. See [Configuring Interfaces for Layer 2 VPNs \(CLI Procedure\)](#).
3. For all of the routers in the VPN configuration, configure the appropriate protocols.
 - a. MPLS—If you are using RSVP, use MPLS to advertise the Layer 3 VPN interfaces on the PE routers and provider routers that communicate with other PE routers

and provider routers. See [“Configuring MPLS for Layer 2 VPNs \(CLI Procedure\)” on page 52.](#)

- b. BGP, EBGp, and internal BGP (IBGP)—For PE routers, configure a BGP session to enable the routers to exchange information about routes originating and terminating in the VPN. (The PE routers use this information to determine which labels to use for traffic destined to the remote sites. The IBGP session for the VPN runs through the loopback address.) In addition, CE routers require a BGP connection to the PE routers. See [“Configuring a BGP Session for MPLS VPNs \(CLI Procedure\)” on page 53.](#)
- c. IGP and a signaling protocol—For PE routers and provider, configure a signaling protocol (either LDP or RSVP) to dynamically set up label-switched paths (LSPs) through the provider network. (LDP routes traffic using IGP metrics. RSVP has traffic engineering that lets you override IGP metrics as needed.) You must use LDP or RSVP between PE routers and provider routers, but cannot use them for interfaces between PE routers and CE routers.

In addition, configure an IGP such as OSPF or static routes on the PE routers in order to enable exchanges of routing information between the PE routers and provider routers. Each PE router's loopback address must appear as a separate route. Do not configure any summarization of the PE router's loopback addresses at the area boundary. Configure the provider network to run OSPF or IS-IS as an IGP, as well as IBGP sessions through either a full mesh or route reflector.

See [“Configuring an IGP and the LDP Signaling Protocol \(CLI Procedure\)” on page 54](#) and [“Configuring an IGP and the RSVP Signaling Protocol \(CLI Procedure\)” on page 55.](#)

- 4. For all of the routers in the VPN configuration, configure routing options. The only required routing option for VPNs is the autonomous system (AS) number. You must specify it on each router involved in the VPN. See [“Configuring Routing Options for MPLS VPNs \(CLI Procedure\)” on page 55.](#)
- 5. For each PE router in the VPN configuration, configure a routing instance for each VPN. The routing instance should have the same name on each PE router. Each routing instance must have a unique route distinguisher associated with it. (VPN routing instances need a route distinguisher to help BGP distinguish between potentially identical network layer reachable information [NLRI] messages received from different VPNs.) See [“Configuring a Routing Instance for MPLS VPNs \(CLI Procedure\)” on page 55.](#)
- 6. For CE routers, configure a routing policy. In addition, if you are not using a route target, configure a VPN routing policy for each PE router in the VPN configuration. Within the policy, describe which packets are sent and received across the VPN and specify how routes are imported into and exported from the router's VRF table. Each advertisement must have an associated route target that uniquely identifies the VPN for which the advertisement is valid. See [“Configuring a Routing Policy for MPLS Layer 3 VPNs \(CLI Procedure\)” on page 60.](#)

Configuring a Routing Policy for MPLS Layer 3 VPNs (CLI Procedure)

To configure a Layer 3 VPN routing policy on a CE router:

1. Configure the routing policy for the loopback interface.

```
[edit]
user@host# edit policy-options policy-statement policy-name
```

2. Define the term for accepting packets.

```
[edit policy-options policy-statement policy-name]
user@host# set term term-name-accept from protocol direct route-filter
local-loopback-address/netmask exact
user@host# set term term-name-accept then accept
```

3. Define the term for rejecting packets.

```
[edit policy-options policy-statement policy-name]
user@host# set term term-name-reject then reject
```

4. Commit the configuration if you are finished configuring the device.

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

Verifying an MPLS Layer 3 VPN Configuration

Purpose Verify the connectivity of MPLS Layer 3 VPNs using the **ping mpls** command. This command helps to verify that a VPN has been enabled by testing the integrity of the VPN connection between the source and destination routers. The destination prefix corresponds to a prefix in the Layer 3 VPN. However, ping tests only whether the prefix is present in a PE VRF table.

Action To a combination of a IPv4 destination prefix and a Layer 3 VPN name on the destination PE router, use the following command:

```
ping mpls l3vpn l3vpn-name prefix prefix count count
```

Layer 2 Circuits

- [Understanding MPLS Layer 2 Circuits on page 60](#)
- [MPLS Layer 2 Circuit Configuration Overview on page 61](#)
- [Configuring an MPLS Layer 2 Circuit \(CLI Procedure\) on page 62](#)
- [Verifying an MPLS Layer 2 Circuit Configuration on page 62](#)

Understanding MPLS Layer 2 Circuits

An MPLS Layer 2 circuit is a point-to-point Layer 2 connection that transports traffic by means of MPLS or another tunneling technology on the service provider network. The Layer 2 circuit creates a virtual connection to direct traffic between two customer edge (CE) routers across a service provider network. The main difference between a Layer 2 VPN and a Layer 2 circuit is the method of setting up the virtual connection. As with a

leased line, a Layer 2 circuit forwards all packets received from the local interface to the remote interface.

Each Layer 2 circuit is represented by the logical interface connecting the local provider edge (PE) router to the local CE router. All Layer 2 circuits using a particular remote PE router neighbor is identified by its IP address and is usually the endpoint destination for the label-switched path (LSP) tunnel transporting the Layer 2 circuit.

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. Based on the virtual circuit ID and the neighbor relationship, an LDP label is bound to an LDP circuit. LDP uses the binding for sending traffic on that Layer 2 circuit to the remote CE router.

MPLS Layer 2 Circuit Configuration Overview

To configure an MPLS Layer 2 circuit:

1. Determine all of the routers that you want to participate in the circuit, and then complete the initial configuration of their interfaces. See the [Junos OS Interfaces Configuration Guide for Security Devices](#).
2. For all of the routers in the circuit configuration, update the interface configurations to enable participation in the Layer 2 circuit.
 - a. On the interface communicating with the other provider edge (PE) router, specify MPLS and IPv4, and include the IP address. For the loopback interface, specify **inet**, and include the IP address. For IPv4, designate the loopback interface as primary so it can receive control packets. (Because it is always operational, the loopback interface is best able to perform the control function.)
 - b. On the PE router interface facing the customer edge (CE) router, specify a circuit cross-connect (CCC) encapsulation type. The type of encapsulation depends on the interface type. For example, an Ethernet interface uses **ethernet-ccc**. (The encapsulation type determines how the packet is constructed for that interface.)
 - c. On the CE router interface that faces the PE router, specify **inet** (for IPv4), and include the IP address. In addition, specify a routing protocol such as Open Shortest Path First (OSPF), which specifies the area and IP address of the router interface.

See [Configuring Interfaces for Layer 2 VPNs \(CLI Procedure\)](#).

3. For all of the routers in the circuit configuration, configure the appropriate protocols.
 - a. MPLS—For PE routers and provider routers, use MPLS to advertise the Layer 2 circuit interfaces that communicate with other PE routers and provider routers. See [“Configuring MPLS for Layer 2 VPNs \(CLI Procedure\)” on page 52](#).
 - b. BGP—For PE routers, configure a BGP session.
 - c. IGP and a signaling protocol—For PE routers, configure a signaling protocol (either LDP or RSVP) to dynamically set up label-switched paths (LSPs) through the provider network. (LDP routes traffic using IGP metrics. RSVP has traffic engineering that lets you override IGP metrics as needed.) You must use LDP or RSVP between

PE routers and provider routers, but cannot use them for interfaces between PE routers and CE routers.

In addition, configure an IGP such as OSPF or static routes on the PE routers to enable exchanges of routing information between the PE routers and provider routers. Each PE router's loopback address must appear as a separate route. Do not configure any summarization of the PE router's loopback addresses at the area boundary. Configure the provider network to run OSPF or IS-IS as an IGP, as well as IBGP sessions through either a full mesh or route reflector.

See [“Configuring an IGP and the LDP Signaling Protocol \(CLI Procedure\)” on page 54](#) and [“Configuring an IGP and the RSVP Signaling Protocol \(CLI Procedure\)” on page 55](#).

4. For all of the routers in the circuit configuration, configure routing options. The only required routing option for circuits is the autonomous system (AS) number. You must specify it on each router involved in the circuit. See [“Configuring Routing Options for MPLS VPNs \(CLI Procedure\)” on page 55](#).
5. For PE routers, configure Layer 2 circuits on the appropriate interfaces. See [“Configuring an MPLS Layer 2 Circuit \(CLI Procedure\)” on page 62](#).

Configuring an MPLS Layer 2 Circuit (CLI Procedure)

To configure a Layer 2 circuit on a PE router:

1. Enable a Layer 2 circuit on the appropriate interface.

```
[edit]
user@host# edit protocols l2circuit neighbor interface-name interface interface-name
```

2. Enter the circuit ID number.

```
[edit protocols l2circuit neighbor interface-name interface interface-name]
user@host# set virtual-circuit-id id-number
```

For **neighbor**, specify the local loopback address, and for **interface**, specify the interface name of the remote PE router.

3. Commit the configuration if you are finished configuring the device.

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

Verifying an MPLS Layer 2 Circuit Configuration

Purpose To verify the connectivity of MPLS Layer 2 circuits, use the **ping mpls** command. This command helps to verify that the circuit has been enabled by testing the integrity of the Layer 2 circuit between the source and destination routers.

Action

- To ping an interface configured for the Layer 2 circuit on the PE router, enter the following command:

```
ping mpls l2circuit interface interface-name
```

- To ping a combination of the IPv4 prefix and the virtual circuit ID on the destination PE router, enter the following command:

ping mpls l2circuit virtual-circuit~~prefix~~ virtual-circuit-id

CHAPTER 4

CLNS VPNs

- [CLNS Overview on page 65](#)
- [CLNS Configuration Overview on page 66](#)
- [Example: Configuring a VPN Routing Instance for CLNS on page 67](#)
- [ES-IS for CLNS on page 69](#)
- [IS-IS for CLNS on page 71](#)
- [Example: Configuring Static Routes for CLNS on page 74](#)
- [BGP for CLNS VPNs on page 76](#)
- [Verifying a CLNS VPN Configuration on page 78](#)

CLNS Overview

Connectionless Network Service (CLNS) is a Layer 3 protocol similar to IP version 4 (IPv4) for linking hosts (end systems) with routers (intermediate systems) in an Open Systems Interconnection (OSI) network. CLNS and its related OSI protocols, Intermediate System-to-Intermediate System (IS-IS) and End System-to-Intermediate System (ES-IS), are International Organization for Standardization (ISO) standards.

You can configure devices running Junos OS as provider edge (PE) routers within a CLNS network. CLNS networks can be connected over an IP MPLS network core using Border Gateway Protocol (BGP) and MPLS Layer 3 virtual private networks (VPNs). See RFC 2547, *BGP/MPLS VPNs*.

CLNS uses network service access points (NSAPs), similar to IP addresses found in IPv4, to identify end systems (hosts) and intermediate systems (routers). ES-IS enables the hosts and routers to discover each other. IS-IS is the interior gateway protocol (IGP) that carries ISO CLNS routes through a network.

For more information about CLNS, see the ISO 8473 standards.

Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [CLNS Configuration Overview on page 66](#)
- [Understanding ES-IS for CLNS on page 69](#)
- [Understanding IS-IS for CLNS on page 71](#)
- [Understanding Static Routes for CLNS on page 74](#)

- [Understanding BGP for CLNS VPNs on page 76](#)

CLNS Configuration Overview

To configure CLNS:

1. Configure the network interfaces. See the [Junos OS Interfaces Configuration Guide for Security Devices](#).
2. If applicable, configure BGP and VPNs. See:
 - [Example: Configuring BGP for CLNS VPNs on page 77](#)
 - [MPLS Layer 2 VPN Configuration Overview on page 50](#)
 - [MPLS Layer 3 VPN Configuration Overview on page 58](#)
3. Configure a VPN routing instance. You typically configure ES-IS, IS-IS, and CLNS static routes using a VPN routing instance. See “[Example: Configuring a VPN Routing Instance for CLNS](#)” on page 67.
4. Configure one or more of the following protocols for CLNS (depending on your network).
 - ES-IS—If a device is a PE router within a CLNS island that contains any end systems, you must configure ES-IS on the device. If a CLNS island does not contain any end systems, you do not need to configure ES-IS on a device. See “[Example: Configuring ES-IS for CLNS](#)” on page 70.



NOTE: ES-IS is enabled only if either ES-IS or IS-IS is configured on the router. ES-IS must not be disabled. If ES-IS is not explicitly configured, the interface sends and receives only intermediate system hello (ISH) messages. If ES-IS is explicitly configured and disabled, the interface does not send or receive ES-IS packets. If ES-IS is explicitly configured and not disabled, the interface sends and receives ISH messages as well as ES-IS packets.

One of the interfaces that is configured for ES-IS must be configured with an ISO address for hello messages. The ISO address family must be configured on an interface to support ES-IS on that interface.

- IS-IS—You can configure IS-IS to exchange CLNS routes within a CLNS island. See “[Example: Configuring IS-IS for CLNS](#)” on page 72.



NOTE: If you have a pure CLNS island—an island that does not contain any IP devices—you must disable IPv4 and IPv6 routing. Also, to export BGP routes into IS-IS, you must configure and apply an export policy.

- Static routes—If some devices in your network do not support IS-IS, you must configure CLNS static routes. You can use static routing with or without IS-IS. You might also consider using static routes if your network is simple. See [“Example: Configuring Static Routes for CLNS” on page 74](#).
- BGP—See [“Example: Configuring BGP for CLNS VPNs” on page 77](#).



NOTE: Many of the configuration statements used to configure CLNS and routing protocols can be included at different hierarchy levels in the configuration.

Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Junos OS Routing Protocols Configuration Guide](#)
- [CLNS Overview on page 65](#)
- [Verifying a CLNS VPN Configuration on page 78](#)

Example: Configuring a VPN Routing Instance for CLNS

This example shows how to create a CLNS routing instance and set the instance type for Layer 3 VPNs.

- [Requirements on page 67](#)
- [Overview on page 67](#)
- [Configuration on page 67](#)
- [Verification on page 69](#)

Requirements

Before you begin, configure the network interfaces. See the [Junos OS Interfaces Configuration Guide for Security Devices](#).

Overview

The following example shows how to create a CLNS routing instance called `aaaa` and set the instance type to VRF for Layer 3 VPNs. Within the example, you specify that the `lo0.1` interface, `e1-2/0/0.0` interface, and `t1-3/0/0.0` interface all belong to the routing instance. The route distinguisher is set as `10.255.245.1:1` and the policy for the Layer 3 VRF table is set as `target:11111:1`.

Configuration

CLI Quick Configuration

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

```
set routing-instances aaaa instance-type vrf
set routing-instances aaaa interface lo0.1
set routing-instances aaaa interface ge-0/0/3
set routing-instances aaaa interface ge-0/0/2
set routing-instances aaaa route-distinguisher 10.255.245.1:1
set routing-instances aaaa vrf-target target:11111:1
```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see [Using the CLI Editor in Configuration Mode](#) in the *Junos OS CLI User Guide*.

To configure a VPN routing instance:

1. Create the routing instance.

```
[edit]
user@host# edit routing-instances aaaa
```
2. Specify the routing instance type.

```
[edit routing-instances aaaa]
user@host# set instance-type vrf
```
3. Specify the interfaces that belong to the routing instance.

```
[edit routing-instances aaaa]
user@host# set interface lo0.1
user@host# set interface ge-0/0/3
user@host# set interface ge-0/0/2
```
4. Specify the route distinguisher.

```
[edit routing-instances aaaa]
user@host# set route-distinguisher 10.255.245.1:1
```
5. Specify the policy for the Layer 3 VRF table.

```
[edit routing-instances aaaa]
user@host# set vrf-target target:11111:1
```
6. Enable family ISO on the interfaces edit interfaces interface-name unit-id.

```
[edit routing-instances aaaa]
user@host# set family ISO
```

Results From configuration mode, confirm your configuration by entering the **show routing-instances** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit ]
user@host# show routing-instances
instance-type vrf;
interface ge-0/0/2.0;
interface ge-0/0/3.0;
interface lo0.1;
route-distinguisher 10.255.245.1:1;
vrf-target target:11111:1;
```

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

Verification

To confirm that the configuration is working properly, perform the following task:

- [Verifying the Configured CLNS Routing Instance on page 69](#)

Verifying the Configured CLNS Routing Instance

Purpose Verify that the CLNS routing instance is configured.

Action From operational mode, enter the **show routing-instances** command.

Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [CLNS Configuration Overview on page 66](#)
- [Verifying a CLNS VPN Configuration on page 78](#)

ES-IS for CLNS

- [Understanding ES-IS for CLNS on page 69](#)
- [Example: Configuring ES-IS for CLNS on page 70](#)

Understanding ES-IS for CLNS

End System-to-Intermediate System (ES-IS) is a protocol that resolves Layer 3 ISO network service access points (NSAP) to Layer 2 addresses. ES-IS has an equivalent role as Address Resolution Protocol (ARP) in IP version 4 (IPv4).

ES-IS provides the basic interaction between Connectionless Network Service (CLNS) hosts (end systems) and routers (intermediate systems). ES-IS allows hosts to advertise NSAP addresses to other routers and hosts attached to the network. Those routers can then advertise the address to the rest of the network by using Intermediate System-to-Intermediate System (IS-IS). Routers use ES-IS to advertise their network entity title (NET) to hosts and routers that are attached to that network.

ES-IS routes are exported to Layer 1 IS-IS by default. You can also export ES-IS routes into Layer 2 IS-IS by configuring a routing policy. ES-IS generates and receives end system hello (ESH) hello messages when the protocol is configured on an interface. ES-IS is a resolution protocol that allows a network to be fully ISO integrated at both the network layer and the data layer.

The resolution of Layer 3 ISO NSAPs to Layer 2 subnetwork point of attachments (SNPAs) by ES-IS is equivalent to ARP within an IPv4 network. If a device is a provider edge (PE) router within a CLNS island that contains any end systems, you must configure ES-IS on the device.

For more information about ES-IS, see the ISO 9542 standard.

Example: Configuring ES-IS for CLNS

This example shows how to create a routing instance and enable ES-IS for CLNS on all interfaces.

- [Requirements on page 70](#)
- [Overview on page 70](#)
- [Configuration on page 70](#)
- [Verification on page 71](#)

Requirements

Before you begin, configure the network interfaces. See the [Junos OS Interfaces Configuration Guide for Security Devices](#).

Overview

The configuration instructions in this topic describe how to create a routing-instance called `aaaa`, set the end system configuration timer for the interfaces to 180, and set a preference value to 30 for ES-IS.

Configuration

CLI Quick Configuration

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

```
set routing-instances aaaa protocols esis interface all end-system-configuration-timer 180
set routing-instances aaaa protocols esis preference 30
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see [Using the CLI Editor in Configuration Mode](#) in the [Junos OS CLI User Guide](#).

To configure ES-IS for CLNS:

1. Configure the routing instance.

```
[edit]
user@host# edit routing-instances aaaa
```
2. Enable ES-IS on all interfaces.

```
[edit routing-instances aaaa]
user@host# set protocols esis interface all
```
3. Configure the end system configuration timer.

```
[edit routing-instances aaaa]
user@host# set protocols esis interface all end-system-configuration-timer 180
```
4. Configure the preference value.

```
[edit routing-instances aaaa]
```

```
user@host# set protocols esis preference 30
```

Results From configuration mode, confirm your configuration by entering the **show routing-instances** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show routing-instances
aaaa {
  protocols {
    esis {
      preference 30;
      interface all {
        end-system-configuration-timer 180;
      }
    }
  }
}
```

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

Verification

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

- [Verifying Routing-Instance for CLNS on page 71](#)
- [Verifying ES-IS for CLNS on page 71](#)

Verifying Routing-Instance for CLNS

Purpose Verify that the policy options are enabled for the routing instance.

Action From operational mode, enter the **show routing-instances** command.

Verifying ES-IS for CLNS

Purpose Verify that ES-IS is enabled.

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

IS-IS for CLNS

- [Understanding IS-IS for CLNS on page 71](#)
- [Example: Configuring IS-IS for CLNS on page 72](#)

Understanding IS-IS for CLNS

Intermediate System-to-Intermediate System (IS-IS) extensions provide the basic interior gateway protocol (IGP) support for collecting intradomain routing information for Connectionless Network Service (CLNS) destinations within a CLNS network. Routers

that learn host addresses through End System-to-Intermediate System (ES-IS) can advertise the addresses to other routers (intermediate systems) by using IS-IS.

For more information about IS-IS, see the ISO 10589 standard.

Example: Configuring IS-IS for CLNS

This example shows how to create a routing instance and enable IS-IS protocol on all interfaces.

- [Requirements on page 72](#)
- [Overview on page 72](#)
- [Configuration on page 72](#)
- [Verification on page 73](#)

Requirements

Before you begin, configure the network interfaces. See the [Junos OS Interfaces Configuration Guide for Security Devices](#).

Overview

The configuration instructions in this topic describe how to create a routing-instance called `aaaa`, enable IS-IS on all interfaces, and define BGP export policy name (`dist-bgp`), family (`ISO`), and protocol (`BP`), and apply the export policy to IS-IS.

Configuration

CLI Quick Configuration

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

```
set routing-instances aaaa protocols isis clns-routing
set routing-instances aaaa protocols isis interface all
set routing-instances aaaa protocols isis no-ipv4-routing no-ipv6-routing
set policy-options policy-statement dist-bgp from family iso protocol bgp
set policy-options policy-statement dist-bgp then accept
set routing-instances aaaa protocols isis export dist-bgp
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see [Using the CLI Editor in Configuration Mode](#) in the [Junos OS CLI User Guide](#).

To configure IS-IS for CLNS:

1. Configure the routing instance.

```
[edit]
user@host# edit routing-instances aaaa
```
2. Enable CLNS routing.

```
[edit routing-instances aaaa]
user@host# set protocols isis clns-routing
```


3. Enable IS-IS on all interfaces.

```
[edit routing-instances aaaa]
user@host# set protocols isis interface all
```
4. (Optional) Disable IPv4 and IPv6 routing to configure a pure CLNS network .

```
[edit routing-instances aaaa]
user@host# set protocols isis no-ipv4-routing no-ipv6-routing
```
5. Define the BGP export policy name, family, and protocol.

```
[edit policy-options]
user@host# set policy-statement dist-bgp from family iso protocol bgp
```
6. Define the action for the export policy.

```
[edit policy-options]
user@host# set policy-statement dist-bgp then accept
```
7. Apply the export policy to IS-IS.

```
[edit routing-instances aaaa]
user@host# set protocols isis export dist-bgp
```

Results From configuration mode, confirm your configuration by entering the **show routing-instances** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show routing-instances
aaaa {
  protocols {
    isis {
      export dist-bgp;
      no-ipv4-routing;
      no-ipv6-routing;
      clns-routing;
      interface all;
    }
  }
}
```

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

Verification

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

- [Verifying Routing-Instance for CLNS on page 73](#)
- [Verifying IS-IS for CLNS on page 74](#)

Verifying Routing-Instance for CLNS

Purpose Verify that the policy options are enabled for the routing instance.

Action From operational mode, enter the **show routing-instances** command.

Verifying IS-IS for CLNS

Purpose Verify that IS-IS is enabled.

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

Example: Configuring Static Routes for CLNS

- [Understanding Static Routes for CLNS on page 74](#)
- [Example: Configuring Static Routes for CLNS on page 74](#)

Understanding Static Routes for CLNS

The Connectionless Network Service (CLNS) is an ISO Layer 3 protocol that uses network service access point (NSAP) reachability information instead of IPv4 or IPv6 prefixes.

You can configure static routes to exchange CLNS routes within a CLNS island. A *CLNS island* is typically an IS-IS level 1 area that is part of a single IGP routing domain. An island can contain more than one area. CLNS islands can be connected by VPNs.

Example: Configuring Static Routes for CLNS

This example shows how to configure static routes for CLNS.

- [Requirements on page 74](#)
- [Overview on page 74](#)
- [Configuration on page 75](#)
- [Verification on page 76](#)

Requirements

Before you begin, configure the network interfaces. See the [Junos OS Interfaces Configuration Guide for Security Devices](#).

Overview

In this example, you configure static routes for CLNS. In the absence of an interior gateway protocol (IGP) on a certain link, a routing device might need to be configured with static routes for CLNS prefixes to be reachable by way of that link. This might be useful, for example, at an autonomous system (AS) boundary.

When you configure static routes for CLNS, consider the following tasks:

- Specify the **iso.0** routing table option to configure a primary instance CLNS static route.
- Specify the **instance-name.iso.0** routing table option to configure a CLNS static route for a particular routing instance.
- Specify the **route nsap-prefix** statement to configure the destination for the CLNS static route.

- Specify the **next-hop** (*interface-name* | *iso-net*) statement to configure the next hop, specified as an ISO network entity title (NET) or interface name.
- Include the **qualified-next-hop** (*interface-name* | *iso-net*) statement to configure a secondary backup next hop, specified as an ISO network entity title or interface name.

Configuration

CLI Quick Configuration

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

```
set routing-options rib iso.0 static iso-route 47.0005.80ff.f800.0000.ffff.ffff/152 next-hop
  47.0005.80ff.f800.0000.0108.0001.1921.6800.4212
set routing-options rib iso.0 static iso-route
  47.0005.80ff.f800.0000.0108.0001.1921.6800.4212/152 next-hop t1-0/2/2.0
set routing-options rib iso.0 static iso-route 47.0005.80ff.f800.0000.0000.0000/152
  qualified-next-hop 47.0005.80ff.f800.0000.0108.0001.1921.6800.4002 preference
  20
set routing-options rib iso.0 static iso-route 47.0005.80ff.f800.0000.0000.0000/152
  qualified-next-hop 47.0005.80ff.f800.0000.0108.0001.1921.6800.4002 metric 10
```

Step-by-Step Procedure

To configure static routes for CLNS:

1. Configure the routes.

```
[edit routing-options rib iso.0 static]
user@host# set iso-route 47.0005.80ff.f800.0000.ffff.ffff/152 next-hop
  47.0005.80ff.f800.0000.0108.0001.1921.6800.4212
user@host# set iso-route 47.0005.80ff.f800.0000.0108.0001.1921.6800.4212/152
  next-hop t1-0/2/2.0
user@host# set iso-route 47.0005.80ff.f800.0000.0000.0000/152 qualified-next-hop
  47.0005.80ff.f800.0000.0108.0001.1921.6800.4002 preference 20
user@host# set iso-route 47.0005.80ff.f800.0000.0000.0000/152 qualified-next-hop
  47.0005.80ff.f800.0000.0108.0001.1921.6800.4002 metric 10
```

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

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

Results

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

```
user@host# show routing-options
rib iso.0 {
  static {
    iso-route 47.0005.80ff.f800.0000.ffff.ffff/152 next-hop
      47.0005.80ff.f800.0000.0108.0001.1921.6800.4212;
    iso-route 47.0005.80ff.f800.0000.0108.0001.1921.6800.4212/152 next-hop t1-0/2/2.0;
    iso-route 47.0005.80ff.f800.0000.0000.0000/152 {
      qualified-next-hop 47.0005.80ff.f800.0000.0108.0001.1921.6800.4002 {
        preference 20;
      }
    }
  }
}
```

```
        metric 10;
    }
}
}
```

Verification

Confirm that the configuration is working properly.

Checking the Routing Table

Purpose Make sure that the expected routes appear in the routing table.

Action user@host> show route table iso.0

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

47.0005.80ff.f800.0000.0108.0001.1921.6800.4212/152
    *[Static/5] 00:00:25
    > via t1-0/2/2.0
47.0005.80ff.f800.0000.eee0/84
    *[Static/20] 00:04:01, metric 10, metric2 10
    > to #75 0.12.0.34.0.56 via fe-0/0/1.0
47.0005.80ff.f800.0000.ffff.ffff/104
    *[Static/5] 00:04:01, metric2 0
    > via t1-0/2/2.0
```

Meaning The static routes appear in the routing table.

BGP for CLNS VPNs

- [Understanding BGP for CLNS VPNs on page 76](#)
- [Example: Configuring BGP for CLNS VPNs on page 77](#)

Understanding BGP for CLNS VPNs

BGP extensions allow BGP to carry Connectionless Network Service (CLNS) virtual private network (VPN) network layer reachability information (NLRI) between provider edge (PE) routers. Each CLNS route is encapsulated into a CLNS VPN NLRI and propagated between remote sites in a VPN.

CLNS is a Layer 3 protocol similar to IP version 4 (IPv4). CLNS uses network service access points (NSAPs) to address end systems. This allows for a seamless autonomous system (AS) based on International Organization for Standardization (ISO) NSAPs.



NOTE: CLNS is supported for the J Series Services Router only.

A single routing domain consisting of ISO NSAP devices are considered to be CLNS islands. CLNS islands are connected together by VPNs.

You can configure BGP to exchange ISO CLNS routes between PE routers connecting various CLNS islands in a VPN using multiprotocol BGP extensions. These extensions are the ISO VPN NLRIs.

Each CLNS network island is treated as a separate VPN routing and forwarding instance (VRF) instance on the PE router.

You can configure CLNS on the global level, group level, and neighbor level.

Example: Configuring BGP for CLNS VPNs

This example shows how to create a BGP group for CLNS VPNs, define the BGP peer neighbor address for the group, and define the family.

- [Requirements on page 77](#)
- [Overview on page 77](#)
- [Configuration on page 77](#)
- [Verification on page 78](#)

Requirements

Before you begin, configure the network interfaces. See the [Junos OS Interfaces Configuration Guide for Security Devices](#).

Overview

In this example, you create the BGP group called pedge-pegde, define the BGP peer neighbor address for the group as 10.255.245.215, and define the BGP family.

Configuration

CLI Quick Configuration

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

```
set protocols bgp group pedge-pegde neighbor 10.255.245.213
set protocols bgp family iso-vpn unicast
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see [Using the CLI Editor in Configuration Mode in the Junos OS CLI User Guide](#).

To configure BGP for CLNS VPNs:

1. Configure the BGP group and define the BGP peer neighbor address.


```
[edit protocols bgp]
user@host# set group pedge-pegde neighbor 10.255.245.213
```
2. Define the family.


```
[edit protocols bgp]
user@host# set family iso-vpn unicast
```

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

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

Verification

Confirm that the configuration is working properly.

Verifying the Neighbor Status

Purpose Display information about the BGP peer.

Action From operational mode, run the **show bgp neighbor 10.255.245.213** command. Look for **iso-vpn-unicast** in the output.

```
user@host> show bgp neighbor 10.255.245.213  
Peer: 10.255.245.213+179 AS 200 Local: 10.255.245.214+3770 AS 100  
Type: External State: Established Flags: <ImportEval Sync>  
Last State: OpenConfirm Last Event: RecvKeepAlive  
Last Error: None  
Options: <Multihop Preference LocalAddress HoldTime AddressFamily PeerAS  
Rib-group Refresh>  
Address families configured: iso-vpn-unicast  
Local Address: 10.255.245.214 Holdtime: 90 Preference: 170  
Number of flaps: 0  
Peer ID: 10.255.245.213 Local ID: 10.255.245.214 Active Holdtime: 90  
Keepalive Interval: 30 Peer index: 0  
NLRI advertised by peer: iso-vpn-unicast  
NLRI for this session: iso-vpn-unicast  
Peer supports Refresh capability (2)  
Table bgp.isovpn.0 Bit: 10000  
RIB State: BGP restart is complete  
RIB State: VPN restart is complete  
Send state: in sync  
Active prefixes: 3  
Received prefixes: 3  
Suppressed due to damping: 0  
Advertised prefixes: 3  
Table aaaa.iso.0  
RIB State: BGP restart is complete  
RIB State: VPN restart is complete  
Send state: not advertising  
Active prefixes: 3  
Received prefixes: 3  
Suppressed due to damping: 0  
Last traffic (seconds): Received 6 Sent 5 Checked 5  
Input messages: Total 1736 Updates 4 Refreshes 0 Octets 33385  
Output messages: Total 1738 Updates 3 Refreshes 0 Octets 33305  
Output Queue[0]: 0  
Output Queue[1]: 0
```

Verifying a CLNS VPN Configuration

Purpose Verify that the device is configured correctly for CLNS VPNs.

Action From configuration mode in the CLI, enter the **show** command.

```
[edit]
user@host# show
interfaces {
  e1-2/0/0.0 {
    unit 0 {
      family inet {
        address 192.168.37.51/31;
      }
      family iso;
      family mpls;
    }
  }
  t1-3/0/0.0 {
    unit 0 {
      family inet {
        address 192.168.37.24/32;
      }
      family iso;
      family mpls;
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 127.0.0.1/32;
        address 10.255.245.215/32;
      }
      family iso {
        address 47.0005.80ff.f800.0000.0108.0001.1921.6800.4215.00;
      }
    }
    unit 1 {
      family iso {
        address 47.0005.80ff.f800.0000.0108.aaa2.1921.6800.4215.00;
      }
    }
  }
}
routing-options {
  autonomous-system 230;
}
protocols {
  bgp {
    group pedge-pegde {
      type internal;
      local-address 10.255.245.215;
      neighbor 10.255.245.212 {
        family iso-vpn {
          unicast;
        }
      }
    }
  }
}
policy-options {
  policy-statement dist-bgp {
```

```
        from {
            protocol bgp;
            family iso;
        }
        then accept;
    }
}
routing-instances {
    aaaa {
        instance-type vrf;
        interface lo0.1;
        interface e1-2/0/0.0;
        interface t1-3/0/0.0;
        route-distinguisher 10.255.245.1:1;
        vrf-target target:11111:1;
        routing-options {
            rib aaaa.iso.0 {
                static {
                    iso-route 47.0005.80ff.f800.0000.bbbb.1022/104
                        next-hop 47.0005.80ff.f800.0000.aaaa.1000.1921.6800.4196.00;
                }
            }
        }
    }
    protocols {
        esis {
            interface all;
        }
        isis {
            export dist-bgp;
            no-ipv4-routing;
            no-ip64-routing;
            clns-routing;
            interface all;
        }
    }
}
```

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
 - [Junos OS Routing Protocols and Policies Command Reference](#)
 - [CLNS Configuration Overview on page 66](#)

CHAPTER 5

VPLS

- [VPLS Overview on page 81](#)
- [VPLS Configuration Overview on page 85](#)
- [VPLS Interfaces on page 86](#)
- [VPLS Routing Instances on page 90](#)
- [VPLS VLAN Encapsulation on page 95](#)
- [VPLS Filters and Policers on page 102](#)
- [Example: Configuring OSPF on the VPLS PE Router on page 106](#)
- [Example: Configuring RSVP on the VPLS PE Router on page 107](#)
- [Example: Configuring MPLS on the VPLS PE Router on page 108](#)
- [Example: Configuring LDP on the VPLS PE Router on page 110](#)
- [Example: Configuring Routing Options on the VPLS PE Router on page 111](#)
- [Example: Configuring BGP on the VPLS PE Router on page 112](#)
- [Example: Configuring VPLS over GRE with IPSec VPNs on page 113](#)

VPLS Overview

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

VPLS, in its implementation and configuration, has much in common with an MPLS Layer 2 VPN. In a VPLS topology, a packet originating within a 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 an 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 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 signaled using BGP.

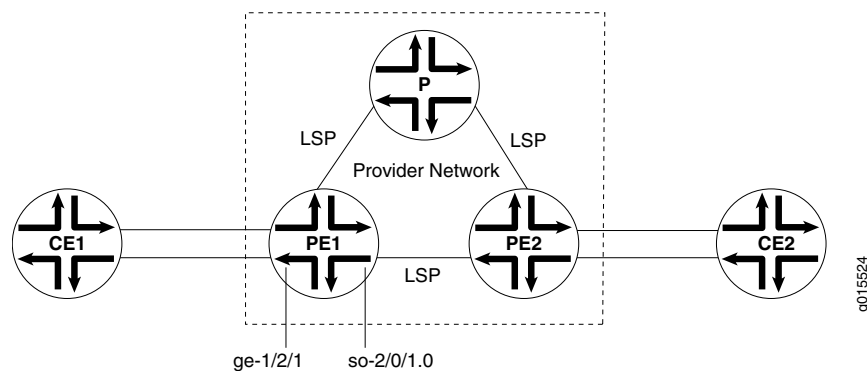
This topic contains the following sections:

- [Sample VPLS Topology on page 82](#)
- [VPLS on PE Routers on page 82](#)
- [Using an Ethernet Switch as the VPLS CE Device on page 84](#)
- [VPLS Exceptions on J Series and SRX Series Devices on page 84](#)

Sample VPLS Topology

Figure 7 on page 82 shows a basic VPLS topology.

Figure 7: Basic VPLS Topology



In this sample, the PE routers use the same autonomous system (AS). Within the AS, routing information is communicated through an interior gateway protocol (IGP). Outside the AS, routing information is shared with other ASs through BGP. The PE routers must use the same signaling protocols to communicate.

VPLS on PE Routers

Within a VPLS configuration, a device running Junos OS can act as a PE router. Junos OS passes the VPLS traffic through the following ports and PIMs on the Juniper Networks device to CE routers in the VPLS network:

- Built-in Ethernet ports on front panel
- Gigabit Ethernet uPIMs
- Gigabit Ethernet ePIMs
- Fast Ethernet PIMs
- Fast Ethernet ePIMs



NOTE: Ports on uPIMs and ePIMs must be in routing mode before you can configure the corresponding interfaces for VPLS.

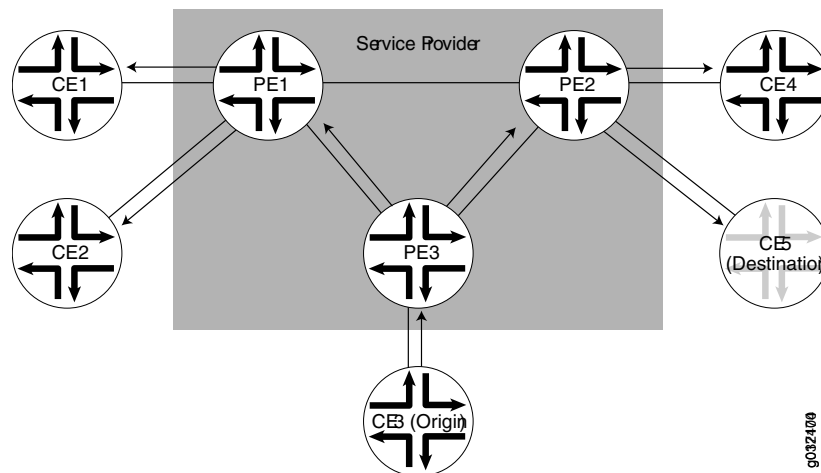
Because a VPLS carries Ethernet traffic across a service provider network, it must mimic an Ethernet network in some ways. When a PE router configured with a VPLS routing instance receives a packet from a CE device, it first determines whether it has the destination of the VPLS packet in the appropriate 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. [Figure 8 on page 83](#) illustrates this process.

Figure 8: Flooding a Packet with an Unknown Destination



A VPLS interface can be directly connected to an Ethernet switch. Layer 2 information gathered by an Ethernet switch, for example, MAC addresses and interface ports, is included in the VPLS routing instance table.

An MPLS label-switched interface (LSI) label is used as the inner label for VPLS. This label maps to a VPLS routing instance on the ingress PE router. On the egress 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.



NOTE: Under certain circumstances, VPLS PE routers might duplicate an Internet Control Message Protocol (ICMP) reply from a CE device 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 device with promiscuous mode enabled is connected to a PE router. The PE router automatically floods the promiscuous mode enabled CE device, which then returns the ICMP request to the VPLS PE routers. The VPLS PE routers consider the ICMP request to be new and flood the request again, creating a duplicate ping reply.

Using an Ethernet Switch as the VPLS 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, be aware of the following configuration issues:

- 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.
- Junos OS 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.

VPLS Exceptions on J Series and SRX Series Devices

The VPLS implementation on a J Series or SRX Series device is similar to VPLS implementations on M Series, T Series, and MX Series routers, with the following exceptions:

- J Series or SRX Series devices do not support aggregated Ethernet interfaces. Therefore, aggregated Ethernet interfaces between CE devices and PE routers are not supported for VPLS routing instances on J Series or SRX Series devices.
- VPLS routing instances on J Series or SRX Series devices use BGP to send signals to other PE routers. LDP signaling is not supported.
- VPLS multihoming, which allows connecting a CE device to multiple PE routers to provide redundant connectivity, is not supported on J Series or SRX Series devices.
- J Series or SRX Series devices do not support BGP mesh groups.
- J Series or SRX Series devices support only the following encapsulation types on VPLS interfaces that face CE devices: extended VLAN VPLS, Ethernet VPLS, and VLAN VPLS. Ethernet VPLS over ATM LLC encapsulation is not supported.

- Virtual ports are generated dynamically on a Tunnel Services PIC on some Juniper Networks routing platforms. J Series or SRX Series devices do not support Tunnel Services modules or virtual ports.
- The VPLS implementation on J Series or SRX Series devices does not support dual-tagged frames. Therefore, VLAN rewrite operations are not supported on dual-tagged frames. VLAN rewrite operations such as pop-pop, pop-swap, push-push, swap-push, and swap-swap, which are supported on M Series and T Series routing platforms, are not supported on J Series or SRX Series devices.
- Firewall filters for VPLS are not supported.

Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [MPLS Layer 2 VPN Configuration Overview on page 50](#)
- [Understanding VPLS Interfaces on page 86](#)
- [Understanding VPLS Routing Instances on page 90](#)
- [Understanding VPLS VLAN Encapsulation on page 95](#)
- [Understanding VPLS VLAN Encapsulation on a Logical Interface on page 97](#)
- [VPLS Configuration Overview on page 85](#)

VPLS Configuration Overview

To configure VPLS functionality, you must enable VPLS support on the provider edge (PE) routers. 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) devices, as explained in the steps that follow.



NOTE: Many configuration procedures for VPLS are identical to the procedures for Layer 2 and Layer 3 VPNs.

To configure VPLS:

1. Determine which uPIM and ePIM ports correspond to the interfaces that will carry the VPLS traffic and enable routing mode on those ports.
2. Configure the interfaces that will carry the VPLS traffic between the PE router and CE devices. On the PE router interfaces that are facing the CE devices, specify a VPLS encapsulation type. The type of encapsulation depends on the interface type. See [“Example: Configuring Routing Interfaces on the VPLS PE Router” on page 88](#) and [“Example: Configuring the Interface to the VPLS CE Device” on page 89](#).
3. Create a VPLS routing instance on each PE router that is participating in the VPLS. For each VPLS routing instance, specify which interfaces will carry the VPLS traffic between the PE and CE devices. On the CE device interface that faces the PE router, you must specify inet (for IPv4), and include the IP address. Additionally, each routing instance must have a unique route distinguisher associated with it. (VPN routing

instances need a route distinguisher to help BGP identify overlapping network layer reachability information (NLRI) messages from different VPNs.) See [“Example: Configuring the VPLS Routing Instance”](#) on page 93.

4. Configure routing options on the PE router. See [“Example: Configuring Routing Options on the VPLS PE Router”](#) on page 111.
5. Configure MPLS LSPs between the PE routers. See [“Example: Configuring MPLS on the VPLS PE Router”](#) on page 108.
6. Configure RSVP on the PE routers. Enable RSVP for all connections that participate in the MPLS LSP. See [“Example: Configuring RSVP on the VPLS PE Router”](#) on page 107.
7. Configure an IBGP session between PE routers so that the routers can exchange information about routes originating and terminating in the VPLS. See [“Example: Configuring BGP on the VPLS PE Router”](#) on page 112.
8. Configure an IGP on the PE routers to exchange routing information. See [“Example: Configuring OSPF on the VPLS PE Router”](#) on page 106.
9. Configure VLAN encapsulation. See [“Example: Configuring VPLS VLAN Encapsulation on Gigabit Ethernet Interfaces”](#) on page 96, [“Example: Configuring VPLS VLAN Encapsulation”](#) on page 97, and [“Example: Configuring Extended VLAN VPLS Encapsulation”](#) on page 100.

Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [MPLS Layer 2 VPN Configuration Overview](#) on page 50
- [MPLS Layer 2 VPN Configuration Overview](#) on page 50
- [Example: Configuring MPLS on the VPLS PE Router](#) on page 108
- [Example: Configuring RSVP on the VPLS PE Router](#) on page 107
- [Example: Configuring BGP on the VPLS PE Router](#) on page 112
- [Example: Configuring OSPF on the VPLS PE Router](#) on page 106

VPLS Interfaces

- [Understanding VPLS Interfaces](#) on page 86
- [Example: Configuring Routing Interfaces on the VPLS PE Router](#) on page 88
- [Example: Configuring the Interface to the VPLS CE Device](#) on page 89

Understanding VPLS Interfaces

For each VPLS routing instance on a PE router, you specify which interfaces are to be used to carry VPLS traffic between the PE and CE devices.

This topic contains the following sections:

- [Interface Name](#) on page 87
- [Encapsulation Type](#) on page 87

- [Flexible VLAN Tagging on page 87](#)
- [VLAN Rewrite on page 88](#)

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-1/0/1 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.

Encapsulation Type

The physical link-layer encapsulation type for a VPLS interface can be one of the following:

- **ethernet-vpls**—Use Ethernet VPLS encapsulation on Ethernet interfaces that have VPLS enabled and that must accept packets carrying standard Tag Protocol Identifier (TPID) values.
- **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. All VLAN IDs from 1 through 1023 are valid for VPLS VLANs on Fast Ethernet interfaces, and all VLAN IDs from 1 through 4094 are valid for VPLS VLANs on Gigabit Ethernet interfaces.
- **vlan-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. You must configure this encapsulation type on both the physical interface and the logical interface. VLAN IDs 1 through 511 are reserved for normal Ethernet VLANs, IDs 512 through 1023 are reserved for VPLS VLANs on Fast Ethernet interfaces, and IDs 512 through 4094 are reserved for VPLS VLANs on Gigabit Ethernet interfaces.
- **flexible-ethernet-services**—Use flexible Ethernet services encapsulation when you want to configure multiple per-unit Ethernet encapsulations. This encapsulation type allows you to configure any combination of route, TCC, CCC, and VPLS encapsulations on a single physical port. Aggregated Ethernet bundles cannot use this encapsulation type.

For flexible Ethernet services encapsulation, VLAN IDs from 1 through 511 are no longer reserved for normal VLANs.

Flexible VLAN Tagging

For untagged packets to be accepted on an 802.1Q VLAN-tagged port, specify the native VLAN ID with the flexible VLAN tagging option. (No other flexible VLAN tagging features are supported.)

VLAN Rewrite

You can rewrite VLAN tags on VPLS interfaces. Rewriting VLAN tags allows you to use an additional (outer) VLAN tag to differentiate between CE devices that share a VLAN ID.

You can configure rewrite operations to stack (push), remove (pop), or rewrite (swap) tags on single-tagged frames. If a port is not configured for VLAN tagging, rewrite operations are not supported on any logical interface on that port.

You can configure the following VLAN rewrite operations:

- **pop**—Remove a VLAN tag from the top of the VLAN tag stack. The outer VLAN tag of the frame is removed.
- **push**—Add a new VLAN tag to the top of the VLAN stack. An outer VLAN tag is pushed in front of the existing VLAN tag.
- **swap**—Replace the VLAN tag at the top of the VLAN tag stack with a user-specified VLAN tag value.

You perform VLAN rewrite operations by applying input and output VLAN maps at the ingress and egress, respectively, of the interface. For incoming frames, use the `input-vlan-map`; for outgoing frames, use the `output-vlan-map`.

The VPLS implementation on J Series or SRX Series devices does not support dual-tagged frames. Therefore, VLAN rewrite operations are not supported on dual-tagged frames. VLAN rewrite operations such as pop-pop, pop-swap, push-push, swap-push, and swap-swap, which are supported on M Series and T Series routing platforms, are not supported on J Series or SRX Series devices.

Example: Configuring Routing Interfaces on the VPLS PE Router

This example shows how to configure routing interfaces on the VPLS PE router.

- [Requirements on page 88](#)
- [Overview on page 88](#)
- [Configuration on page 89](#)
- [Verification on page 89](#)

Requirements

Before you begin, see Understanding Selective Stateless Packet-Based Services in the *Junos OS Security Configuration Guide*.

Overview

In this example, you configure the PE1 router loopback interface and the interface to the PE2 router ge-2/0/1.

Configuration

Step-by-Step Procedure

To configure the routing interface on the VPLS PE router:

1. Configure the loopback interface.

```
[edit]
user@host# set interfaces lo0 unit 0 family inet address 10.255.7.168/32 primary
```
2. Configure the IP address on the MPLS core interface.

```
[edit]
user@host# set interfaces ge-3/0/2 unit 0 family inet address 100.1.1.1/30
```
3. Configure the MPLS family.

```
[edit]
user@host# set interfaces ge-3/0/2 unit 0 family mpls
```
4. If you are done configuring the device, commit the configuration.

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

Verification

To verify the configuration is working properly, enter the **show interfaces** command.

Example: Configuring the Interface to the VPLS CE Device

This example shows how to configure the router interface that is connected to the CE device to include VPLS encapsulation.

- [Requirements on page 89](#)
- [Overview on page 89](#)
- [Configuration on page 89](#)
- [Verification on page 90](#)

Requirements

Before you begin, see Understanding Selective Stateless Packet-Based Services in the *Junos OS Security Configuration Guide*.

Overview

In this example, you configure the router interface ge-1/2/1 that is connected to the CE device to include VPLS encapsulation.

Configuration

Step-by-Step Procedure

To configure the interface to the VPLS CE device:

1. Configure VPLS encapsulation for the interface facing the CE router.

```
[edit]
user@host# set interfaces ge-1/2/1 encapsulation ethernet-vpls
```

2. Configure the interface for the VPLS family group.

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

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

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

Verification

To verify the configuration is working properly, enter the **show interfaces ge-1/2/1** command.

VPLS Routing Instances

- [Understanding VPLS Routing Instances on page 90](#)
- [Example: Configuring the VPLS Routing Instance on page 93](#)

Understanding VPLS Routing Instances

To configure VPLS functionality, you must enable VPLS support on the 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 CE devices.

You create a VPLS routing instance on each PE router that is participating in the VPLS. The routing instance has the same name on each PE router. To configure the VPLS routing instance, you specify the following:

- Route distinguisher—Helps BGP distinguish between potentially identical network layer reachability information (NLRI) messages received from different VPLS instances. Each routing instance that you configure on a PE router must have a unique route distinguisher.
- Route target—Defines which route is part of a VPLS. A unique route target helps distinguish between different VPLS services on the same router.
- Site name—Provides unique name for the VPLS site.
- Site identifier—Provides unique numerical identifier for the VPLS site.
- Site range—Specifies total number of sites in the VPLS. The site range must be greater than the site identifier.
- Interface to the CE router—Specifies the physical interface to the CE router that carries VPLS traffic. The interface must be configured for a VPLS encapsulation type.



NOTE: In addition to the VPLS routing instance, you must configure 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 routers.



CAUTION: MPLS is disabled by default on J Series and SRX Series devices. You must explicitly configure your router to allow MPLS traffic. However, when MPLS is enabled, all flow-based security features are deactivated and the router performs packet-based processing. Flow-based services such as security policies, zones, NAT, ALGs, chassis clustering, screens, firewall authentication, and IPsec VPNs are unavailable on the router.

This topic contains the following sections:

- [BGP Signaling on page 91](#)
- [VPLS Routing Table on page 91](#)
- [Trace Options on page 92](#)

BGP Signaling

BGP is used to signal the paths between each of the PE routers participating in the VPLS routing instance. These paths carry VPLS traffic across the service provider's network between the VPLS sites.



NOTE: LDP signaling is not supported for the VPLS routing instance.

To configure BGP signaling, you specify the following:

- VPLS site name and site identifier—When you configure BGP signaling for the VPLS routing instance, you must specify each VPLS site that has a connection to the router. For each VPLS site, you must configure a site name and site identifier (a numerical identifier between 1 to 65,534 that uniquely identifies the VPLS site).
- Site range—When you enable BGP signaling for the VPLS routing instance, you need to configure a site range. The site range specifies the total number of sites in the VPLS.



NOTE: The site range value must be greater than the largest site identifier.

- 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.

VPLS Routing Table

The VPLS routing table contains MAC addresses and interface information for both physical and virtual ports. You can configure the following characteristics for the table:

- Table size—You can modify the size of the VPLS 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.

The interfaces affected include all of the interfaces within the VPLS routing instance, including the local interfaces and the LSI interfaces.

- **Timeout interval**—You can modify the timeout interval for the VPLS table. The default timeout interval is 300 seconds; the minimum is 10 seconds, and the maximum is 1,000,000 seconds. 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.
- **Number of addresses learned from an interface**—You can configure a limit on the number of MAC addresses learned by a VPLS routing instance by setting the MAC table size. The default is 512 addresses; the minimum is 16, 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.

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 all interfaces configured for a VPLS routing instance, as well as limit the number of MAC addresses learned from a specific interface.

The MAC limit configured for an individual interface overrides the limit configured for all interfaces for the VPLS routing instance. Also, the table limit can override the limits configured for the interfaces.

The MAC address limit applies only to interfaces to CE devices.

Trace Options

The following trace flags display 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

Example: Configuring the VPLS Routing Instance

This example shows how to create a VPLS routing instance on each PE router that is participating in the VPLS.

- [Requirements on page 93](#)
- [Overview on page 93](#)
- [Configuration on page 93](#)
- [Verification on page 95](#)

Requirements

Before you begin:

- Before you begin, see Understanding Selective Stateless Packet-Based Services in the *Junos OS Security Configuration Guide*.
- Configure the interfaces that will carry the VPLS traffic between the PE router and the CE devices. See “[Example: Configuring Routing Interfaces on the VPLS PE Router](#)” on page 88 and “[Example: Configuring the Interface to the VPLS CE Device](#)” on page 89.

Overview

This example describes how to create a VPLS routing instance; configure VPLS site identifier, site range, no tunnel services option, route distinguisher, and route target for the VPLS routing instance; and specify the VPLS interface to the CE router.



NOTE: You must specify no tunnel services in the VPLS routing instance configuration, because J Series and SRX Series devices do not support tunnel serial PICs.

Configuration

CLI Quick Configuration

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

```
set routing-instances green instance-type vpls
set routing-instances green protocols vpls site-range 10 site R3 site-identifier 2
set routing-instances green protocols vpls no-tunnel-services
set routing-instances green route-distinguisher 10.255.7.1:1
set routing-instances green vrf-target target:1111:1
set routing-instances green instance-type vpls interface ge-1/2/1.0
```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see Using the CLI Editor in Configuration Mode in the [Junos OS CLI User Guide](#).

To configure a VPLS routing instance:

1. Configure the routing instance of type VPLS.

```
[edit]
user@host# edit routing-instances green
```
2. Enable the VPLS instance type.

```
[edit routing-instances green]
user@host# set instance-type vpls
```
3. Configure the VPLS site identifier and range for the VPLS routing instance.

```
[edit routing-instances green protocols vpls]
user@host# set site-range 10 site R3 site-identifier 2
```
4. Configure the no-tunnel-services option for the VPLS routing instance.

```
[edit routing-instances green protocols vpls]
user@host# set no-tunnel-services
```
5. Configure the route distinguisher.

```
[edit routing-instances green]
user@host# set route-distinguisher 10.255.7.1:1
```
6. Configure the route target.

```
[edit routing-instances green]
user@host# set vrf-target target:11111:1
```
7. Specify the VPLS interface to the CE router.

```
[edit routing-instances green]
user@host# set instance-type vpls interface ge-1/2/1.0
```

Results From configuration mode, confirm your configuration by entering the **show routing-instances green** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show routing-instances green
instance-type vpls;
interface ge-1/2/1.0;
route-distinguisher 10.255.7.1:1;
vrf-target target:11111:1;
protocols {
  vpls {
    site-range 10;
    no-tunnel-services;
    site R3 {
      site-identifier 2;
    }
  }
}
```

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

Verification

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

- [Verifying VPLS Routing Instance Is Configured on page 95](#)
- [Verifying VPLS Routing Attributes Are Configured on page 95](#)

Verifying VPLS Routing Instance Is Configured

Purpose Verify that the VPLS routing instance is configured.

Action From operational mode, enter the **show routing-instances** command.

Verifying VPLS Routing Attributes Are Configured

Purpose Verify that attributes such as VPLS site identifier, site range, no tunnel services option, route distinguisher, and route target for the VPLS routing instance are configured.

Action From operational mode, enter the **show routing-instances green protocols vpls** command.

VPLS VLAN Encapsulation

- [Understanding VPLS VLAN Encapsulation on page 95](#)
- [Example: Configuring VPLS VLAN Encapsulation on Gigabit Ethernet Interfaces on page 96](#)
- [Understanding VPLS VLAN Encapsulation on a Logical Interface on page 97](#)
- [Example: Configuring VPLS VLAN Encapsulation on page 97](#)
- [Example: Configuring Extended VLAN VPLS Encapsulation on page 100](#)

Understanding VPLS VLAN Encapsulation

Gigabit Ethernet IQ, Gigabit Ethernet PIMs with small form-factor pluggable optics (SFPs), SRX Series devices with Gigabit Ethernet, J Series devices with Gigabit Ethernet, Tri-Rate Ethernet copper, and 10-Gigabit Ethernet interfaces with VLAN tagging enabled can use flexible Ethernet services, VLAN virtual private LAN service (VPLS) encapsulation.



NOTE: VLAN encapsulation is not supported on SRX100 devices because there is no Gigabit Ethernet port.

Aggregated Ethernet interfaces configured for VPLS can use Ethernet VPLS or VLAN VPLS.

Example: Configuring VPLS VLAN Encapsulation on Gigabit Ethernet Interfaces

This example shows how to configure the VPLS VLAN encapsulation on either a Gigabit Ethernet IQ or Gigabit Ethernet physical interface.

- [Requirements on page 96](#)
- [Overview on page 96](#)
- [Configuration on page 96](#)
- [Verification on page 97](#)

Requirements

Before you begin:

- Before you begin, see Understanding Selective Stateless Packet-Based Services in the *Junos OS Security Configuration Guide*.
- Configure the interfaces that will carry the VPLS traffic between the PE router and the CE devices. See “[Example: Configuring Routing Interfaces on the VPLS PE Router](#)” on page 88 and “[Example: Configuring the Interface to the VPLS CE Device](#)” on page 89.
- Create a VPLS routing instance on each PE router that is participating in the VPLS. See “[Example: Configuring the VPLS Routing Instance](#)” on page 93.
- Configure an IGP on the PE routers to exchange routing information. See “[Example: Configuring OSPF on the VPLS PE Router](#)” on page 106.
- Configure RSVP-TE, see “[Example: Configuring RSVP on the VPLS PE Router](#)” on page 107 and then MPLS LSPs on the PE routers, see “[Example: Configuring MPLS on the VPLS PE Router](#)” on page 108. Alternatively configure LDP on the PE routers, see “[Example: Configuring LDP on the VPLS PE Router](#)” on page 110.
- Configure routing options on the PE router. See “[Example: Configuring Routing Options on the VPLS PE Router](#)” on page 111.
- Configure an IBGP session between PE routers so that the routers can exchange information about routes originating and terminating in the VPLS. See “[Example: Configuring BGP on the VPLS PE Router](#)” on page 112

Overview

This example describes how to configure Ethernet VPLS encapsulation on a Gigabit Ethernet IQ or Gigabit Ethernet physical interface and enable the VPLS family on the interface.

Configuration

Step-by-Step Procedure

To configure VPLS VLAN encapsulation on a Gigabit Ethernet IQ or Gigabit Ethernet physical interface:

1. Configure the ethernet-vpls encapsulation on the interface.

[edit]
user@host# set interfaces ge-3/0/6 encapsulation ethernet-vpls

2. Enable the VPLS family on the interface.

```
[edit ]  
user@host# set interfaces ge-3/0/6 unit 0 family vpls
```
3. If you are done configuring the device, commit the configuration.

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

Verification

To verify the configuration is working properly, enter the **show interfaces** command.

Understanding VPLS VLAN Encapsulation on a Logical Interface

You cannot configure a logical interface with VLAN VLAN VPLS encapsulation unless you also configure the physical device with the same encapsulation or with flexible Ethernet services encapsulation. In general, the logical interface must have a VLAN ID of 512 or higher; if the VLAN ID is 511 or lower, it will be subject to the normal destination filter lookups in addition to source address filtering. However if you configure flexible Ethernet services encapsulation, this VLAN ID restriction is removed.

Ethernet interfaces in VLAN mode can have multiple logical interfaces. In VPLS mode, VLAN IDs from 1 through 511 are reserved for normal VLANs, and VLAN IDs 512 through 4094 are reserved for VPLS VLAN. For 4-port Fast Ethernet interfaces, you can use VLAN IDs 512 through 1024 for VPLS VLAN.

For encapsulation type **flexible-ethernet-services**, all VLAN IDs are valid.

Example: Configuring VPLS VLAN Encapsulation

This example shows how to configure VPLS VLAN encapsulation and enable it on the physical and the logical interfaces.

- [Requirements on page 97](#)
- [Overview on page 98](#)
- [Configuration on page 98](#)
- [Verification on page 99](#)

Requirements

Before you begin:

- Before you begin, see Understanding Selective Stateless Packet-Based Services in the *Junos OS Security Configuration Guide*.
- Configure the interfaces that will carry the VPLS traffic between the PE router and the CE devices. See “[Example: Configuring Routing Interfaces on the VPLS PE Router](#)” on page 88 and “[Example: Configuring the Interface to the VPLS CE Device](#)” on page 89.
- Create a VPLS routing instance on each PE router that is participating in the VPLS. See “[Example: Configuring the VPLS Routing Instance](#)” on page 93.

- Configure an IGP on the PE routers to exchange routing information. See [“Example: Configuring OSPF on the VPLS PE Router”](#) on page 106.
- Configure RSVP-TE, see [“Example: Configuring RSVP on the VPLS PE Router”](#) on page 107 and then MPLS LSPs on the PE routers, see [“Example: Configuring MPLS on the VPLS PE Router”](#) on page 108. Alternatively configure LDP on the PE routers, see [“Example: Configuring LDP on the VPLS PE Router”](#) on page 110.
- Configure routing options on the PE router. See [“Example: Configuring Routing Options on the VPLS PE Router”](#) on page 111.
- Configure an IBGP session between PE routers so that the routers can exchange information about routes originating and terminating in the VPLS. See [“Example: Configuring BGP on the VPLS PE Router”](#) on page 112.

Overview

This example describes how to enable VLAN tagging on VPLS interface ge-3/0/6, configure the encapsulation type on the physical and logical interfaces, and configure the VPLS family on the logical interface.



NOTE: Perform the following CLI quick configuration and procedures on all of the PE interfaces (CE facing).

Configuration

CLI Quick Configuration

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

```
set interfaces ge-3/0/6 vlan-tagging
set interfaces ge-3/0/6 encapsulation vlan-vpls
set interfaces ge-3/0/6 unit 0 encapsulation vlan-vpls
set interfaces ge-3/0/6 unit 0 vlan-id 512
set interfaces ge-3/0/6 unit 0 family vpls
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see [Using the CLI Editor in Configuration Mode](#) in the *Junos OS CLI User Guide*.

To configure VPLS VLAN encapsulation:

1. Enable VLAN tagging on the VPLS interface.

```
[edit interfaces ge-3/0/6]
user@host# set vlan-tagging
```
2. Configure the encapsulation type on the physical interface.

```
[edit interfaces ge-3/0/6]
user@host# set interfaces ge-3/0/6 encapsulation vlan-vpls
```
3. Configure the encapsulation type on the logical interface.

```
[edit interfaces ge-3/0/6 unit 0]
user@host# set encapsulation vlan-vpls
```

4. Configure the VLAN ID on the logical interface.

```
[edit interfaces ge-3/0/6 unit 0]
user@host# set vlan-id 512
```

5. Configure the family VPLS on the logical interface.

```
[edit interfaces ge-3/0/6 unit 0]
user@host# set family vpls
```

Results From configuration mode, confirm your configuration by entering the **show interfaces ge-3/0/6** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces ge-3/0/6
vlan-tagging;
encapsulation vlan-vpls;
unit 0 {
    encapsulation vlan-vpls;
    vlan-id 512;
    family vpls;
}
```

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

Verification

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

- [Verifying VPLS VLAN Encapsulation on page 99](#)
- [Verifying VPLS VLAN Encapsulation for Logical Interfaces on page 99](#)

Verifying VPLS VLAN Encapsulation

Purpose Verify that the VPLS VLAN encapsulation is enabled at the interfaces.

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

Verifying VPLS VLAN Encapsulation for Logical Interfaces

Purpose Verify that the VPLS VLAN encapsulation is enabled at the logical interface.

Action From operational mode, enter the **show interfaces ge-3/0/6 unit 0** command.

Example: Configuring Extended VLAN VPLS Encapsulation

This example shows how to configure extended VLAN VPLS encapsulation and enable it on the physical and the logical interfaces.

- [Requirements on page 100](#)
- [Overview on page 100](#)
- [Configuration on page 101](#)
- [Verification on page 102](#)

Requirements

Before you begin:

- Before you begin, see Understanding Selective Stateless Packet-Based Services in the *Junos OS Security Configuration Guide*.
- Configure the interfaces that will carry the VPLS traffic between the PE router and the CE devices. See [“Example: Configuring Routing Interfaces on the VPLS PE Router” on page 88](#) and [“Example: Configuring the Interface to the VPLS CE Device” on page 89](#).
- Create a VPLS routing instance on each PE router that is participating in the VPLS. See [“Example: Configuring the VPLS Routing Instance” on page 93](#).
- Configure an IGP on the PE routers to exchange routing information. See [“Example: Configuring OSPF on the VPLS PE Router” on page 106](#).
- Configure RSVP-TE, see [“Example: Configuring RSVP on the VPLS PE Router” on page 107](#) and then MPLS LSPs on the PE routers, see [“Example: Configuring MPLS on the VPLS PE Router” on page 108](#). Alternatively configure LDP on the PE routers, see [“Example: Configuring LDP on the VPLS PE Router” on page 110](#).
- Configure routing options on the PE router. See [“Example: Configuring Routing Options on the VPLS PE Router” on page 111](#).
- Configure an IBGP session between PE routers so that the routers can exchange information about routes originating and terminating in the VPLS. See [“Example: Configuring BGP on the VPLS PE Router” on page 112](#).

Overview

This example describes how to enable VLAN tagging on the VPLS interface ge-3/0/6, configure the extended-vlan-vpls type on the physical and logical interfaces, and configure the VPLS family on the logical interface.



NOTE: Perform the following CLI quick configurations and procedures on all PE interfaces (CE facing).

Configuration

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

```
set interfaces ge-3/0/6 vlan-tagging
set interfaces ge-3/0/6 encapsulation extended-vlan-vpls
set interfaces ge-3/0/6 unit 0 vlan-id 100
set interfaces ge-3/0/6 unit 0 family vpls
```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see Using the CLI Editor in Configuration Mode in the [Junos OS CLI User Guide](#).

To configure extended VPLS VLAN encapsulation:

1. Enable VLAN tagging on the VPLS interface as it will receive tagged packets from CE.

```
[edit interfaces ge-3/0/6]
user@host# set vlan-tagging
```

2. Configure the encapsulation type on the physical interface.

```
[edit interfaces ge-3/0/6]
user@host# set interfaces ge-3/0/6 encapsulation vlan-vpls
```

3. Configure the VLAN ID on the logical interface.

```
[edit interfaces ge-3/0/6 unit 0]
user@host# set encapsulation vlan-vpls vlan-id 100
```

4. Configure the VPLS family on the logical interface.

```
[edit interfaces ge-3/0/6 unit 0]
user@host# set family vpls
```

Results From configuration mode, confirm your configuration by entering the **show interfaces ge-3/0/6** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces ge-3/0/6
vlan-tagging;
encapsulation extended-vlan-vpls;
unit 0 {
  encapsulation vlan-vpls;
  vlan-id 100;
  family vpls;
}
```

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

Verification

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

- [Verifying Extended VLAN VPLS Encapsulation on page 102](#)
- [Verifying Extended VLAN VPLS Encapsulation for Logical Interfaces on page 102](#)

Verifying Extended VLAN VPLS Encapsulation

Purpose Verify that the extended VLAN VPLS encapsulation is enabled at the interfaces.

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

Verifying Extended VLAN VPLS Encapsulation for Logical Interfaces

Purpose Verify that the extended VLAN VPLS encapsulation is enabled at the logical interface.

Action From operational mode, enter the **show interfaces ge-3/0/6 unit 0** command.

VPLS Filters and Policers

- [VPLS Filters and Policers Overview on page 102](#)
- [Example: Configuring VPLS Policers on page 102](#)
- [Example: Configuring VPLS Filters on page 104](#)

VPLS Filters and Policers Overview

This feature permits users to configure both firewall filters and policers for virtual private LAN service (VPLS). Firewall filters enable you to filter packets based on their components and perform an action on packets that match the filter. Policers enable you to limit the amount of traffic that passes into or out of an interface.

This feature can be enabled by configuring VPLS filters, policers, and accounting through various CLI commands. VPLS filters and policers act on a Layer 2 frame that includes the media access control (MAC) header (after any VLAN rewrite or other rules are applied), but that does not include the cyclical redundancy check (CRC) field.



NOTE: You can apply VPLS filters and policers on the PE routers only to customer-facing (PE-CE) interfaces.

Example: Configuring VPLS Policers

This example shows how to configure VPLS policers.

- [Requirements on page 103](#)
- [Overview on page 103](#)

- [Configuration on page 103](#)
- [Verification on page 104](#)

Requirements

Before you begin:

- Configure the interfaces that will carry the VPLS traffic between the PE router and the CE devices. See [“Example: Configuring Routing Interfaces on the VPLS PE Router” on page 88](#) and [“Example: Configuring the Interface to the VPLS CE Device” on page 89](#).
- Create a VPLS routing instance on each PE router that is participating in the VPLS. See [“Example: Configuring the VPLS Routing Instance” on page 93](#).
- Configure an IGP on the PE routers to exchange routing information. See [“Example: Configuring OSPF on the VPLS PE Router” on page 106](#).
- Configure RSVP-TE on the PE routers. See [“Example: Configuring RSVP on the VPLS PE Router” on page 107](#).

Overview

This example describes how to configure policing and apply it on the interface for VPLS.



CAUTION: MPLS is disabled by default on J Series and SRX Series devices. You must explicitly configure your device to allow MPLS traffic. However, when MPLS is enabled, all flow-based security features are deactivated and the device performs packet-based processing. Flow-based services such as security policies, zones, NAT, ALGs, chassis clustering, screens, firewall authentication, and IPsec VPNs are unavailable on the device.

Configuration

CLI Quick Configuration

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

```
set firewall policer police2 if-exceeding bandwidth-percent 10
set firewall policer police2 if-exceeding burst-size-limit 1500
set firewall policer police2 then discard
set interfaces ge-0/0/1 unit 512 family vpls policer input police2
```

Step-by-Step Procedure

To configure filters for VPLS:

1. Configure bandwidth percentage.
[edit]
user@host# set firewall policer police2 if-exceeding bandwidth-percent 10
2. Configure the burst size limit.
[edit]

```
user@host# set firewall policer police2 if-exceeding burst-size-limit 1500
```

3. Configure the terminal action on the packet.

```
[edit ]
user@host# set firewall policer police2 then discard
```

4. Apply the policer to the interface.

```
[edit ]
user@host# set interfaces ge-0/0/1 unit 512 family vpls policer input police2
```

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

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

Verification

To verify the configuration is working properly, enter the following command:

```
show firewall < >
```

Example: Configuring VPLS Filters

This example shows how to configure VPLS filters.

- [Requirements on page 104](#)
- [Overview on page 104](#)
- [Configuration on page 105](#)
- [Verification on page 106](#)

Requirements

Before you begin:

- Configure the interfaces that will carry the VPLS traffic between the PE router and the CE devices. See [“Example: Configuring Routing Interfaces on the VPLS PE Router” on page 88](#) and [“Example: Configuring the Interface to the VPLS CE Device” on page 89](#).
- Create a VPLS routing instance on each PE router that is participating in the VPLS. See [“Example: Configuring the VPLS Routing Instance” on page 93](#).
- Configure an IGP on the PE routers to exchange routing information. See [“Example: Configuring OSPF on the VPLS PE Router” on page 106](#).
- Configure RSVP-TE on the PE routers. See [“Example: Configuring RSVP on the VPLS PE Router” on page 107](#).

Overview

This example describes how to configure filtering and accounting for VPLS.



CAUTION: MPLS is disabled by default on J Series and SRX Series devices. You must explicitly configure your device to allow MPLS traffic. However,

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

Configuration

CLI Quick Configuration

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

```
set firewall family vpls filter blue term term1 from interface ge-3/0/0.512
set firewall family vpls filter blue term term1 from interface fe-5/0/0.512
set firewall family vpls filter blue term term1 then count count1
set firewall family vpls filter blue accounting-profile fw_profile
set accounting-options file fw_acc size 500k
set accounting-options file fw_acc transfer-interval 5
set accounting-options filter-profile fw_profile file fw_acc
set accounting-options filter-profile fw_profile interval 1
set accounting-options filter-profile fw_profile counters count1
set interfaces ge-0/0/1 unit 512 family vpls filter input blue
```

Step-by-Step Procedure

To configure filters for VPLS:

1. Configure a filter with a GE interface as the match condition and count as the action.

```
[edit ]
user@host# set firewall family vpls filter blue term term1 from interface ge-3/0/0.512
```
2. Configure a filter with an FE interface as the match condition and count as the action.

```
[edit ]
user@host# set firewall family vpls filter blue term term1 from interface fe-5/0/0.512
```
3. Configure the count.

```
[edit ]
user@host# set firewall family vpls filter blue term term1 then count count1
```
4. Configure the accounting profile to refer it to the counter.

```
[edit ]
user@host# set firewall family vpls filter blue accounting-profile fw_profile
```
5. Configure the account file size.

```
[edit ]
user@host# set accounting-options file fw_acc size 500k
```
6. Configure the account transfer interval.

```
[edit ]
user@host# set accounting-options file fw_acc transfer-interval 5
```
7. Configure the filter for the accounting profile.

```
[edit ]
user@host# set accounting-options filter-profile fw_profile file fw_acc
```

8. Configure the filter for the interval.

```
[edit ]
user@host# set accounting-options filter-profile fw_profile interval 1
```

9. Configure the counter.

```
[edit ]
user@host# set accounting-options filter-profile fw_profile counters count1
```

10. Apply the filter to the interface.

```
[edit ]
user@host# set interfaces ge-0/0/1 unit 512 family vpls filter input blue
```

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

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

Verification

To verify the configuration is working properly, enter the following commands:

```
show firewall < >
```

```
show accounting records < >
```

Example: Configuring OSPF on the VPLS PE Router

This example shows how to configure OSPF on the VPLS PE router.

- [Requirements on page 106](#)
- [Overview on page 107](#)
- [Configuration on page 107](#)
- [Verification on page 107](#)

Requirements

Before you begin:

- Before you begin, see Understanding Selective Stateless Packet-Based Services in the *Junos OS Security Configuration Guide*.
- Configure the interfaces that will carry the VPLS traffic between the PE router and the CE devices. See “[Example: Configuring Routing Interfaces on the VPLS PE Router](#)” on [page 88](#) and “[Example: Configuring the Interface to the VPLS CE Device](#)” on [page 89](#).
- Create a VPLS routing instance on each PE router that is participating in the VPLS. See “[Example: Configuring the VPLS Routing Instance](#)” on [page 93](#).

Overview

The PE routers exchange routing information using an IGP such as OSPF. In this example, you configure OSPF area 0.0.0.0 on the VPLS PE router and traffic engineering for OSPF.

Configuration

Step-by-Step Procedure

To configure OSPF on the VPLS PE router:

1. Configure the OSPF area on the VPLS PE router.

```
[edit]  
user@host# set protocols ospf area 0.0.0.0 interface t1-1/0/1.0  
user@host# set protocols ospf area 0.0.0.0 interface lo0.0
```
2. Configure traffic engineering for OSPF.

```
[edit]  
user@host# set protocols ospf traffic-engineering
```
3. If you are done configuring the device, commit the configuration.

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

Verification

To verify the configuration is working properly, enter the **show protocols** command.

Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [VPLS Configuration Overview on page 85](#)
- [VPLS Overview on page 81](#)

Example: Configuring RSVP on the VPLS PE Router

This example shows how to configure RSVP on the VPLS PE router.

- [Requirements on page 107](#)
- [Overview on page 108](#)
- [Configuration on page 108](#)
- [Verification on page 108](#)

Requirements

Before you begin:

- Before you begin, see Understanding Selective Stateless Packet-Based Services in the [Junos OS Security Configuration Guide](#).
- Configure the interfaces that will carry the VPLS traffic between the PE router and the CE devices. See “[Example: Configuring Routing Interfaces on the VPLS PE Router](#)” on [page 88](#) and “[Example: Configuring the Interface to the VPLS CE Device](#)” on [page 89](#).

- Create a VPLS routing instance on each PE router that is participating in the VPLS. See [“Example: Configuring the VPLS Routing Instance” on page 93](#).
- Configure an IGP on the PE routers to exchange routing information. See [“Example: Configuring OSPF on the VPLS PE Router” on page 106](#).

Overview

This example describes how to enable RSVP for all connections that participate in the LSP on the PE1 router.

Configuration

Step-by-Step Procedure

To configure RSVP on the VPLS PE router:

1. Configure the interface to the PE2 router for RSVP.

```
[edit ]
user@host# set protocols rsvp interface t1-1/0/1.0
```
2. Configure the loopback interface for RSVP.

```
[edit ]
user@host# set protocols rsvp interface lo0.0
```
3. If you are done configuring the device, commit the configuration.

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

Verification

To verify the configuration is working properly, enter the **show protocols** command.

Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [VPLS Configuration Overview on page 85](#)
- [VPLS Overview on page 81](#)

Example: Configuring MPLS on the VPLS PE Router

This example shows how to configure MPLS on the VPLS PE router.

- [Requirements on page 109](#)
- [Overview on page 109](#)
- [Configuration on page 109](#)
- [Verification on page 110](#)

Requirements

Before you begin:

- Before you begin, see Understanding Selective Stateless Packet-Based Services in the *Junos OS Security Configuration Guide*.
- Configure the interfaces that will carry the VPLS traffic between the PE router and the CE devices. See “Example: Configuring Routing Interfaces on the VPLS PE Router” on page 88 and “Example: Configuring the Interface to the VPLS CE Device” on page 89.
- Create a VPLS routing instance on each PE router that is participating in the VPLS. See “Example: Configuring the VPLS Routing Instance” on page 93.
- Configure an IGP on the PE routers to exchange routing information. See “Example: Configuring OSPF on the VPLS PE Router” on page 106.
- Configure RSVP-TE on the PE routers. See “Example: Configuring RSVP on the VPLS PE Router” on page 107.

Overview

This example shows you how to configure MPLS on the PE1 router to advertise the Layer 2 VPN interface that communicates with the PE2 router.



CAUTION: MPLS is disabled by default on J Series and SRX Series devices. You must explicitly configure your router to allow MPLS traffic. However, when MPLS is enabled, all flow-based security features are deactivated and the router performs packet-based processing. Flow-based services such as security policies, zones, NAT, ALGs, chassis clustering, screens, firewall authentication, and IPsec VPNs are unavailable on the router.

Configuration

Step-by-Step Procedure

To configure MPLS on the VPLS PE router:

1. Configure the interface to the PE2 router for MPLS.

```
[edit ]
user@host# set protocols mpls interface t1-1/0/1.0
```
2. Configure the loopback for MPLS.

```
[edit ]
user@host# set protocols mpls interface lo0.0
```
3. Configure the path to destination 10.255.7.164.

```
[edit ]
user@host# set protocols mpls label-switched-path chelsea-sagar to 10.255.7.164
```
4. If you are done configuring the device, commit the configuration.

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

Verification

To verify the configuration is working properly, enter the **show mpls** command.

Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [VPLS Configuration Overview on page 85](#)
- [VPLS Overview on page 81](#)

Example: Configuring LDP on the VPLS PE Router

This example shows how to configure LDP on the VPLS PE router.

- [Requirements on page 110](#)
- [Overview on page 110](#)
- [Configuration on page 110](#)
- [Verification on page 111](#)

Requirements

Before you begin:

- Before you begin, see Understanding Selective Stateless Packet-Based Services in the *Junos OS Security Configuration Guide*.
- Configure the interfaces that will carry the VPLS traffic between the PE router and the CE devices. See “[Example: Configuring Routing Interfaces on the VPLS PE Router](#)” on page 88 and “[Example: Configuring the Interface to the VPLS CE Device](#)” on page 89.
- Create a VPLS routing instance on each PE router that is participating in the VPLS. See “[Example: Configuring the VPLS Routing Instance](#)” on page 93.
- Configure an IGP on the PE routers to exchange routing information. See “[Example: Configuring OSPF on the VPLS PE Router](#)” on page 106.

Overview

This example describes how to enable LDP for all connections that participate in the LSP on the PE1 router.

Configuration

Step-by-Step Procedure

To configure LDP on the VPLS PE router:

1. Configure the interface to the PE2 router for LDP.

[edit]
user@host# **set protocols ldp interface ge-3/0/2**
2. Configure the loopback interface for LDP.

[edit]
user@host# **set protocols ldp interface lo0**

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

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

Verification

To verify the configuration is working properly, enter the **show protocols** command.

Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [VPLS Configuration Overview on page 85](#)
- [VPLS Overview on page 81](#)

Example: Configuring Routing Options on the VPLS PE Router

This example shows how to configure the routing options on the VPLS PE router.

- [Requirements on page 111](#)
- [Overview on page 111](#)
- [Configuration on page 112](#)
- [Verification on page 112](#)

Requirements

Before you begin:

- Before you begin, see Understanding Selective Stateless Packet-Based Services in the *Junos OS Security Configuration Guide*.
- Configure the interfaces that will carry the VPLS traffic between the PE router and the CE devices. See “[Example: Configuring Routing Interfaces on the VPLS PE Router](#)” on [page 88](#) and “[Example: Configuring the Interface to the VPLS CE Device](#)” on [page 89](#).
- Create a VPLS routing instance on each PE router that is participating in the VPLS. See “[Example: Configuring the VPLS Routing Instance](#)” on [page 93](#).
- Configure an IGP on the PE routers to exchange routing information. See “[Example: Configuring OSPF on the VPLS PE Router](#)” on [page 106](#)
- Configure RSVP-TE, see “[Example: Configuring RSVP on the VPLS PE Router](#)” on [page 107](#) and then MPLS LSPs on the PE routers, see “[Example: Configuring MPLS on the VPLS PE Router](#)” on [page 108](#). Alternatively configure LDP on the PE routers, see “[Example: Configuring LDP on the VPLS PE Router](#)” on [page 110](#).

Overview

This example describes how to specify the router ID and the AS number for each router involved in the VPLS. In this example, the routers PE1 and PE2 use the same AS number (100).

Configuration

Step-by-Step Procedure

To configure the routing options on the VPLS PE router:

1. Configure the router ID on the VPLS PE router.
`[edit]`
`user@host# set routing-options router-id 10.255.7.168`
2. Configure the AS number on the VPLS PE router.
`[edit]`
`user@host# set routing-options autonomous-system 100`
3. If you are done configuring the device, commit the configuration.
`[edit]`
`user@host# commit`

Verification

To verify the configuration is working properly, enter the **show routing-options** command.

Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [VPLS Configuration Overview on page 85](#)
- [VPLS Overview on page 81](#)

Example: Configuring BGP on the VPLS PE Router

This example shows how to configure BGP on the VPLS PE router.

- [Requirements on page 112](#)
- [Overview on page 113](#)
- [Configuration on page 113](#)
- [Verification on page 113](#)

Requirements

Before you begin:

- See Understanding Selective Stateless Packet-Based Services in the [Junos OS Security Configuration Guide](#).
- Configure the interfaces that will carry the VPLS traffic between the PE router and the CE devices. See “[Example: Configuring Routing Interfaces on the VPLS PE Router](#)” on [page 88](#) and “[Example: Configuring the Interface to the VPLS CE Device](#)” on [page 89](#).
- Create a VPLS routing instance on each PE router that is participating in the VPLS. See “[Example: Configuring the VPLS Routing Instance](#)” on [page 93](#).
- Configure an IGP on the PE routers to exchange routing information. See “[Example: Configuring OSPF on the VPLS PE Router](#)” on [page 106](#).

- Configure RSVP-TE. See [“Example: Configuring RSVP on the VPLS PE Router” on page 107](#). Then configure MPLS LSPs on the PE routers. See [“Example: Configuring MPLS on the VPLS PE Router” on page 108](#). Alternatively, configure LDP on the PE routers. See [“Example: Configuring LDP on the VPLS PE Router” on page 110](#).
- Configure routing options on the PE router. See [“Example: Configuring Routing Options on the VPLS PE Router” on page 111](#).

Overview

In this example, you configure an internal BGP session between PE routers so that the routers can exchange information about routes originating and terminating in the VPLS. The PE routers use this information to determine which labels to use for traffic destined for remote sites.

Configuration

Step-by-Step Procedure

To configure BGP on the VPLS PE router:

1. Configure the BGP internal group on the VPLS PE router.

```
[edit ]
user@host# set protocols bgp group ibgp type internal local-address 10.255.7.168
neighbor 10.255.7.164
```
2. Configure the BGP family L2vpn and specify NLRI signaling.

```
[edit ]
user@host# set protocols bgp family L2 VPN signaling
```
3. If you are done configuring the device, commit the configuration.

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

Verification

To verify the configuration is working properly, enter the **show protocols** command.

Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [VPLS Configuration Overview on page 85](#)
- [VPLS Overview on page 81](#)

Example: Configuring VPLS over GRE with IPSec VPNs

This example demonstrates a network scenario consisting of a central office and one branch office that will use VPLS, MPLS, GRE, and IPsec to create secure Ethernet connectivity over a Layer 3 network. This configuration can be expanded to add many other branch sites.

- [Requirements on page 114](#)
- [Overview on page 114](#)

- [Configuration on page 119](#)
- [Verification on page 130](#)

Requirements

Before you begin:

- Ensure that a layer 3 network is in place for all branch offices and that there is a head-end device at the central office configured to terminate the VPNs from each branch office.
- Obtain IDP licenses for each SRX Series device. IDP is used to reassemble GRE packets that might become fragmented.

Overview

Junos OS can selectively choose whether traffic is processed by the flow engine or packet engine using the selective stateless packet-based feature. This feature allows you to combine flow and packet-based services in a single device. In this example, we describe a deployment scenario that uses this feature to deploy large-scale VPLS over GRE. This enables branch devices to securely transport Ethernet traffic over Layer 3 networks when used in conjunction with IPsec.

In this scenario you configure a central office head-end using an SRX650 device and one branch office using an SRX240 device. This setup is accomplished by carrying MPLS pseudowires over GRE, which in turn, is encapsulated in IPsec in order to guarantee data integrity and confidentiality. By default, SRX Series devices and J Series devices use secure flow forwarding. Because VPLS services are provided in packet-mode only, the configuration requires the GRE tunnel to be terminated in a packet-mode routing instance (the default routing instance).



NOTE: You can also use an MX Series device as the head-end device, which is mentioned later in this topic.

To better understand this configuration, we will discuss two scenarios. The first scenario uses pseudowires to allow the creation of point-to-point circuits between two endpoints carried over the MPLS network. If we leave the signaling protocols aside (that is, there are a few ways to provision the pseudowires), these connections are just point-to-point connections. Using this approach provides an end-to-end wire between sites. This is beneficial from a traffic processing point of view because the gateways do not need to do MAC address learning, they simply forward anything they receive to the pseudowire. Because of this, it may be difficult to deploy this setup when trying to provide connectivity to multiple branch offices.

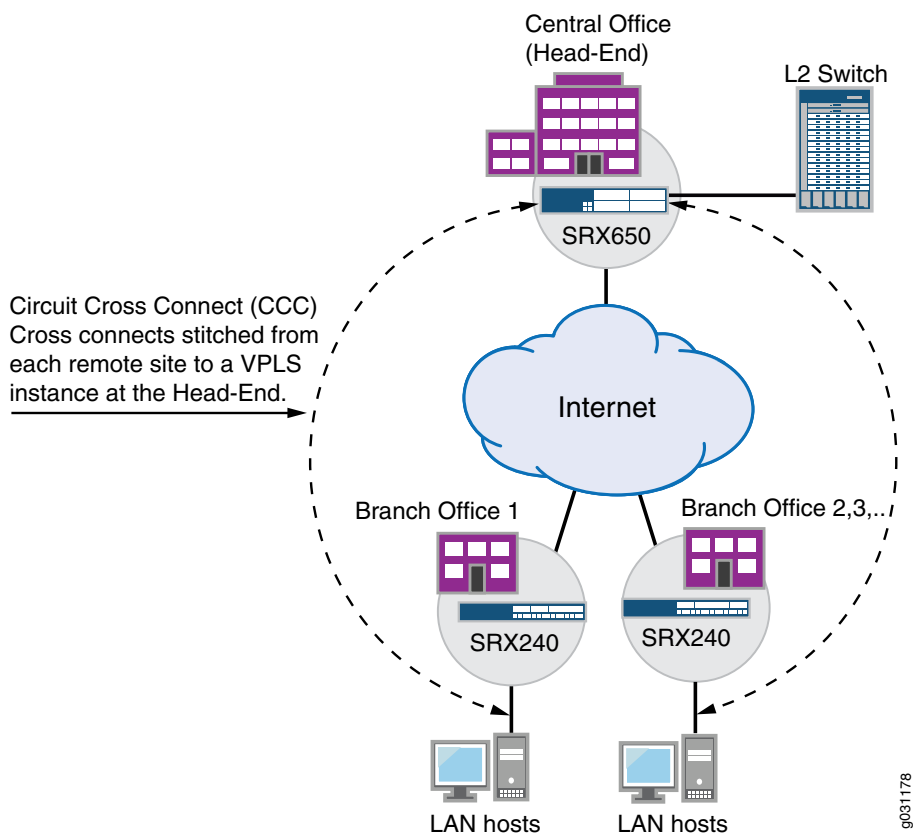
The second scenario could use VPLS to provide a Layer 2 network abstraction. With VPLS, endpoints are expected to negotiate LSPs and pseudowires with every other endpoint (that is, they are fully meshed). When a node receives an Ethernet frame from one of its LAN interfaces the source MAC address is learned, if it's not already known, and flooded using every pseudowire connecting to all other branch nodes. However, if

the destination has been previously learned, then the frame is sent to the appropriate destination. When an Ethernet frame is received through one of the pseudowires (that is, from the MPLS network), source MAC address learning is performed. The next time a frame is sent to that MAC it does not need to be flooded and the frame is flooded to every single LAN interface in the node, but not over the pseudowires. In other words, the network acts as a distributed Layer 2 switch providing any-to-any Ethernet connectivity between the devices connected to the different nodes in the network.

While the advantages of this second scenario is evident (any-to-any connectivity, automated provisioning, and simple abstraction), it comes at the cost of complexity. Every PE node has to perform Layer 2 learning and flooding of traffic, which can cause problems when either multiple broadcast/multicast or frames to unknown MAC addresses are used. As an example, if you had a topology with a thousand branch offices, each office that receives a broadcast packet must replicate it 999 times, encapsulate each copy in GRE and IPsec and forward the resulting traffic. Additionally, because each node performs Layer 2 learning, there are limitations in the maximum number of MAC addresses that each node can learn, limiting the total number of nodes in the domain.

In this example, we use a hybrid approach to these two scenarios. We use a circuit cross connect (CCC) at each branch office stitched to a VPLS instance at central office (ingress). This solution makes sense if most of the traffic flows from the branch offices to central office, and the branch-to-branch office traffic is always forwarded through the hub. The use of CCCs at branch offices combined with VPLS stitching at the central office provides a scalable way to deploy large hub-and-spoke topologies where Ethernet must be transported over an IP network (with or without encryption). At the expense of configuration complexity, it is possible to use branch SRX Series devices to terminate such connections, providing a scalable and cost-effective way to deploy small-to-large networks where Ethernet traffic is carried transparently using lower cost IP connections. [Figure 9 on page 116](#) shows this topology.

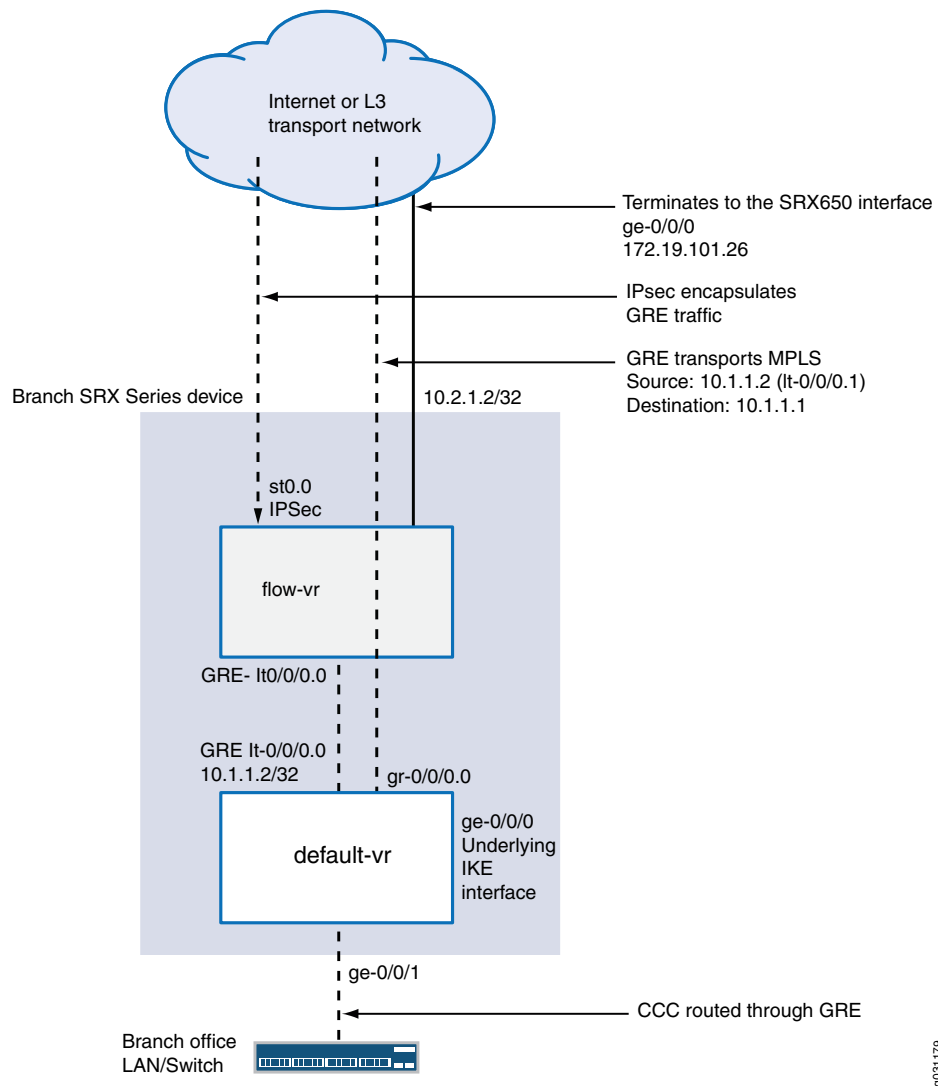
Figure 9: VPLS Deployment Scenario



g031178

In this deployment, VPLS services are provided only in packet mode and must be configured in the default routing instance. Unfortunately, IPsec is only provided in flow mode. Hence, a flow-mode routing-instance is used that provides both GRE reassembly and IPsec termination. While the GRE termination is done in the default routing instance, a flow-mode routing instance is connected between the default routing instance and the Internet (or whatever Layer 3 network is used as a transport), and it terminates the IPsec tunnel towards the ingress device. Because it is likely that a single public IP address is available, the Internet-facing Interface is connected to the default routing instance and is used to terminate IKE; however, the tunnel interface (st0) is bound to the flow-mode routing instance. See [Figure 10 on page 117](#).

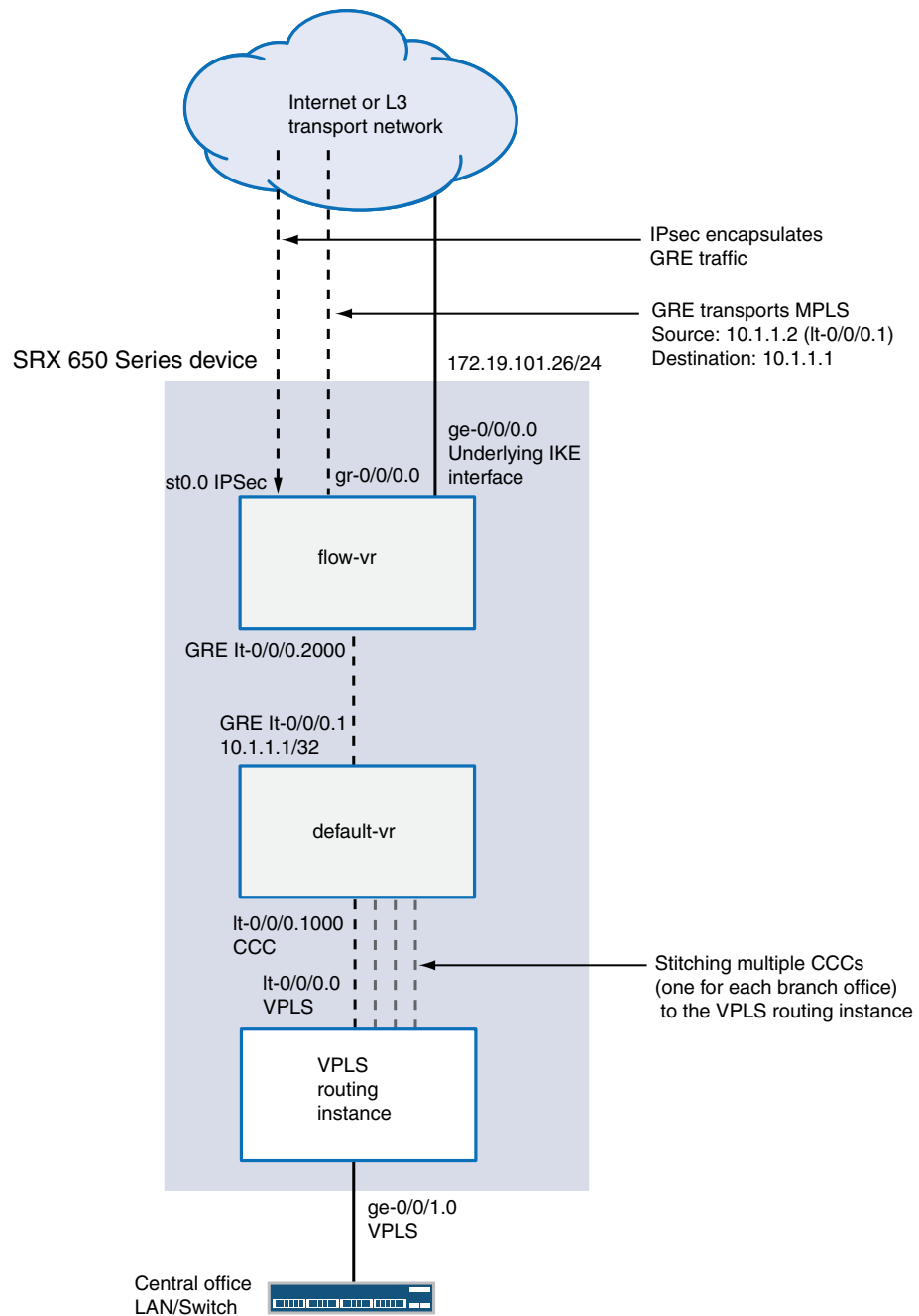
Figure 10: Branch Office Circuit Cross Connect Termination



When configuring the central office SRX650, the first thing you do is terminate the IPsec tunnels, GRE, and CCC connections. Because a branch SRX Series device is used as the head-end, the configuration to terminate the CCC circuits is identical to the one used at each branch office, with the exception that instead of one tunnel, multiple tunnels (and pseudowires) are terminated.

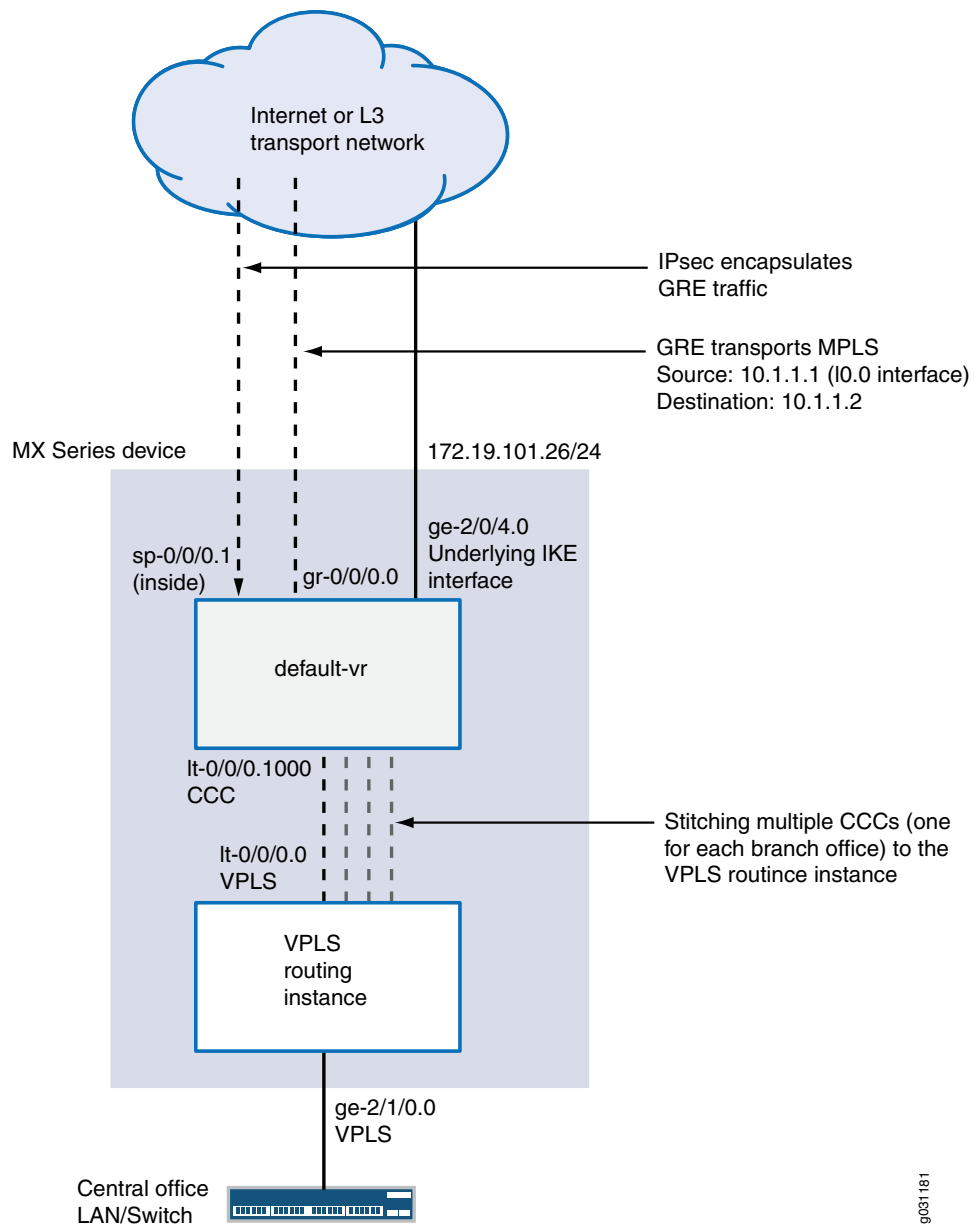
The pseudowires are stitched to a VPLS routing instance using logical tunnel (It) interfaces. It is possible to use an It interface unit to terminate a CCC connection and connect this unit to a different unit that is part of a VPLS routing instance. The overall result is as if the pseudowires were terminated directly in the VPLS routing instance. [Figure 11 on page 118](#) illustrates this configuration.

Figure 11: Central Office Head-End Configuration with an SRX Series Device



You can also use an MX Series device as the central office head-end to terminate all branch office connections. The differences in the configuration are due to the way IPsec is configured and the fact that on MX Series devices IDP is not required to reassemble the GRE packets; MX Series devices natively support GRE reassembly. With this configuration, you still use It interfaces to stitch the CCCs between the remote branch offices and the VPLS routing instance as shown in [Figure 12 on page 119](#).

Figure 12: Central Office Head-End Configuration with an MX Series Device



Configuration

In this example, we use SRX devices and the branch and head-end sites will typically be connected to the Internet by Frame-Relay/T1-E1/xDSL/T3/E3 or even Ethernet. A provider MPLS network is not required.

- [Configuring the SRX240 Device at the Branch Office on page 120](#)
- [Configuring the SRX650 Device at the Central Office on page 124](#)

Configuring the SRX240 Device at the Branch Office

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

```

set interfaces gr-0/0/0 description "GRE tunnel to SRX650"
set interfaces gr-0/0/0 unit 0 clear-dont-fragment-bit
set interfaces gr-0/0/0 unit 0 tunnel source 10.1.1.2
set interfaces gr-0/0/0 unit 0 tunnel destination 10.1.1.1
set interfaces gr-0/0/0 unit 0 tunnel allow-fragmentation
set interfaces gr-0/0/0 unit 0 family inet mtu 2000
set interfaces gr-0/0/0 unit 0 family inet filter input inet-packet-mode
set interfaces gr-0/0/0 unit 0 family mpls mtu 1900
set interfaces gr-0/0/0 unit 0 family mpls filter input mpls-packet-mode
set interfaces lt-0/0/0 unit 0 encapsulation frame-relay
set interfaces lt-0/0/0 unit 0 dlci 16
set interfaces lt-0/0/0 unit 0 peer-unit 1
set interfaces lt-0/0/0 unit 0 family inet
set interfaces lt-0/0/0 unit 0 description "Flow-vr Instance"
set interfaces lt-0/0/0 unit 1 encapsulation frame-relay
set interfaces lt-0/0/0 unit 1 dlci 16
set interfaces lt-0/0/0 unit 1 peer-unit 0
set interfaces lt-0/0/0 unit 1 family inet filter input inet-packet-mode
set interfaces lt-0/0/0 unit 1 family inet address 10.1.1.2/32
set interfaces ge-0/0/1 encapsulation ethernet-ccc
set interfaces ge-0/0/1 unit 0 description "CCC Interface to customer LAN"
set interfaces ge-0/0/1 unit 0 family ccc filter input ccc-packet-mode
set interfaces ge-0/0/0 unit 0 family inet address 172.19.101.45/24
set interfaces lo0 unit 0 family inet address 10.2.1.2/32
set interfaces st0 unit 0 family inet
set routing-options static route 0.0.0.0/0 next-hop 172.19.101.1
set routing-options static route 10.1.1.1/32 next-hop lt-0/0/0.1
set routing-options static route 10.2.1.1/32 next-hop gr-0/0/0.0
set routing-options router-id 10.2.1.2
set protocols mpls interface gr-0/0/0.0
set protocols ldp interface gr-0/0/0.0
set protocols ldp interface lo0.0
set protocols l2circuit neighbor 10.2.1.1 interface ge-0/0/1.0 virtual-circuit-id 1
set security ike policy SRX650 mode main
set security ike policy SRX650 proposal-set standard
set security ike policy SRX650 pre-shared-key ascii-text
"$9$awGjqTz6uORmFORhSMWJGD"
set security ike gateway SRX650 ike-policy SRX650
set security ike gateway SRX650 address 172.19.101.26
set security ike gateway SRX650 external-interface ge-0/0/0.0
set security ipsec policy SRX650 proposal-set standard
set security ipsec vpn SRX650 bind-interface st0.0
set security ipsec vpn SRX650 ike gateway SRX650
set security ipsec vpn SRX650 ike ipsec-policy SRX650
set security ipsec vpn SRX650 establish-tunnels immediately
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces gr-0/0/0.0

```



```

set security zones security-zone untrust interfaces lo0.0
set security zones security-zone untrust interfaces lt-0/0/0.1
set security zones security-zone untrust interfaces ge-0/0/0.0
set security zones security-zone vpn host-inbound-traffic system-services all
set security zones security-zone vpn host-inbound-traffic protocols all
set security zones security-zone vpn interfaces st0.0
set security zones security-zone trust-flow host-inbound-traffic system-services all
set security zones security-zone trust-flow host-inbound-traffic protocols all
set security zones security-zone trust-flow interfaces lt-0/0/0.0
set security policies from-zone trust-flow to-zone vpn policy gre match source-address
  any
set security policies from-zone trust-flow to-zone vpn policy gre match destination-address
  any
set security policies from-zone trust-flow to-zone vpn policy gre match application
  junos-gre
set security policies from-zone trust-flow to-zone vpn policy gre then permit
  application-services idp
set security idp idp-policy gre-reassembly rulebase-ips rule match-all match application
  junos-gre
set security idp idp-policy gre-reassembly rulebase-ips rule match-all then action
  ignore-connection
set security idp active-policy gre-reassembly
set firewall family inet filter inet-packet-mode term control-traffic from protocol tcp
set firewall family inet filter inet-packet-mode term control-traffic from port 22
set firewall family inet filter inet-packet-mode term control-traffic from port 80
set firewall family inet filter inet-packet-mode term control-traffic from port 8080
set firewall family inet filter inet-packet-mode term control-traffic then accept
set firewall family inet filter inet-packet-mode term packet-mode then packet-mode
set firewall family inet filter inet-packet-mode term packet-mode then accept
set firewall family mpls filter mpls-packet-mode term packet-mode then packet-mode
set firewall family mpls filter mpls-packet-mode term packet-mode then accept
set firewall family ccc filter ccc-packet-mode term all then packet-mode
set firewall family ccc filter ccc-packet-mode term all then accept
set routing-instances flow-vr instance-type virtual-router
set routing-instances flow-vr interface lt-0/0/0.0
set routing-instances flow-vr interface st0.0
set routing-instances flow-vr routing-options static route 10.1.1.1/32 next-hop st0.0
set routing-instances flow-vr routing-options static route 10.1.1.2/32 next-hop lt-0/0/0.0

```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the [Junos OS CLI User Guide](#).

To configure the SRX240 at the branch office:

1. Configure a GRE tunnel to the central office.

```

[edit interfaces]
user@host# set gr-0/0/0 description "GRE tunnel to SRX650"
user@host# set gr-0/0/0 unit 0 clear-dont-fragment-bit
user@host# set gr-0/0/0 unit 0 tunnel source 10.1.1.2
user@host# set gr-0/0/0 unit 0 tunnel destination 10.1.1.1
user@host# set gr-0/0/0 unit 0 tunnel allow-fragmentation
user@host# set gr-0/0/0 unit 0 family inet mtu 2000
user@host# set gr-0/0/0 unit 0 family inet filter input inet-packet-mode
user@host# set gr-0/0/0 unit 0 family mpls mtu 1900

```

- ```
user@host# set gr-0/0/0 unit 0 family mpls filter input mpls-packet-mode
```
2. Create a logical interface that connects to the default routing instance.  

```
[edit interfaces]
user@host# set lt-0/0/0 unit 0 encapsulation frame-relay
user@host# set lt-0/0/0 unit 0 dlci 16
user@host# set lt-0/0/0 unit 0 peer-unit 1
user@host# set lt-0/0/0 unit 0 family inet
user@host# set lt-0/0/0 unit 0 description "Flow-vr Instance"
```
  3. Connect the logical tunnel interface to the flow mode virtual router.  

```
[edit interfaces]
user@host# set lt-0/0/0 unit 1 encapsulation frame-relay
user@host# set lt-0/0/0 unit 1 dlci 16
user@host# set lt-0/0/0 unit 1 peer-unit 0
user@host# set lt-0/0/0 unit 1 family inet filter input inet-packet-mode
user@host# set lt-0/0/0 unit 1 family inet address 10.1.1.2/32
```
  4. Connect the CCC interface to the branch LAN.  

```
[edit interfaces]
user@host# set ge-0/0/1 encapsulation ethernet-ccc
user@host# set ge-0/0/1 unit 0 description "CCC Interface to customer LAN"
user@host# set ge-0/0/1 unit 0 family ccc filter input ccc-packet-mode
```
  5. Configure the interface bound to the default virtual router.  

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 family inet address 172.19.101.45/24
```
  6. Set the loopback interface to terminate the CCC connection.  

```
[edit interfaces]
user@host# set lo0 unit 0 family inet address 10.2.1.2/32
```
  7. Bind the IPsec tunnel interface to the flow-mode virtual router.  

```
[edit interfaces]
user@host# set st0 unit 0 family inet
```
  8. Set a static route address, which will be the default gateway to the Internet.  

```
[edit routing-options]
user@host# set static route 0.0.0.0/0 next-hop 172.19.101.1
```
  9. Set a static route for the remote GRE tunnel endpoint.  

```
[edit routing-options]
user@host# set static route 10.1.1.1/32 next-hop lt-0/0/0.1
```
  10. Set a static route for the loopback interface of the SRX650 head-end device.  

```
[edit routing-options]
user@host# set static route 10.2.1.1/32 next-hop gr-0/0/0.0
```
  11. Configure MPLS and the CCC using LDP as the label protocol.  

```
[edit]
user@host# set routing-options router-id 10.2.1.2
user@host# set protocols mpls interface gr-0/0/0.0
user@host# set protocols ldp interface gr-0/0/0.0
```

```

user@host# set protocols ldp interface lo0.0
user@host# set protocols l2circuit neighbor 10.2.1.1 interface ge-0/0/1.0
virtual-circuit-id 1

```

12. Configure the IPsec tunnel.



**NOTE:** The underlying IKE interface is not in the same routing instance as the tunnel interface.

```

[edit security]
user@host# set ike policy SRX650 mode main
user@host# set ike policy SRX650 proposal-set standard
user@host# set ike policy SRX650 pre-shared-key ascii-text
"9awGjqTz6uORmfORhSMWJGD"
user@host# set ike gateway SRX650 ike-policy SRX650
user@host# set ike gateway SRX650 address 172.19.101.26
user@host# set ike gateway SRX650 external-interface ge-0/0/0.0
user@host# set ipsec policy SRX650 proposal-set standard
user@host# set ipsec vpn SRX650 bind-interface st0.0
user@host# set ipsec vpn SRX650 ike gateway SRX650
user@host# set ipsec vpn SRX650 ike ipsec-policy SRX650
user@host# set ipsec vpn SRX650 establish-tunnels immediately

```

13. Configure security zones.



**NOTE:** In a production environment, host-inbound traffic should be restricted to only allow the necessary protocols and services.

```

[edit security]
user@host# set zones security-zone untrust host-inbound-traffic system-services
all
user@host# set zones security-zone untrust host-inbound-traffic protocols all
user@host# set zones security-zone untrust interfaces gr-0/0/0.0
user@host# set zones security-zone untrust interfaces lo0.0
user@host# set zones security-zone untrust interfaces lt-0/0/0.1
user@host# set zones security-zone untrust interfaces ge-0/0/0.0
user@host# set zones security-zone vpn host-inbound-traffic system-services all
user@host# set zones security-zone vpn host-inbound-traffic protocols all
user@host# set zones security-zone vpn interfaces st0.0
user@host# set zones security-zone trust-flow host-inbound-traffic system-services
all
user@host# set zones security-zone trust-flow host-inbound-traffic protocols all
user@host# set zones security-zone trust-flow interfaces lt-0/0/0.0

```

14. Configure IDP.

```

[edit security]
user@host# set policies from-zone trust-flow to-zone vpn policy gre match
source-address any
user@host# set policies from-zone trust-flow to-zone vpn policy gre match
destination-address any

```

```

user@host# set policies from-zone trust-flow to-zone vpn policy gre match
application junos-gre
user@host# set policies from-zone trust-flow to-zone vpn policy gre then permit
application-services idp
user@host# set idp idp-policy gre-reassembly rulebase-ips rule match-all match
application junos-gre
user@host# set idp idp-policy gre-reassembly rulebase-ips rule match-all then
action ignore-connection
user@host# set idp active-policy gre-reassembly

```

15. Configure packet-mode filters.

```

[edit firewall]
user@host# set family inet filter inet-packet-mode term control-traffic from protocol
tcp
user@host# set family inet filter inet-packet-mode term control-traffic from port
22
user@host# set family inet filter inet-packet-mode term control-traffic from port
80
user@host# set family inet filter inet-packet-mode term control-traffic from port
8080
user@host# set family inet filter inet-packet-mode term control-traffic then accept
user@host# set family inet filter inet-packet-mode term packet-mode then
packet-mode
user@host# set family inet filter inet-packet-mode term packet-mode then accept
user@host# set family mpls filter mpls-packet-mode term packet-mode then
packet-mode
user@host# set family mpls filter mpls-packet-mode term packet-mode then accept
user@host# set family ccc filter ccc-packet-mode term all then packet-mode
user@host# set family ccc filter ccc-packet-mode term all then accept

```

16. Configure the flow-mode virtual router.

```

[edit routing-instances]
user@host# set flow-vr instance-type virtual-router
user@host# set flow-vr interface lt-0/0/0.0
user@host# set flow-vr interface st0.0
user@host# set flow-vr routing-options static route 10.1.1/32 next-hop st0.0
user@host# set flow-vr routing-options static route 10.1.2/32 next-hop lt-0/0/0.0

```

**Results** From configuration mode, confirm your configuration by entering the **show** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

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

### Configuring the SRX650 Device at the Central Office

#### CLI Quick Configuration

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

```

set interfaces ge-0/0/0 unit 0 family inet address 172.19.101.26/24
set interfaces gr-0/0/0 unit 0 clear-dont-fragment-bit
set interfaces gr-0/0/0 unit 0 tunnel source 10.1.1.1

```

```
set interfaces gr-0/0/0 unit 0 tunnel destination 10.1.1.2
set interfaces gr-0/0/0 unit 0 tunnel allow-fragmentation
set interfaces gr-0/0/0 unit 0 family inet mtu 1500
set interfaces gr-0/0/0 unit 0 family inet filter input inet-packet-mode
set interfaces gr-0/0/0 unit 0 family mpls filter input mpls-packet-mode
set interfaces lt-0/0/0 unit 0 description "VPLS hub port - Interconnect for CCC to
SRX240"
set interfaces lt-0/0/0 unit 0 encapsulation ethernet-vpls
set interfaces lt-0/0/0 unit 0 peer-unit 1000
set interfaces lt-0/0/0 unit 1000 description "Stitch to VPLS for CCC to SRX240"
set interfaces lt-0/0/0 unit 1000 encapsulation ethernet-ccc
set interfaces lt-0/0/0 unit 1000 peer-unit 0
set interfaces lt-0/0/0 unit 1000 family ccc filter input ccc-packet-mode
set interfaces lt-0/0/0 unit 2000 encapsulation frame-relay
set interfaces lt-0/0/0 unit 2000 dlci 1
set interfaces lt-0/0/0 unit 2000 peer-unit 2001
set interfaces lt-0/0/0 unit 2000 family inet
set interfaces lt-0/0/0 unit 2001 encapsulation frame-relay
set interfaces lt-0/0/0 unit 2001 dlci 1
set interfaces lt-0/0/0 unit 2001 peer-unit 2000
set interfaces lt-0/0/0 unit 2001 family inet filter input inet-packet-mode
set interfaces lt-0/0/0 unit 2001 family inet address 10.1.1.1/32
set interfaces ge-0/0/1 unit 0
set interfaces ge-0/0/1 encapsulation ethernet-vpls
set interfaces lo0 unit 0 family inet address 10.2.1.1/32
set interfaces st0 unit 0 family inet
set routing-options static route 10.1.1.2/32 next-hop lt-0/0/0.2001
set routing-options static route 10.2.1.2/32 next-hop gr-0/0/0.0
set protocols mpls interface gr-0/0/0.0
set protocols ldp interface gr-0/0/0.0
set protocols ldp interface lo0.0
set protocols l2circuit neighbor 10.2.1.2 interface lt-0/0/0.1000 virtual-circuit-id 1
set security ike policy SRX mode main
set security ike policy SRX proposal-set standard
set security ike policy SRX pre-shared-key ascii-text "9yhxeMXVwgUjq7-jqmfn6rev"
set security ike gateway SRX240-1 ike-policy SRX
set security ike gateway SRX240-1 address 172.19.101.45
set security ike gateway SRX240-1 external-interface ge-0/0/0.0
set security ipsec policy SRX proposal-set standard
set security ipsec vpn SRX240-1 bind-interface st0.0
set security ipsec vpn SRX240-1 ike gateway SRX240-1
set security ipsec vpn SRX240-1 ike ipsec-policy SRX
set security ipsec vpn SRX240-1 establish-tunnels immediately
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces lo0.0
set security zones security-zone untrust interfaces lt-0/0/0.2001
set security zones security-zone untrust interfaces gr-0/0/0.0
set security zones security-zone untrust interfaces ge-0/0/0.0
set security zones security-zone vpn host-inbound-traffic system-services all
set security zones security-zone vpn host-inbound-traffic protocols all
set security zones security-zone vpn interfaces st0.0
set security zones security-zone trust-flow host-inbound-traffic system-services all
set security zones security-zone trust-flow host-inbound-traffic protocols all
set security zones security-zone trust-flow interfaces lt-0/0/0.2000
```

```

set security policies from-zone trust-flow to-zone vpn policy gre match source-address
any
set security policies from-zone trust-flow to-zone vpn policy gre match destination-address
any
set security policies from-zone trust-flow to-zone vpn policy gre match application
junos-gre
set security policies from-zone trust-flow to-zone vpn policy gre then permit
application-services idp
set security policies from-zone vpn to-zone trust-flow policy gre match source-address
any
set security policies from-zone vpn to-zone trust-flow policy gre match destination-address
any
set security policies from-zone vpn to-zone trust-flow policy gre match application
junos-gre
set security policies from-zone vpn to-zone trust-flow policy gre then permit
application-services idp
set security idp idp-policy gre-reassembly rulebase-ips rule match-gre match application
junos-gre
set security idp idp-policy gre-reassembly rulebase-ips rule match-gre then action
ignore-connection
set security idp active-policy gre-reassembly
set firewall family inet filter inet-packet-mode term control-traffic from protocol tcp
set firewall family inet filter inet-packet-mode term control-traffic from port 22
set firewall family inet filter inet-packet-mode term control-traffic from port 80
set firewall family inet filter inet-packet-mode term control-traffic from port 8080
set firewall family inet filter inet-packet-mode term control-traffic then accept
set firewall family inet filter inet-packet-mode term packet-mode then packet-mode
set firewall family inet filter inet-packet-mode term packet-mode then accept
set firewall family mpls filter mpls-packet-mode term packet-mode then packet-mode
set firewall family mpls filter mpls-packet-mode term packet-mode then accept
set firewall family ccc filter ccc-packet-mode term all then packet-mode
set firewall family ccc filter ccc-packet-mode term all then accept
set routing-instances flow-vr instance-type virtual-router
set routing-instances flow-vr interface lt-0/0/0.2000
set routing-instances flow-vr interface st0.0
set routing-instances flow-vr routing-options static route 10.1.1.1/32 next-hop
lt-0/0/0.2000
set routing-instances flow-vr routing-options static route 10.1.1.2/32 next-hop st0.0
set routing-instances vpls-hub instance-type vpls
set routing-instances vpls-hub interface lt-0/0/0.0
set routing-instances vpls-hub interface ge-0/0/1.0

```

#### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the [Junos OS CLI User Guide](#).

To configure the head-end SRX650 device at the central office:

1. Configure the interface bound to the default virtual router.  

```

[edit interfaces]
user@host# set ge-0/0/0 unit 0 family inet address 172.19.101.26/24

```
2. Create the GRE tunnel from the SRX650 to the SRX240 device.



**NOTE:** As the network expands to include multiple branch offices, you will need to add a similar GRE tunnel configuration on the SRX650 device (head-end) along with a corresponding IPsec configuration to connect to each additional branch device (SRX240).

[edit interfaces]

```
user@host# set gr-0/0/0 unit 0 clear-dont-fragment-bit
user@host# set gr-0/0/0 unit 0 tunnel source 10.1.1.1
user@host# set gr-0/0/0 unit 0 tunnel destination 10.1.1.2
user@host# set gr-0/0/0 unit 0 tunnel allow-fragmentation
user@host# set gr-0/0/0 unit 0 family inet mtu 1500
user@host# set gr-0/0/0 unit 0 family inet filter input inet-packet-mode
user@host# set gr-0/0/0 unit 0 family mpls filter input mpls-packet-mode
```

3. Configure a logical tunnel interface to stitch the CCC connection to the VPLS instance.

[edit interfaces]

```
user@host# set lt-0/0/0 unit 0 description "VPLS hub port - Interconnect for CCC
to SRX240"
user@host# set lt-0/0/0 unit 0 encapsulation ethernet-vpls
user@host# set lt-0/0/0 unit 0 peer-unit 1000
```

4. Set unit 1000 to terminate the CCC connection.

[edit interfaces]

```
user@host# set lt-0/0/0 unit 1000 description "Stitch to VPLS for CCC to SRX240"
user@host# set lt-0/0/0 unit 1000 encapsulation ethernet-ccc
user@host# set lt-0/0/0 unit 1000 peer-unit 0
user@host# set lt-0/0/0 unit 1000 family ccc filter input ccc-packet-mode
```

5. Configure the logical tunnel interface.

[edit interfaces]

```
user@host# set lt-0/0/0 unit 2000 encapsulation frame-relay
user@host# set lt-0/0/0 unit 2000 dlci 1
user@host# set lt-0/0/0 unit 2000 peer-unit 2001
user@host# set lt-0/0/0 unit 2000 family inet
```

6. Bind the logical tunnel interface to the default virtual router.

[edit interfaces]

```
user@host# set lt-0/0/0 unit 2001 encapsulation frame-relay
user@host# set lt-0/0/0 unit 2001 dlci 1
user@host# set lt-0/0/0 unit 2001 peer-unit 2000
user@host# set lt-0/0/0 unit 2001 family inet filter input inet-packet-mode
user@host# set lt-0/0/0 unit 2001 family inet address 10.1.1.1/32
```

7. Set the interface to the central office LAN network.

[edit interfaces]

```
user@host# set ge-0/0/1 unit 0
user@host# set ge-0/0/1 encapsulation ethernet-vpls
```

8. Set the loopback interface to terminate the CCC connections to each branch device.

[edit interfaces]

- ```

user@host# set lo0 unit 0 family inet address 10.2.1.1/32

```
9. Bind the IPsec interface to the flow-mode virtual router.


```

[edit interfaces]
user@host# set st0 unit 0 family inet

```
 10. Set a static route for the remote GRE tunnel endpoint.


```

[edit routing-options]
user@host# set static route 10.1.1.2/32 next-hop lt-0/0/0.2001

```
 11. Set a static route for the loopback interface of the branch device.


```

[edit]
user@host# set routing-options static route 10.2.1.2/32 next-hop gr-0/0/0.0

```
 12. Configure MPLS and CCC using LDP as the label protocol.


```

[edit protocols]
user@host# set mpls interface gr-0/0/0.0
user@host# set ldp interface gr-0/0/0.0
user@host# set ldp interface lo0.0
user@host# set l2circuit neighbor 10.2.1.2 interface lt-0/0/0.1000 virtual-circuit-id
1

```
 13. Configure the IPsec tunnel.



NOTE: The underlying IKE interface is not in the same routing instance as the tunnel interface.

- ```

[edit security]
user@host# set ike policy SRX mode main
user@host# set ike policy SRX proposal-set standard
user@host# set ike policy SRX pre-shared-key ascii-text
"9yhxeMXVwgUjq7-jqmfn6rev"
user@host# set ike gateway SRX240-1 ike-policy SRX
user@host# set ike gateway SRX240-1 address 172.19.101.45
user@host# set ike gateway SRX240-1 external-interface ge-0/0/0.0
user@host# set ipsec policy SRX proposal-set standard
user@host# set ipsec vpn SRX240-1 bind-interface st0.0
user@host# set ipsec vpn SRX240-1 ike gateway SRX240-1
user@host# set ipsec vpn SRX240-1 ike ipsec-policy SRX
user@host# set ipsec vpn SRX240-1 establish-tunnels immediately

```
14. Configure security zones.



**NOTE:** In a production environment, restrict host-inbound traffic to only the necessary protocols and services.

```

[edit security]
user@host# set zones security-zone untrust host-inbound-traffic system-services
all
user@host# set zones security-zone untrust host-inbound-traffic protocols all

```



```

user@host# set zones security-zone untrust interfaces lo0.0
user@host# set zones security-zone untrust interfaces lt-0/0/0.2001
user@host# set zones security-zone untrust interfaces gr-0/0/0.0
user@host# set zones security-zone untrust interfaces ge-0/0/0.0
user@host# set zones security-zone vpn host-inbound-traffic system-services all
user@host# set zones security-zone vpn host-inbound-traffic protocols all
user@host# set zones security-zone vpn interfaces st0.0
user@host# set zones security-zone trust-flow host-inbound-traffic system-services
 all
user@host# set zones security-zone trust-flow host-inbound-traffic protocols all
user@host# set zones security-zone trust-flow interfaces lt-0/0/0.2000

```

15. Configure IDP.

```

[edit security]
user@host# set policies from-zone trust-flow to-zone vpn policy GRE match
 source-address any
user@host# set policies from-zone trust-flow to-zone vpn policy GRE match
 destination-address any
user@host# set policies from-zone trust-flow to-zone vpn policy GRE match
 application junos-gre
user@host# set policies from-zone trust-flow to-zone vpn policy GRE then permit
 application-services idp
user@host# set policies from-zone vpn to-zone trust-flow policy GRE match
 source-address any
user@host# set policies from-zone vpn to-zone trust-flow policy GRE match
 destination-address any
user@host# set policies from-zone vpn to-zone trust-flow policy GRE match
 application junos-gre
user@host# set policies from-zone vpn to-zone trust-flow policy GRE then permit
 application-services idp
user@host# set idp idp-policy gre-reassembly rulebase-ips rule match-gre match
 application junos-gre
user@host# set idp idp-policy gre-reassembly rulebase-ips rule match-gre then
 action ignore-connection
user@host# set idp active-policy gre-reassembly

```

16. Configure packet-mode filters.

```

[edit firewall]
user@host# set family inet filter inet-packet-mode term control-traffic from protocol
 tcp
user@host# set family inet filter inet-packet-mode term control-traffic from port
 22
user@host# set family inet filter inet-packet-mode term control-traffic from port
 80
user@host# set family inet filter inet-packet-mode term control-traffic from port
 8080
user@host# set family inet filter inet-packet-mode term control-traffic then accept
user@host# set family inet filter inet-packet-mode term packet-mode then
 packet-mode
user@host# set family inet filter inet-packet-mode term packet-mode then accept
user@host# set family mpls filter mpls-packet-mode term packet-mode then
 packet-mode
user@host# set family mpls filter mpls-packet-mode term packet-mode then accept
user@host# set family ccc filter ccc-packet-mode term all then packet-mode
user@host# set family ccc filter ccc-packet-mode term all then accept

```

17. Configure the flow-mode virtual router.

```
[edit routing-instances]
user@host# set flow-vr instance-type virtual-router
user@host# set flow-vr interface lt-0/0/0.2000
user@host# set flow-vr interface st0.0
user@host# set flow-vr routing-options static route 10.1.1.1/32 next-hop
lt-0/0/0.2000
user@host# set flow-vr routing-options static route 10.1.1.2/32 next-hop st0.0
```

18. Configure the VPLS instance.

```
[edit routing-instances]
user@host# set vpls-hub instance-type vpls
user@host# set vpls-hub interface lt-0/0/0.0
user@host# set vpls-hub interface ge-0/0/1.0
```

**Results** From configuration mode, confirm your configuration by entering the **show** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

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

## Verification

Confirm that the configuration is working properly.

- [Verifying Interfaces on page 130](#)
- [Verifying an IPsec tunnel on page 130](#)
- [Verifying GRE on page 130](#)
- [Verifying the CCC/L2 circuit. on page 131](#)
- [Verifying that LDP sessions are working. on page 131](#)

### Verifying Interfaces

---

**Purpose** Verify that the interfaces are configured properly on each device in the VPLS network.

**Action** From configuration mode, enter **show interfaces** and verify that the IP addressing is correct for each interface, including logical tunnel (lt), loopback (lo), GRE (gr), IPsec tunnel st0, and GE interfaces.

### Verifying an IPsec tunnel

---

**Purpose** Verify that an IPsec tunnel is working.

**Action** From operational mode, enter the **show security ipsec security associations** and the **show security ipsec statistics** command.

### Verifying GRE

---

**Purpose** Verify that GRE is working.

**Action** From operational mode, enter the **show security flow session protocol gre** command. You can also do a ping between loopback addresses.

#### Verifying the CCC/L2 circuit.

---

**Purpose** Verify that the CCC/L2 circuit is working.

**Action** From operational mode, enter the **show connections** command.

#### Verifying that LDP sessions are working.

---

**Purpose** Verify that LDP sessions are being created between devices.

**Action** From operational mode, enter the **show interfaces gr-0/0/0 detail** command.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [VPLS Overview on page 81](#)
- [Understanding VPLS Interfaces on page 86](#)
- Understanding Selective Stateless Packet-Based Services in the [Junos OS Security Configuration Guide](#)
- [MPLS Overview on page 3](#)



## PART 2

# Index

- [Index on page 135](#)



# Index

## Symbols

|                                              |      |
|----------------------------------------------|------|
| #, comments in configuration statements..... | xiii |
| ( ), in syntax descriptions.....             | xiii |
| < >, in syntax descriptions.....             | xiii |
| [ ], in configuration statements.....        | xiii |
| { }, in configuration statements.....        | xiii |
| (pipe), in syntax descriptions.....          | xiii |

## A

|                          |    |
|--------------------------|----|
| ASs (autonomous systems) |    |
| AS number, in VPNs.....  | 55 |
| LSPs through.....        | 4  |

## B

|                                          |         |
|------------------------------------------|---------|
| BGP                                      |         |
| CLNS.....                                | 77      |
| BGP (Border Gateway Protocol)            |         |
| export policy for CLNS.....              | 72      |
| for CLNS VPN NLRI.....                   | 77      |
| VPLS.....                                | 91, 112 |
| VPNs.....                                | 53      |
| braces, in configuration statements..... | xiii    |
| brackets                                 |         |
| angle, in syntax descriptions.....       | xiii    |
| square, in configuration statements..... | xiii    |

## C

|                                            |    |
|--------------------------------------------|----|
| CE (customer edge) routers.....            | 84 |
| description.....                           | 47 |
| See also VPLS                              |    |
| circuit See Layer 2 circuits               |    |
| CLNS                                       |    |
| BGP.....                                   | 77 |
| CLNS (Connectionless Network Service) VPNs |    |
| BGP export policy.....                     | 72 |
| BGP, to carry CLNS VPN NLRI.....           | 77 |
| displaying configurations.....             | 78 |
| ES-IS.....                                 | 70 |
| IS-IS.....                                 | 72 |
| linking hosts.....                         | 65 |
| overview.....                              | 65 |

|                                                |      |
|------------------------------------------------|------|
| requirements.....                              | 66   |
| static routes (without IS-IS).....             | 74   |
| verifying configuration.....                   | 78   |
| VPN routing instance.....                      | 67   |
| comments, in configuration statements.....     | xiii |
| Connectionless Network Service See CLNS        |      |
| conventions                                    |      |
| notice icons.....                              | xii  |
| text and syntax.....                           | xii  |
| CSPF algorithm See CSPF                        |      |
| curly braces, in configuration statements..... | xiii |
| customer edge routers See CE routers           |      |
| customer support.....                          | xiv  |
| contacting JTAC.....                           | xiv  |

## D

|                                         |     |
|-----------------------------------------|-----|
| diagnosis                               |     |
| displaying CLNS VPN configurations..... | 78  |
| RSVP neighbors.....                     | 24  |
| RSVP sessions.....                      | 25  |
| RSVP-signaled LSP.....                  | 25  |
| documentation                           |     |
| comments on.....                        | xiv |
| dynamic LSPs.....                       | 7   |

## E

|                                                |    |
|------------------------------------------------|----|
| egress router See LSPs; outbound router        |    |
| Enabling MPLS.....                             | 10 |
| End System-to-Intermediate System See ES-IS    |    |
| ES-IS (End System-to-Intermediate System)..... | 69 |
| for a PE router in a CLNS island.....          | 70 |
| overview.....                                  | 65 |
| export routing policy, for Layer 2 VPNs.....   | 56 |

## F

|                       |     |
|-----------------------|-----|
| font conventions..... | xii |
|-----------------------|-----|

## H

|                          |     |
|--------------------------|-----|
| hardware                 |     |
| supported platforms..... | xii |

## I

|                                              |    |
|----------------------------------------------|----|
| IGPs (interior gateway protocols).....       | 47 |
| VPNs.....                                    | 47 |
| See also OSPF                                |    |
| import routing policy, for Layer 2 VPNs..... | 56 |
| inbound router, in an LSP.....               | 5  |
| ingress router See inbound router; LSPs      |    |

|                                                    |        |
|----------------------------------------------------|--------|
| interfaces                                         |        |
| VPLS.....                                          | 86, 89 |
| VPLS encapsulation types.....                      | 87     |
| Intermediate System-to-Intermediate System See     |        |
| IS-IS                                              |        |
| IS-IS (Intermediate System-to-Intermediate System) |        |
| for CLNS route exchange.....                       | 72     |
| with CLNS.....                                     | 65     |
| iso-vpn statement                                  |        |
| usage guidelines.....                              | 77     |

## L

|                                      |        |
|--------------------------------------|--------|
| Label Distribution Protocol See LDP  |        |
| label switching.....                 | 4      |
| label-switched paths See LSPs        |        |
| label-switching routers (LSRs).....  | 5      |
| labels, MPLS.....                    | 6      |
| label operations.....                | 6      |
| PHP.....                             | 7      |
| Layer 2 circuits                     |        |
| AS number.....                       | 55     |
| basic, description.....              | 60     |
| IGPs.....                            | 47     |
| MPLS.....                            | 52, 53 |
| neighbor address.....                | 62     |
| signaling protocols.....             | 47     |
| verifying PE router connections..... | 62     |
| verifying PE router interfaces.....  | 62     |
| virtual circuit ID.....              | 62     |
| Layer 2 VPNs                         |        |
| AS number.....                       | 55     |
| basic, description.....              | 50     |
| BGP.....                             | 53     |
| export routing policies.....         | 56     |
| IGPs.....                            | 47     |
| import routing policies.....         | 56     |
| MPLS.....                            | 52, 53 |
| routing instance.....                | 55     |
| signaling protocols.....             | 47     |
| verifying PE router connections..... | 57     |
| verifying PE router interfaces.....  | 57     |
| Layer 3 VPNs                         |        |
| AS number.....                       | 55     |
| basic, description.....              | 58     |
| BGP.....                             | 53     |
| IGPs.....                            | 47     |
| route target.....                    | 49     |
| routing instance.....                | 55     |

|                                      |        |
|--------------------------------------|--------|
| routing policies.....                | 60     |
| signaling protocols.....             | 47     |
| verifying PE router connections..... | 60     |
| LDP (Label Distribution Protocol)    |        |
| and OSPF for VPNs.....               | 54     |
| overview.....                        | 14     |
| requirements.....                    | 15     |
| VPLS.....                            | 110    |
| LSPs (label-switched paths)          |        |
| description.....                     | 4      |
| dynamic LSPs.....                    | 7      |
| for RSVP in a VPN.....               | 52, 53 |
| label operations.....                | 6      |
| label switching.....                 | 4      |
| labels.....                          | 6      |
| LSR types.....                       | 5      |
| PHP.....                             | 7      |
| static LSPs.....                     | 7      |
| LSRs (label-switching routers).....  | 5      |

## M

|                                                  |        |
|--------------------------------------------------|--------|
| manuals                                          |        |
| comments on.....                                 | xiv    |
| metric statement                                 |        |
| CLNS                                             |        |
| usage guidelines.....                            | 74     |
| MPLS (Multiprotocol Label Switching).....        | 47     |
| dynamic LSPs.....                                | 7      |
| enabling and disabling.....                      | 10     |
| label operations.....                            | 6      |
| label switching.....                             | 4      |
| labels.....                                      | 6      |
| Layer 2 VPNs and Layer 2 circuits.....           | 52, 53 |
| LSP for RSVP in a VPN.....                       | 52, 53 |
| LSPs.....                                        | 4      |
| LSR types.....                                   | 5      |
| PHP.....                                         | 7      |
| static LSPs.....                                 | 7      |
| traffic engineering See MPLS traffic engineering |        |
| VPLS.....                                        | 108    |
| See also VPNs                                    |        |
| MPLS traffic engineering                         |        |
| LDP signaling.....                               | 14     |
| overview.....                                    | 13     |
| requirements.....                                | 15     |
| RSVP signaling.....                              | 19     |
| verifying RSVP neighbors.....                    | 24     |
| verifying RSVP sessions.....                     | 25     |
| verifying RSVP-signaled LSPs.....                | 25     |



multiple push label operation.....6

## N

network layer reachability information See NLRI

network service access points See NSAPs

networks

sample LSP topology.....5

sample RSVP topology.....20

next-hop statement

CLNS

usage guidelines.....74

NLRI (network layer reachability information), BGP

for CLNS.....77

for VPNs.....49

notice icons.....xii

NSAPs (network service access points)

overview.....65

sample configurations.....74

## O

Open Systems Interconnection (OSI) networks,

CLNS VPNs.....65

OSI (Open Systems Interconnection) networks,

CLNS VPNs.....65

OSPF (Open Shortest Path First)

and LDP for VPNs.....54

and RSVP for VPNs.....55

VPLS.....106

outbound router, in an LSP.....5

## P

p2mp-lsp-next-hop statement

RSVP

usage guidelines.....27

parentheses, in syntax descriptions.....xiii

PE (provider edge) routers.....82

description.....47

ES-IS for a CLNS island.....70

route distinguishers.....49

verifying Layer 2 circuit connections.....62

verifying Layer 2 circuit interfaces.....62

verifying Layer 2 VPN connections.....57

verifying Layer 2 VPN interfaces.....57

verifying Layer 3 VPN connections.....60

See also VPLS

penultimate hop popping (PHP).....7

penultimate router, in an LSP.....5

PHP (penultimate hop popping).....7

ping mpls l2circuit interface command.....62

ping mpls l2circuit virtual-circuit command.....62

ping mpls l2vpn instance.....57

ping mpls l3vpn command.....60

point-to-multipoint LSPs

with RSVP signaling.....27

pop label operation.....6

preference statement

CLNS static routes

usage guidelines.....74

protocols

LDP See LDP

RSVP See RSVP

provider edge routers See PE routers

provider routers

description.....47

push label operation.....6

## Q

qualified-next-hop statement

CLNS

usage guidelines.....74

## R

reachability.....49

See also NLRI

Resource Reservation Protocol See RSVP

route distinguishers

description.....49

formats for.....49

route targets, VPN

in a routing instance.....49

routing

configuring VPNs.....47

MPLS traffic engineering.....13

VPNs.....47

routing instance

for CLNS static routes (with IS-IS).....67

for CLNS static routes (without IS-IS).....74

VPLS.....85, 90, 93

VPN configuration.....55

VPN route target.....49

VRF instances.....48

VRF table.....49

routing policies

BGP export, for CLNS.....72

Layer 2 VPN export policy.....56

Layer 2 VPN import policy.....56

Layer 3 VPNs.....60

|                                          |     |
|------------------------------------------|-----|
| routing solutions                        |     |
| MPLS traffic engineering.....            | 13  |
| VPNs.....                                | 47  |
| routing table                            |     |
| verifying RSVP-signaled LSPs.....        | 25  |
| VPLS.....                                | 91  |
| RSVP                                     |     |
| for point-to-multipoint LSPs.....        | 27  |
| RSVP (Resource Reservation Protocol)     |     |
| and OSPF for VPNs.....                   | 55  |
| overview.....                            | 19  |
| requirements.....                        | 15  |
| verifying LSPs.....                      | 25  |
| verifying neighbors.....                 | 24  |
| verifying sessions.....                  | 25  |
| verifying the routing table on the entry |     |
| router.....                              | 25  |
| VPLS.....                                | 107 |
| RSVP neighbors, verifying.....           | 24  |

## S

|                                          |        |
|------------------------------------------|--------|
| sample configurations                    |        |
| CLNS VPN configuration.....              | 78     |
| sessions                                 |        |
| RSVP, verifying.....                     | 25     |
| show ldp neighbor command.....           | 17     |
| show ldp session detail command.....     | 17     |
| show route table inet.3 command.....     | 18, 25 |
| show rsvp neighbor command.....          | 24     |
| show rsvp session detail command.....    | 25     |
| signaling protocols                      |        |
| VPNs.....                                | 47     |
| site identifier, VPLS.....               | 91     |
| site name, VPLS.....                     | 91     |
| site preference, VPLS.....               | 91     |
| site range, VPLS.....                    | 91     |
| static LSPs.....                         | 7      |
| static routes                            |        |
| CLNS VPNs (with IS-IS).....              | 67     |
| CLNS VPNs (without IS-IS).....           | 74     |
| support, technical See technical support |        |
| swap and push label operation.....       | 6      |
| swap label operation.....                | 6      |
| syntax conventions.....                  | xii    |

## T

|                      |     |
|----------------------|-----|
| technical support    |     |
| contacting JTAC..... | xiv |

|                                                   |    |
|---------------------------------------------------|----|
| topology                                          |    |
| point-to-multipoint LSPs.....                     | 26 |
| sample LSP network.....                           | 5  |
| sample RSVP-signaled LSP.....                     | 20 |
| trace options                                     |    |
| VPLS.....                                         | 92 |
| traceroute source bypass-routing gateway          |    |
| command.....                                      | 18 |
| traffic engineering See MPLS traffic engineering; |    |
| traffic engineering database                      |    |
| traffic-engineering statement                     |    |
| usage guidelines.....                             | 27 |
| transit routers, in an LSP.....                   | 5  |

## V

|                                               |             |
|-----------------------------------------------|-------------|
| verification                                  |             |
| CLNS static routes.....                       | 76          |
| CLNS VPNs.....                                | 78          |
| network interfaces.....                       | 44          |
| RSVP neighbors.....                           | 24          |
| RSVP sessions.....                            | 25          |
| RSVP-signaled LSP.....                        | 25          |
| virtual circuit ID, for Layer 2 circuits..... | 62          |
| virtual private LAN service See VPLS          |             |
| virtual private networks See VPNs             |             |
| VLAN CCC                                      |             |
| example configuration.....                    | 96          |
| VLAN encapsulation                            |             |
| configuring.....                              | 95          |
| VLAN VPLS                                     |             |
| example configuration.....                    | 96, 97, 100 |
| VLANs (virtual LANs)                          |             |
| rewrite.....                                  | 88          |
| tagging.....                                  | 87          |
| VPLS (virtual private LAN service).....       | 81          |
| BGP.....                                      | 91, 112     |
| CE device.....                                | 84          |
| configuration overview.....                   | 85          |
| exceptions on J Series Services Routers.....  | 84          |
| filtering.....                                | 104         |
| filtering and accounting.....                 | 104         |
| functions.....                                | 82          |
| interface encapsulation.....                  | 87          |
| interfaces.....                               | 86, 89      |
| LDP.....                                      | 110         |
| MPLS.....                                     | 108         |
| OSPF.....                                     | 106         |
| overview.....                                 | 81          |
| policers.....                                 | 102         |

|                                                                |            |
|----------------------------------------------------------------|------------|
| policing.....                                                  | 102        |
| routing instance.....                                          | 85, 90, 93 |
| routing interfaces.....                                        | 88         |
| routing options.....                                           | 111        |
| routing table.....                                             | 91         |
| RSVP.....                                                      | 107        |
| sample topology.....                                           | 82         |
| site identifier.....                                           | 91         |
| site name.....                                                 | 91         |
| site preference.....                                           | 91         |
| site range.....                                                | 91         |
| supported devices and interfaces.....                          | 81         |
| trace options.....                                             | 92         |
| VLAN rewrite on interfaces.....                                | 88         |
| VLAN tagging.....                                              | 87         |
| VPLS configuration over GRE and IPsec.....                     | 113        |
| VPN routing and forwarding (VRF) instances.....                | 48         |
| VPN routing and forwarding table See VRF table                 |            |
| VPNs (virtual private networks).....                           | 47         |
| AS number.....                                                 | 55         |
| basic Layer 2 circuit description.....                         | 60         |
| basic Layer 2 VPN description.....                             | 50         |
| basic Layer 3 VPN description.....                             | 58         |
| BGP.....                                                       | 53         |
| CLNS See CLNS                                                  |            |
| components.....                                                | 47         |
| IGPs.....                                                      | 47         |
| Layer 2 circuit configuration.....                             | 62         |
| LSP for RSVP.....                                              | 52, 53     |
| MPLS.....                                                      | 52, 53     |
| overview.....                                                  | 47         |
| route distinguishers.....                                      | 49         |
| route target.....                                              | 49         |
| routing information.....                                       | 48         |
| routing instance See routing instance                          |            |
| routing requirements.....                                      | 47         |
| signaling protocols.....                                       | 47         |
| tunneling process.....                                         | 47         |
| VRF instances.....                                             | 48         |
| VRF table See VRF table                                        |            |
| See also Layer 2 circuits; Layer 2 VPNs; Layer 3<br>VPNs; MPLS |            |
| VRF (VPN routing and forwarding) table.....                    | 49         |
| VRF instances.....                                             | 48         |
| VRF instances                                                  |            |
| overview.....                                                  | 48         |

