JUNIPER NETWORKS

Junos® OS

Virtual Private LAN Service Feature Guide

Release

10.4

Published: 2010-11-05
Revision 1

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

Junos® OS Virtual Private LAN Service Feature Guide
Release 10.4

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

Revision History
October 2010—R1 Junos 10.4

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

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

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

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

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

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

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

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

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

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

4. **Use Prohibitions.** Notwithstanding the foregoing, the license provided herein does not permit the Customer to, and Customer agrees not to and shall not: (a) modify, unbundle, reverse engineer, or create derivative works based on the Software; (b) make unauthorized copies of the Software (except as necessary for backup purposes); (c) rent, sell, transfer, or grant any rights in and to any copy of the Software, in any form, to any third party; (d) remove any proprietary notices, labels, or marks on or in any copy of the Software or any product in which the Software is embedded; (e) distribute any copy of the Software to any third party, including as may be embedded in Juniper equipment sold in the secondhand market; (f) use any 'locked' or key-restricted feature, function, service, application, operation, or capability without first purchasing the applicable license(s) and obtaining a valid key from Juniper, even if such feature, function, service, application, operation, or capability is enabled without a key; (g) distribute any key for the Software provided by Juniper to any third party; (h) use the

Software in any manner that extends or is broader than the uses purchased by Customer from Juniper or an authorized Juniper reseller; (i) use Embedded Software on non-Juniper equipment; (j) use Embedded Software (or make it available for use) on Juniper equipment that the Customer did not originally purchase from Juniper or an authorized Juniper reseller; (k) disclose the results of testing or benchmarking of the Software to any third party without the prior written consent of Juniper; or (l) use the Software in any manner other than as expressly provided herein.

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

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

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

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

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

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

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

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

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

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

15. **Miscellaneous.** This Agreement shall be governed by the laws of the State of California without reference to its conflicts of laws principles. The provisions of the U.N. Convention for the International Sale of Goods shall not apply to this Agreement. For any disputes arising under this Agreement, the Parties hereby consent to the personal and exclusive jurisdiction of, and venue in, the state and federal courts within Santa Clara County, California. This Agreement constitutes the entire and sole agreement between Juniper and the Customer with respect to the Software, and supersedes all prior and contemporaneous agreements relating to the Software, whether oral or written (including any inconsistent terms contained in a purchase order), except that the terms of a separate written agreement executed by an authorized Juniper representative and Customer shall govern to the extent such terms are inconsistent or conflict with terms contained herein. No modification to this Agreement nor any waiver of any rights hereunder shall be effective unless expressly assented to in writing by the party to be charged. If any portion of this Agreement is held invalid, the Parties agree that such invalidity shall not affect the validity of the remainder of this Agreement. This Agreement and associated documentation has been written in the English language, and the Parties agree that the English version will govern. (For Canada: Les parties aux présentés confirment leur volonté que cette convention de même que tous les documents y compris tout avis qui s'y rattaché, soient redigés en langue anglaise. (Translation: The parties confirm that this Agreement and all related documentation is and will be in the English language)).

Table of Contents

List of Figures

List of Tables

PART 1

Virtual Private LAN Service

CHAPTER 1

Virtual Private LAN Service Concepts and Reference Materials

This chapter covers these topics:

Virtual Private LAN Service Overview

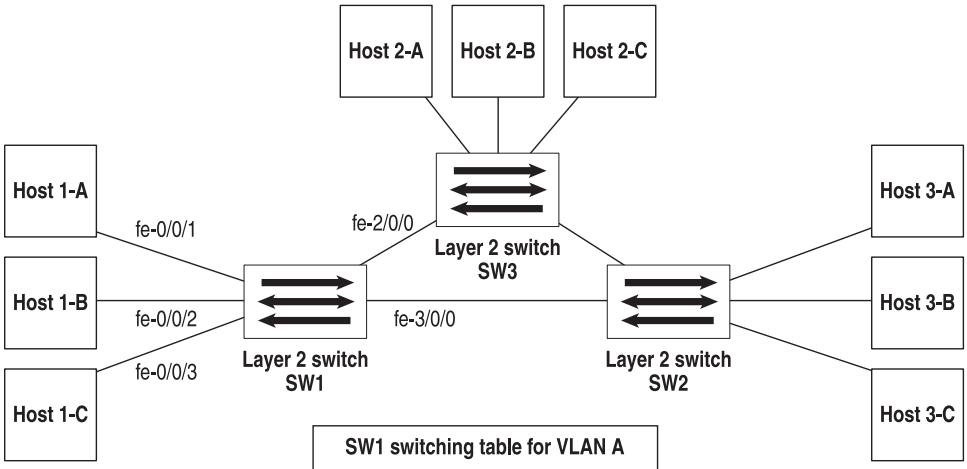
Ethernet is an increasingly important component of a service provider's slate of service offerings. Many customers are requesting the ability to connect local area network (LAN) locations across the country and around the world. To fulfill customer desire, service providers have had to set up complex point-to-point Layer 2 virtual private networks (VPNs) or connect expensive Layer 2 switches to handle traffic.

Virtual private LAN service (VPLS) meets the growing Ethernet needs of service providers and their customers. VPLS is an Ethernet-based multipoint-to-multipoint Layer 2 VPN. With VPLS, multiple Ethernet LAN sites can be connected to each other across an MPLS backbone. To the customer, all sites interconnected by VPLS appear to be on the same Ethernet LAN (even though traffic travels across a service provider network).

This guide explains the background knowledge you need to understand VPLS and provides detailed steps for you to follow to implement it in your network.

Before VPLS, the only way you could connect Ethernet LAN sites together was to set up a non-VPLS Layer 2 VPN or install multiple Layer 2 Ethernet switches. Figure 1 on page 4 shows how three switches can be connected to each other.

Figure 1: Ethernet Switching Example



SW1 switching table for VLAN A

Port	MAC Address
fe-0/0/1	1a1a.1a1a.1a1a
fe-2/0/0	2a2a.2a2a.2a2a
fe-3/0/0	3a3a.3a3a.3a3a

SW1 switching table for VLAN B

Port	MAC Address
fe-0/0/2	1b1b.1b1b.1b1b
fe-2/0/0	2b2b.2b2b.2b2b
fe-3/0/0	3b3b.3b3b.3b3b

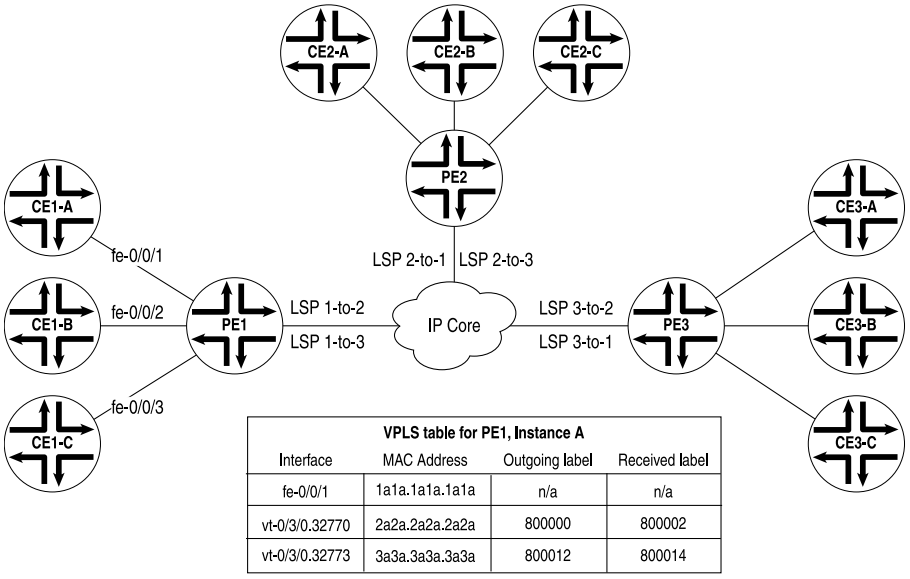
SW1 switching table for VLAN C

Port	MAC Address
fe-0/0/3	1c1c.1c1c.1c1c
fe-2/0/0	2c2c.2c2c.2c2c
fe-3/0/0	3c3c.3c3c.3c3c

A typical switch builds its Layer 2 switching table with MAC address and interface information learned from traffic received from other switches. If a switch does not know how to reach a particular destination, it floods traffic for that destination to all ports except the one where the traffic originated. When information about a previously unknown destination is received, this information is added to the switching table. If a destination is known, the switch sends the traffic directly to the intended recipient through the associated port listed in the switching table.

Figure 2 on page 5 shows a VPLS network comparable to the switch example and explains how VPLS functions similarly to Ethernet switches (assuming a Spanning Tree Protocol is configured).

Figure 2: VPLS Introductory Example



VPLS table for PE1, Instance A

Interface	MAC Address	Outgoing label	Received label
fe-0/0/1	1a1a.1a1a.1a1a	n/a	n/a
vt-0/3/0.32770	2a2a.2a2a.2a2a	800000	800002
vt-0/3/0.32773	3a3a.3a3a.3a3a	800012	800014

VPLS table for PE1, Instance B

Interface	MAC Address	Outgoing label	Received label
fe-0/0/2	1b1b.1b1b.1b1b	n/a	n/a
vt-0/3/0.32771	2b2b.2b2b.2b2b	800004	800006
vt-0/3/0.32774	3b3b.3b3b.3b3b	800016	800018

VPLS table for PE1, Instance C

Interface	MAC Address	Outgoing label	Received label
fe-0/0/3	1c1c.1c1c.1c1c	n/a	n/a
vt-0/3/0.32772	2c2c.2c2c.2c2c	800008	800010
vt-0/3/0.32775	3c3c.3c3c.3c3c	800020	800022

Notice that Layer 2 information gathered by a switch (for example, MAC addresses and interface ports) is included in the VPLS instance table. However, instead of requiring all VPLS interfaces to be physical switch ports, the router allows remote traffic for a VPLS instance to be delivered across an MPLS label-switched path (LSP) and arrive on a virtual port. The virtual port emulates a local, physical port. Traffic can be learned, forwarded, or flooded to the virtual port almost identically to the way traffic is sent to a local port.

The VPLS table learns MAC address and interface information for both physical and virtual ports. If no activity is seen for a particular MAC address, it is purged from the table over time.

As shown in Figure 2 on page 5, the main difference between a physical port and a virtual port is that the router captures additional information from a virtual port—an outgoing MPLS label used to reach the remote site and an incoming MPLS label for VPLS traffic received from the remote site.

When you configure VPLS on a routing platform, a virtual port is generated as a logical interface on a virtual loopback tunnel (**vt**) interface or a label-switched interface (LSI). On Juniper Networks M Series Multiservice Edge Routers and Juniper Networks T Series Core Routers, virtual ports are created dynamically on **vt** interfaces if you install a Physical Interface Card (PIC) that supports virtual tunnels. With VPLS, you must install at least one Tunnel Services, Link Services, or Adaptive Services PIC in each VPLS provider edge (PE) router. On Juniper Networks MX Series Ethernet Services Routers, virtual ports are created dynamically on **vt** interfaces if you configure tunnel services on one of the four

Packet Forwarding Engines (PFEs) included in each DPC. If your routing platform does not offer tunnel services through a PIC or PFE, you can configure VPLS to create virtual ports on LSI logical interfaces.

One property of flooding behavior in VPLS is that traffic received from remote PE routers is never forwarded to other PE routers. This restriction helps prevent loops in the core network. If a customer edge (CE) Ethernet switch has two connections or more to the same PE router, you must enable the Spanning Tree Protocol to prevent loops. For more information on configuring the Spanning Tree Protocol, see the *Junos OS Routing Protocols Configuration Guide*.

The paths carrying VPLS traffic between each PE router participating in a routing instance are called pseudowires. The pseudowires are signaled using either BGP or LDP.

Virtual Private LAN Service System Requirements

To implement VPLS, your system must meet these minimum requirements:

- Junos OS Release 9.5 or later to implement Ethernet VPLS over Frame Relay interface encapsulation on M120 and M320 routers.
- Junos OS Release 9.5 or later to implement Ethernet VPLS over PPP interface encapsulation on M120 and M320 routers.
- Junos OS Release 9.1 or later for nonstop active routing (NSR), VPLS ping on M120, M320, and MX Series routers, and automatic site selection for BGP-signaled VPLS.
- Junos OS Release 9.0 or later for Virtual Spanning Tree Protocol (VSTP) support, 802.1p classification in Bridged Ethernet over ATM mode support, interworking between LDP and BGP signaling in a VPLS instance, and Layer 2 VPLS filters for MX960 routers.
- Junos OS Release 8.4 or later for VPLS with LDP signaling. Also, integrated routing and bridging (IRB) is supported starting in this release.
- Junos OS Release 8.3 or later for point-to-multipoint LSP support on VPLS.
- Junos OS Release 8.2 or later for VPLS support on MX Series routing platform, VPLS graceful Routing Engine switchover (GRES) support, and VPLS support on Gigabit Ethernet IQ2 aggregated Ethernet interfaces.
- Junos OS Release 7.6 or later for VPLS support on LSI logical interfaces.
- Junos OS Release 7.5 or later for multihoming a CE router to multiple PE routers.
- Junos OS Release 7.3 or later for VPLS per-packet load balancing, support for limiting MAC address learning per interface in a VPLS domain, and migration to the VPLS and Layer 2 VPN **signaling** statement at the **[edit protocols bgp groups *group-name* family l2vpn]** hierarchy level.
- Junos OS Release 6.4 or later to implement Ethernet VPLS over ATM LLC interface encapsulation on T Series and M320 routers, to select the tunnel-enabled PICs that provide virtual ports for VPLS operation, and to issue the **show vpls statistics** command.
- Junos OS Release 6.3 or later to clear MAC addresses from the VPLS table and to modify VPLS table timeout intervals.

- Junos OS Release 6.2 or later for VPLS class of service (CoS), VPLS graceful restart, VPLS interinstance bridging and routing, VPLS source and destination MAC address accounting, VPLS virtual port support on the Adaptive Services PIC for M Series routers, and general VPLS support for T Series and M320 routers.
- Junos OS Release 6.1 or later for VPLS policers and filters.
- Junos OS Release 6.0 or later for Ethernet VPLS over ATM LLC interface encapsulation on M Series routers.
- Junos OS Release 5.7 or later for VPLS with BGP signaling and Ethernet VPLS, VLAN VPLS, and extended VLAN VPLS interface encapsulations.
- Two Juniper Networks M Series (except the M160 router), MX Series, T Series, or TX Matrix routing platforms for the provider edge (PE).
- On M Series and T Series routers, one Adaptive Services PIC, Link Services PIC, or Tunnel Services PIC per routing platform to create VPLS virtual ports on **vt** interfaces.
- On M Series and T Series routers, one Fast Ethernet or Gigabit Ethernet PIC per routing platform (from this list):
  - 4-port Fast Ethernet PIC with 10/100 BASE-TX interfaces
  - 1-port, 2-port, or 10-port Gigabit Ethernet PIC
  - 4-port, quad-wide Gigabit Ethernet PIC
  - 1- and 2-port Gigabit Ethernet Intelligent Queuing (IQ) PIC
  - 4- and 8-port Gigabit Ethernet IQ2 PIC with small form-factor pluggable transceivers (SFPs)
  - 1-, 2-, and 4-port Gigabit Ethernet PIC with SFPs
  - 1-port 10-Gigabit Ethernet PIC

Virtual Private LAN Service Terms and Acronyms

V

virtual port A special logical interface that is generated dynamically when you configure VPLS on a PE router. Virtual ports send and receive VPLS traffic for remote PE routers as if the remote VPLS sites had Ethernet-based interfaces directly connected to the local PE router. To generate virtual ports, VPLS PE routing platforms use logical interfaces on a **vt** interface (that is generated by the Tunnel Services PIC, Link Services PIC, Adaptive Services PIC, an LSI interface, or a tunnel services interface configured on MX Series routers).

virtual private LAN service (VPLS) An Ethernet-based multipoint-to-multipoint Layer 2 VPN service used for interconnecting multiple Ethernet LANs across an MPLS backbone. BGP-based VPLS is based on the Internet Engineering Task Force (IETF) Internet draft draft-ietf-l2vpn-vpls-bgp-08.txt, *Virtual Private LAN Service (VPLS) Using BGP for Auto-discovery and Signaling* (expires December 2006). LDP-based VPLS is specified in the IETF draft *Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling*. For more information about VPLS, see the *Junos VPNs Configuration Guide*.

CHAPTER 2

Virtual Private LAN Service Configuration

The following sections show the configuration steps necessary to implement VPLS:

Configuring Routing Protocols on the PE and Core Routers

At a fundamental level, VPLS is a type of Layer 2 VPN. All forms of Layer 2 VPNs require that you configure network protocols to handle the following:

- *intradomain routing* — An interior gateway protocol (IGP) such as Open Shortest Path First (OSPF) or Intermediate System-to-Intermediate System (IS-IS)
- *interdomain routing* — Border Gateway Protocol (BGP)
- *label switching* — Multiprotocol Label Switching (MPLS)
- *path signaling* — Resource Reservation Protocol (RSVP) or Label Distribution Protocol (LDP)

For more information about these protocols and examples of how to configure these protocols to support a Layer 2 VPN, see the *Junos VPNs Configuration Guide*.

NOTE: The 8-port, 12-port, and 48-port dense Fast Ethernet Physical Interface Cards (PICs) cannot push more than two labels onto an MPLS packet. Because of this, we do not recommend that you configure these PICs as core-facing or equivalent interfaces.

Configuring VPLS Encapsulation on CE-Facing Interfaces

There are several types of VPLS interface encapsulation: Ethernet VPLS, Ethernet VPLS over ATM LLC, Ethernet VPLS over Frame Relay, Ethernet VPLS over PPP, VLAN VPLS, extended VLAN VPLS, and flexible Ethernet services. When one of these encapsulations is applied to an interface, a family type of VPLS is enabled by default. The encapsulation types are:

- **ethernet-vpls-fr**—Use Ethernet VPLS over Frame Relay in a VPLS setup when a CE device is connected to a PE device over a time division multiplexing (TDM) link. This encapsulation type enables the PE device to terminate the outer layer 2 Frame Relay connection, use the 802.1p bits inside the inner Ethernet header to classify the packets, look at the MAC address from the Ethernet header, and use it to forward the packet into a given VPLS instance.
- **ether-vpls-over-atm-llc**—Use Ethernet VPLS over ATM LLC encapsulation on ATM2 IQ logical interfaces. Use this encapsulation type to support IEEE 802.1p classification binding on ATM VCs. This encapsulation type enables a VPLS instance to support bridging between Ethernet interfaces and ATM interfaces, as described in RFC 2684,

Multiprotocol Encapsulation over ATM Adaptation Layer 5. When you use this encapsulation type, you configure it on logical interfaces only and you cannot configure multipoint interfaces.

- **ethernet-vpls-fr**—Use in a VPLS setup when a CE device is connected to a PE device over a time division multiplexing (TDM) link. This encapsulation type enables the PE device to terminate the outer layer 2 Frame Relay connection, use the 802.1p bits inside the inner Ethernet header to classify the packets, look at the MAC address from the Ethernet header, and use it to forward the packet into a given VPLS instance.
- **ethernet-vpls-ppp**—Use in a VPLS setup when a CE device is connected to a PE device over a time division multiplexing (TDM) link. This encapsulation type enables the PE device to terminate the outer layer 2 PPP connection, use the 802.1p bits inside the inner Ethernet header to classify the packets, look at the MAC address from the Ethernet header, and use it to forward the packet into a given VPLS instance.
- **extended-vlan-vpls**—Use extended VLAN VPLS encapsulation on Ethernet interfaces that have VLAN 802.1Q tagging and VPLS enabled and that must accept packets carrying TPIDs 0x8100, 0x9100, and 0x9901.

NOTE: The built-in Gigabit Ethernet PIC on the M7i router does not support MPLS.

- **ethernet-vpls**—Use Ethernet VPLS encapsulation on Ethernet interfaces that have VPLS enabled and must accept packets carrying standard Tag Protocol ID (TPID) values.
- **vlan-vpls**—Use VLAN VPLS encapsulation on Ethernet interfaces with VLAN tagging enabled. VLAN VPLS encapsulation supports TPID 0x8100 only. You must configure this encapsulation type on both the physical interface and the logical interface.
- **flexible-ethernet-services**—Use flexible Ethernet services encapsulation when you want to configure multiple per-unit Ethernet encapsulations. This encapsulation type allows you to configure any combination of route, TCC, CCC, and VPLS encapsulations on a single physical port.

Use the following guidelines to configure a VPLS interface:

- For encapsulation type **vlan-vpls**, VLAN IDs 1 through 511 are reserved for normal Ethernet VLANs, IDs 512 through 1023 are reserved for VPLS VLANs on Fast Ethernet interfaces, and IDs 512 through 4094 are reserved for VPLS VLANs on Gigabit Ethernet interfaces.
- For encapsulation type **extended-vlan-vpls**, all VLAN IDs from 1 through 1023 are valid for VPLS VLANs on Fast Ethernet interfaces, and all VLAN IDs from 1 through 4094 are valid for VPLS VLANs on Gigabit Ethernet interfaces. VLAN ID 0 is reserved for priority tagging. For encapsulation type **flexible-ethernet-services**, VLAN IDs from 1 through 511 are no longer reserved for normal VLANs.
- For encapsulation type **flexible-ethernet-services**, VLAN IDs from 1 through 511 are no longer reserved for normal VLANs.

VPLS Interface Encapsulation for an Ethernet Interface

To configure VPLS interface encapsulation for an Ethernet interface, include the **encapsulation** statement at the **[edit interfaces** *interface-fpc/pic/port***]** hierarchy level and select **ethernet-vpls**, **vlan-vpls**, **extended-vlan-vpls**, **flexible-ethernet-services** or **ether-vpls-over-atm-llc** as the encapsulation type. If you select the VLAN VPLS encapsulation, also include the **vlan-vpls** statement at the **[edit interfaces** *ethernet-interface-fpc/pic/port* **unit** *unit-number* **encapsulation]** logical interface hierarchy level. When using either VLAN VPLS or extended VLAN VPLS encapsulations, include the **vlan-tagging** statement at the **[edit interfaces** *ethernet-interface-fpc/pic/port***]** hierarchy level.

VPLS Interface Encapsulation for an ATM2 IQ Interface

To configure VPLS interface encapsulation for an ATM2 IQ interface, include the **encapsulation** statement at the **[edit interfaces at-***fpc/pic/port***]** hierarchy level and select **ether-vpls-over-atm-llc** as the encapsulation type. To configure VPLS interface encapsulation for a Gigabit Ethernet IQ interface or Gigabit Ethernet PICs with small form-factor pluggable transceivers (SFPs), include the **encapsulation** statement at the [**edit interfaces ge-***fe/pic/port*] hierarchy level and select **flexible-ethernet-services** as the encapsulation type.

```
[edit]
interfaces {
   ge-0/1/0 {
      vlan-tagging;
      encapsulation vlan-vpls;
      unit 0 {
         encapsulation vlan-vpls;
         vlan-id 600;
      }
   }
   at-0/2/0 {
      encapsulation ether-vpls-over-atm-llc;
   }
}
```

Configuring LDP Signaling for VPLS

Like other Layer 2 VPNs, you must enable a VPLS instance to isolate VPLS traffic from other network traffic. An important element of a VPLS instance is the signaling protocol. You can configure BGP signaling, LDP signaling, or both BGP and LDP signaling in a VPLS instance.

To configure LDP signaling, you must first enable a VPLS instance to isolate VPLS traffic from other network traffic. To enable a VPLS instance, include the **instance-type vpls** statement at the **[edit routing-instances** *instance-name*] hierarchy level. To configure LDP signaling within the instance, identify the virtual circuit with the **vpls-id** statement and specify the PE routers participating in the instance with the **neighbor** statement:

```
[edit]
routing-instances {
   instance-name {
      instance-type vpls;
```

```
        interface ge-0/1/0.0;
        protocols {
          vpls {
            vpls-id id-name;
            neighbor neighbor-id; # The neighbor-id should be the loopback address of # the
              remote PE router.
          }
        }
      }
    }
```

To fully enable LDP signaling on a PE in a VPLS instance, you must also enable LDP on the loopback interface of the router. To enable LDP on the loopback interface, include the **interface lo0.0** statement at the [**edit protocols ldp**] hierarchy level:

```
[edit]
protocols {
  ldp {
    interface lo0.0;
  }
}
```

For LDP signaling within a VPLS routing instance, the Junos OS supports the following values only:

- FEC—FEC 128
- Control bit—0
- Ethernet pseudowire type—hexadecimal 0x0005

Configuring a VPLS Instance with BGP Signaling

Like other Layer 2 VPNs, you must enable a VPLS instance to isolate VPLS traffic from other network traffic. An important element of a VPLS instance is the signaling protocol. You can configure BGP signaling, LDP signaling, or both BGP and LDP signaling in a VPLS instance.

You must enable a VPLS instance to isolate VPLS traffic from other network traffic. To enable a VPLS instance, include the **instance-type vpls** statement at the **[edit routing-instances *instance-name*]** hierarchy level.

Within the instance, you can define the maximum number of sites that can participate in this VPLS instance, a local site name, and a local site identifier. To configure the maximum number of sites, include the **site-range** statement at the [**edit routing-instances *instance-name* protocols vpls**] hierarchy level. The maximum number of sites is 65,535.

NOTE: The site ID must be less than the site range. If you specify a site ID that is greater than the site range, your connections do not come up, even though the commit operation succeeds.

To configure a site name, include the **site** statement at the **[edit routing-instances *instance-name* protocols vpls]** hierarchy level. To configure the site ID, include the **site-identifier** statement at the **[edit routing-instances *instance-name* protocols vpls site *name*]** hierarchy level.

```
[edit]
routing-instances;
   instance-name {
      instance-type vpls;
      interface ge-0/1/0.0;
      route-distinguisher 10.245.14.218:1;
      vrf-target target:11111:1;
      protocols {
         vpls {
            site-range 10;
            site greenPE1 {
               site-identifier 1;
            }
         }
      }
   }
```

To complete the configuration, you must configure the Layer 2 VPN family for BGP by including the **signaling** statement at the [**edit protocols bgp family l2vpn**] hierarchy level:

```
[edit]
protocols {
   bgp {
      family l2vpn;
         signaling;
      }
   }
}
```

Configuring Interworking Between BGP Signaling and LDP Signaling in VPLS Instances

If you want to configure a VPLS instance with both BGP and LDP-signaled pseudowires, you must configure a VPLS border router. Without a VPLS border router, LDP-signaled PEs and BGP-signaled PEs will be unaware of one another and the VPLS instance will not be fully meshed.

NOTE: Interworking between BGP signaling and LDP signaling in VPLS instances is supported only on MX Series and M320 routers.

To enable interworking between BGP-signaled PE routers and LDP-signaled PE routers, you configure a border router to interconnect both sets of routers within the same VPLS routing instance. You also need to configure mesh groups on the border router to group the sets of PE routers that are fully meshed and which share the same signaling protocol, either BGP or LDP. You can configure multiple mesh groups to map each fully meshed LDP-signaled or BGP-signaled VPLS domain to a mesh group. In the data plane, the border router maintains a common MAC table used to forward traffic between the LDP-signaled and BGP-signaled mesh groups. When forwarding any VPLS traffic received

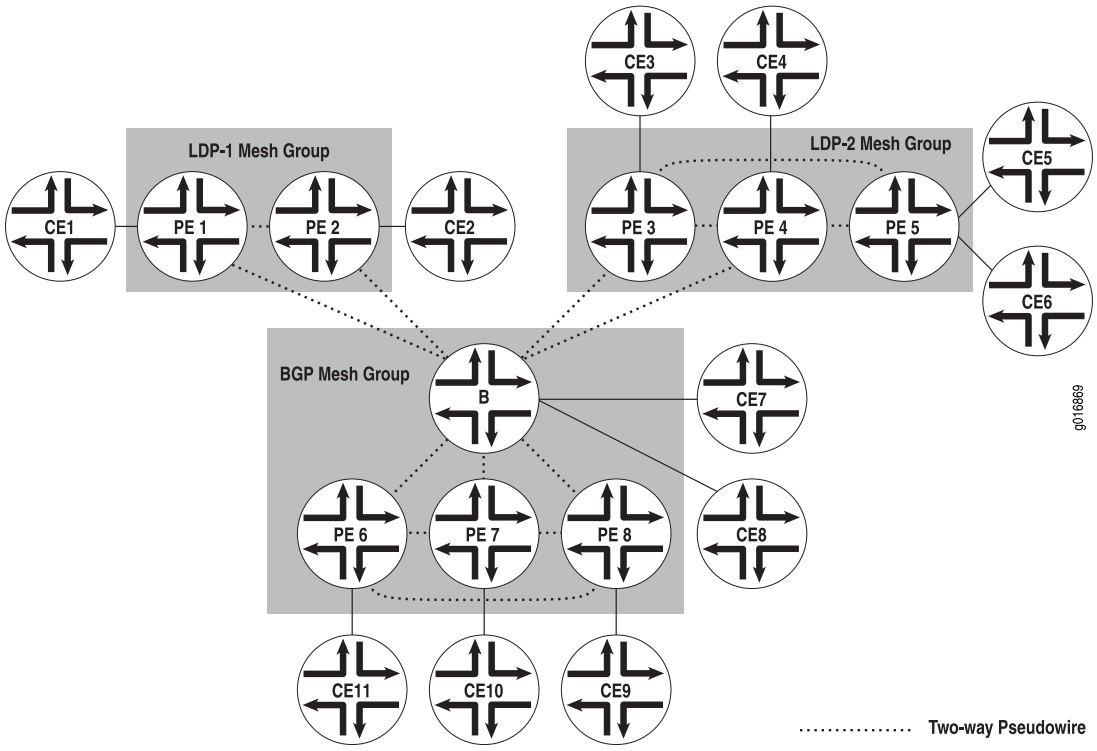
over a PE router pseudowire, the border router ensures that traffic is not forwarded back to the PE routers, which are in same mesh group as the originating PE router.

There is always just one BGP mesh group in a VPLS instance, and it is created automatically when you configure BGP signaling for that instance. You can configure one or more LDP mesh groups. MX Series routers support up to 15 PE mesh groups (including the default BGP mesh group), and M Series and T Series routers support up to 127 PE mesh groups (including the default BGP mesh group).

In Figure 3 on page 15, Routers PE1 and PE2 are in the LDP-signaled mesh group "LDP-1". Routers PE3, PE4, and PE5 are in the LDP-signaled mesh group "LDP-2". Routers PE6, PE7, and PE8 are in the BGP-signaled mesh group. The border router acts as a traditional PE (by connecting to CEs) in addition to being a border router. Every router shown in the topology in Figure 3 on page 15 is in the same VPLS instance, **bgp-ldp-mesh1**.

When Router CE1 sends a frame whose destination MAC address is CE9, PE1 receives the frame and performs a MAC address lookup. The MAC address is not in the PE1 MAC table and so PE1 floods the frame to the other PEs in the LDP-1 mesh group (PE2) and also to Router B, which from the perspective of PE1, are the only members of the VPLS network. When Router B receives the data from PE1, it does not find the MAC address in its MAC table and so it floods the frame to PE3, PE4, PE5, PE6, PE7, and PE8, but not back to PE1 or PE2. The PE routers then perform a MAC table lookup and flood the data to their CE routers.

Figure 3: Topology for BGP/LDP Interworking in a VPLS Instance



In this topology, you configure routers PE6, PE7, and PE8 as you traditionally configure BGP-signaled VPLS routers. You configure routers PE1, PE2, PE3, PE4, and PE5 as you

traditionally configure LDP-signaled VPLS routers. In addition, you create the mesh group LDP-1 for Routers PE1 and PE2 and mesh group LDP-2 for Routers PE3, PE4, and PE5 by including the **mesh-group** *mesh-group-name* statement at the [**edit routing-instances** *routing-instance-name* **protocols vpls**] hierarchy level.

NOTE: The border router can act as a normal PE in addition to being a border router and can support local CE-facing interfaces.

In the example below, the interfaces are CE interfaces. In this case, the router is acting as both a border router and a regular PE router.

To enable interworking between VPLS mesh groups, configure the border router by including the **site** *site-name* statement at the [**edit routing instances** *routing-instance-name* **protocols**] hierarchy level:

```
[edit]
routing-instances {
   bgp-ldp-mesh1 {
      instance-type vpls;
      route-distinguisher 10.245.14.218:1;
      interface fe-1/3/1.0;
      interface fe-1/3/2.0;
   vrf-target target:10:100;
}
protocols {
   vpls {
   site green {
      site-identifier 1;
   }
}
```

Configure LDP signaling with the **vpls-id** and **neighbor** *neighbor-id* statements. You can configure mesh groups LDP-1 and LDP-2 by including the **mesh-group** statement at the [**edit routing instances** *routing-instance-name* **protocols vpls** *vpls-id* and including the **neighbor** *neighbor-id* statement at the [**edit routing-instances** *routing-instance-name* **protocols vpls mesh-group** *mesh-group-name*] hierarchy level:

```
[edit routing-instances bgp-ldp-mesh1 protocols vpls]
vpls-id 100;
mesh-group LDP-1 {
   neighbor 10.1.1.1;
   neighbor 20.1.1.1;
}
mesh-group LDP-2 {
   neighbor 30.1.1.1;
   neighbor 40.1.1.1;
   neighbor 10.1.1.1;
}
```

NOTE: When you configure BGP signaling to interoperate with LDP signaling in a VPLS network, the following features are not supported:

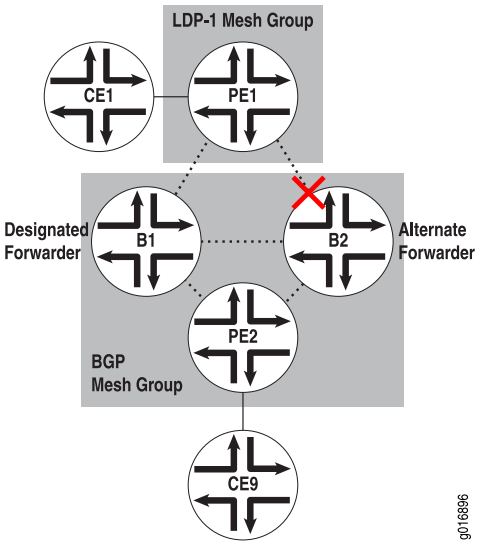
- Point-to-multipoint VPLS
- Integrated routing and bridging

Configuring Multihoming on a VPLS Border Router

Configuring multihoming on VPLS border routers ensures that if one border router is unreachable, BGP/LDP PE connectivity is maintained through the other VPLS border router. With multihoming, one border router is chosen as the designated forwarder for each mesh group. The designated forwarder is chosen through either the BGP or VPLS path-selection procedure. If the designated forwarder loses connectivity with a mesh group, the alternate border router then takes over as designated forwarder for that mesh group. A VPLS instance must be configured with BGP signaling in order for multihoming to work.

Figure 4 on page 17 shows a simplified example of how multihoming works with VPLS border routers. In this example, B1 is the designated forwarder and B2 is the alternate forwarder. If CE1 wanted to send data to CE9, the data would travel from CE1 to PE1, which is part of the LDP-1 mesh group. PE1 would then flood the data to B1 (the designated forwarder), which would forward the data to PE2. It would not send the data to Router B2. PE2 would then send the data to its destination, CE9. If B1 lost connectivity with the LDP-1 mesh group, then B2 would become the designated forwarder. In this case, PE1 would send the data through B2, not through B1.

Figure 4: Multihoming for Border Area Routers



You configure multihoming on border routers by including the **site-identifier** and **multi-homing** statements at the [**edit routing-instances** *routing-instance-name* **protocols**]

hierarchy level. The designated forwarder and alternate forwarder must be configured with the same site identifier.

Router B1

```
[edit routing-instances example protocols]
vpls {
   site mult-home-ldp-1 {
      site-identifier 1;
      mesh-group ldp-1;
      multi-homing;
   }
```

Router B2

```
[edit routing-instances example protocols]
vpls {
   site mult-home-ldp-1 {
      site-identifier 1;
      mesh-group ldp-1;
      multi-homing;
   }
```

For more information on multihoming, see "Option: Configuring VPLS Multihoming with BGP Signaling" on page 19.

Option: Selecting an LSP for the VPLS Routing Instance to Traverse

If you have two or more equal-cost-path LSPs between your VPLS PE router sites, you can select an LSP over which the VPLS traffic will travel. To select an LSP for VPLS traffic, assign the VPLS instance to a BGP community, define a policy that directs community traffic over a specified LSP, and then apply the policy to the forwarding table.

To configure a BGP community, include the **community** *community-name* statement at the **[edit policy-options]** hierarchy level. Be sure to specify the **vrf-export** or **vrf-target** values from the VPLS routing instance as community identifiers with the **members** *community-ids* statement at the **[edit policy-options community** *community-name***]** hierarchy level.

To create a policy that sends community traffic over a specific LSP, include the **community** *community-name* statement at the **[edit policy-options policy-statement** *policy-name* **term** *term-name* **from]** hierarchy level and the **install-nexthop lsp** *lsp-name* statement at the **[edit policy-options policy-statement** *policy-name* **term** *term-name* **then]** hierarchy level. To apply the policy to the forwarding table, include the **export** *policy-name* statement at the **[edit routing-options forwarding-table]** hierarchy level.

```
[edit]
routing-options {
   autonomous-system 69;
   forwarding-table {
      export LSP-policy;
   }
   policy-options {
      policy-statement LSP-policy {
         term a {
            from community gold;
            then {
               install-nexthop lsp pe1-to-pe2;
```

```
                accept;
            }
        }
    }
    community gold members target:11111:1;
  }
}
```

Option: Configuring VPLS Multihoming with BGP Signaling

With VPLS multihoming, you can connect multiple PE router interfaces to one customer site. This feature provides VPLS redundancy should a PE router or PE router interface fail.

To configure multihoming, you must configure the same site IDs on all PE routers and router interfaces that are connected to the same customer site, You must also specify on each PE router which interfaces are connected to the customer site. We recommend that you configure distinct route distinguishers for each multihomed router. Configuring distinct route distinguishers helps with faster convergence when the connection to a primary router goes down. It also requires that the other PE routers maintain additional state information.

To configure a route distinguisher, include the **route-distinguisher** statement at the **[edit routing-instances *instance-name*]** hierarchy level. To assign a site ID, include the **site-identifier** statement at the **[edit routing-instances *instance-name* protocols vpls site *name*]** hierarchy level. To specify the interfaces associated with a site, include the **interface** statement at the **[edit routing-instances *instance-name* protocols vpls site *name*]** hierarchy level.

To connect multiple PE routers to one customer site, you must configure multihoming on each PE router connected to that site. This will prevent routing loops should BGP connectivity fail. BGP automatically determines the primary and backup routers. Alternatively, you can statically configure a primary PE router and backup PE routers for a customer site by specifying the preference value. BGP uses preference values to determine routing paths.

NOTE: Multihoming relies on full BGP connectivity to all other PEs. Configure a dual router reflector topology to provide redundant PE-to-PE BGP connectivity.

To configure multihoming, include the **multi-homing** statement at the **[edit routing-instances *instance-name* protocols vpls site *name*]** hierarchy level. To configure preference value, include the **preference-value** statement at the [**edit routing-instances *instance-name* protocols vpls *site name***] hierarchy level. You can configure the preference value as **primary** or **backup**, or you can specify a preference number. When specifying preference numbers, configure the primary interface with a preference value of 65,535 and any backup interfaces with a number from 1 to 65,534.

When multiple PE router interfaces on a single PE router are connected to one customer site, you must configure an active interface. All traffic will pass through the active interface unless this interface fails, in which case a backup interface will become the active interface.

To specify a multihomed interface as the primary interface for a site, include the **active-interface** statement at the [**edit routing-instances *instance-name* protocols vpls site *name***] hierarchy level. The interface that you specify is called the primary interface. If the primary interface goes down, an alternate interface becomes the active interface. Once the primary interface comes back up, the primary interface becomes the active interface once again and the alternate interface becomes inactive.

If you do not want to specify a primary multihomed interface, you can use the **any** option. With the **any** option, the router dynamically chooses an active interface. If the active interface goes down, an alternate interface becomes the active interface. Once the down interface comes back up, it stays inactive.

If no active interfaces are configured at the site level, it is assumed that all traffic for a VPLS site travels through a single, nonmultihomed PE router.

NOTE: If you add a direct connection between CE devices that are multihomed to the same VPLS site on different PE routers, traffic loops and loss of connectivity might occur. We do not recommend this topology.

The following example shows a multihoming configuration with two PE routers that are connected to a single customer site. Note in the configuration that PE1 is the primary router and PE2 is the backup router.

Router PE1

```
[edit]
routing-instances {
   green {
      instance-type vpls;
      interface fe-0/1/3.0;
      route-distinguisher 10.255.14.218:1;
      vrf-target target:11111:1;
      protocols {
         vpls {
            site-range 10;
            site green4 {
               site-identifier 4;
               multi-homing; # Ensures that BGP is established before forwarding on the
                  # site member interfaces.
               preference value 65535;
               interface fe-1/1/3.0;
            }
         }
      }
   }
}
```

Router PE2

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

```
    green {
      instance-type vpls;
      interface fe-0/1/0.0;
      route-distinguisher 10.255.14.219:1;
      vrf-target target:11111:1;
      protocols {
        vpls {
          site-range 10;
          site green4 {
            site-identifier 4;
            multi-homing;
            preference value 1;
            interface fe-0/1/0.0;
          }
        }
      }
    }
  }
```

The following example shows a multihoming configuration with one PE router with multiple interfaces that are connected to a single customer site.

Router PE3

```
[edit]
routing-instances {
  green {
    instance-type vpls;
    interface fe-1/1/0.0;
    interface fe-1/2/0.0;
    interface fe-1/3/0.0;
    route-distinguisher 10.255.14.218:1;
    vrf-target target:11111:1;
    protocols {
      vpls {
        site-range 10;
        site green4 {
          site-identifier 4;
          active-interface any;
          interface fe-1/1/0.0;
          interface fe-1/2/0.0;
          interface fe-1/3/0.0;
        }
      }
    }
  }
}
```

For more information on VPLS multihoming, see the *Junos VPNs Configuration Guide*.

Option: Configuring VPLS Traffic Flooding over a Point-to-Multipoint LSP

In each VPLS routing instance, you can configure a dedicated point-to-multipoint LSP to carry all unknown unicast, broadcast, and multicast traffic. Enabling this feature increases the efficiency of your network, because duplicate copies of flooded traffic do not have to be created for each PE router in the VPLS routing instance. Figure 5 on page 22 shows how flooded traffic reaches PE routers in a VPLS routing instance when a

point-to-multipoint LSP is not configured for flooding. Figure 6 on page 22 shows an example of a VPLS routing instance configured with point-to-multipoint LSP flooding.

NOTE: You cannot configure point-to-multipoint LSP flooding if your VPLS network is configured for interoperability between BGP and LDP signaling.

Figure 5: Traditional Flooding in a VPLS Routing Instance

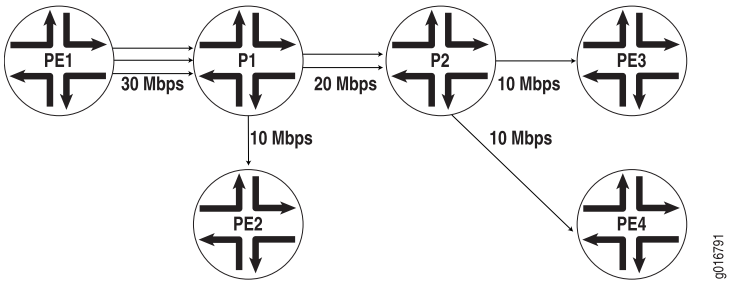
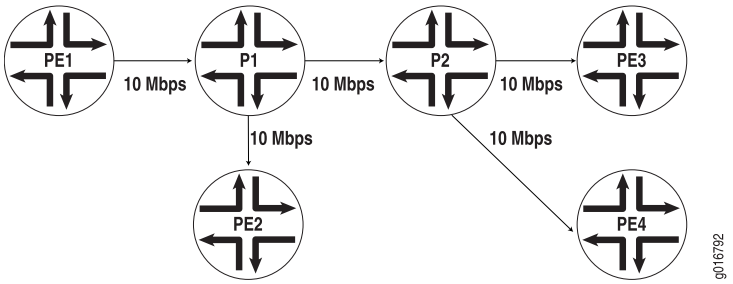


Figure 6: VPLS Routing Instance with Point-to-Multipoint LSP Flooding



You have three options when configuring a point-to-multipoint LSP for flooding:

- Static point-to-multipoint LSP—Configure this option to control which path each PE sub-LSP takes. When using this option, ensure that all PEs within the VPLS routing instance are part of the static point-to-multipoint LSP. When you add PEs to the VPLS routing instance, you must configure a sub-LSP for the new PE and add the sub-LSP to the static point-to-multipoint LSP. To configure a static point-to-multipoint LSP, include the **label-switched-path** ***path-name*** statement at the [**edit protocols mpls**] hierarchy level.
- Dynamic point-to-multipoint LSP with a preconfigured template—Configure this option to create a dynamic point-to-multipoint LSP with specific parameters such as link protection and optimized time. With this option, newly added PEs are automatically added to the point-to-multipoint LSP. To use the preconfigured template, include the **template** statement at the [**edit protocols mpls label-switch-path** ***path-name***] hierarchy level.
- Dynamic point-to-multipoint LSP with a default template—Configure this option to automatically create a dynamic point-to-multipoint LSP with default parameters. With

this option, newly added PEs are automatically added to the point-to-multipoint LSP. To use a default template, include the **default-template** statement at the [**edit routing-instances** *routing-instance-name* **provider-tunnel rsvp-te label-switched-path-template**] hierarchy level.

To define the parameters for a static point-to-multipoint LSP, include the **label-switched-path** *path-name* statement at the [**edit protocols mpls**] hierarchy level:

```
[edit]
protocols {
   mpls {
      label-switched-path vpls-bar-p2mp-s21_lsp_a {
         to 192.168.1.1
         p2mp vpls-bar-p2mp-lsp;
      }
      label-switched-path vpls-bar-p2mp-s21_lsp_b {
         to 192.168.1.2
         p2mp vpls-bar-p2mp-lsp;
      }
   }
}
```

To add a new PE router to the static point-to-multipoint LSP, include the **label-switched-path** *sub-path-name* statement at the [**edit protocols mpls**] hierarchy level:

```
[edit]
protocols {
   mpls {
      label-switched-path added-PE3 {
         to 1.1.1.1
         p2mp vpls-bar-p2mp-lsp;
      }
   }
}
```

For more information on configuring static and dynamic point-to-multipoint LSPs, see the *Junos MPLS Applications Configuration Guide*.

To enable this feature, configure either the **static** or **label-switched-path-template** options for the **rsvp-te** statement at the [**edit routing-instance** *routing-instance-name* **provider-tunnel**] hierarchy level:

```
[edit]
routing-instance foo {
   provider-tunnel {
      rsvp-te {
         static-lsp vpls-bar-p2mp-lsp;
      }
   }
}
```

To verify your work, enter the **show vpls connection extensive** command:

```
Router_1# show vpls connection extensive
....
```

```
status-vector: BF
connection-site Type St Time last up # Up trans
2 rmtUpJan 31 10:14:37 2007 1
Local interface: lsi.32768, Status: Up, Encapsulation: VPLS
Description: Intf -vpls VPLS-A local site 1 remote site 2
Remote PE: 10.255.164.2, Negotiated control-word: No
Incoming label: 262153, Outgoing label: 800000
RSVP-TE P2MP lsp:
Ingress branch LSP: 13:vpls:10.255.164.1:BPLS-A, State: Up
Egress branch LSP: 4:vpls:10.255.164.2:VPLS-A, Statue: Up
TimeEventInterface/Lb1/PE
Jan 31 10:14:37 2007 status update timer
Ingress RSVP-TE P2MP LSP: 11:vpls:10.255.164.1:VPLS-A, Flood next-hop ID: 476
```

Option: Configuring Automatic Site Selection

You can configure BGP-signaled VPLS instances to automatically specify the site IDs for the routers participating in the VPLS domain. Site IDs help to minimize label usage in VPLS instances with numerous PE routers.

The **automatic-site-id** statement includes the following options:

- **startup-wait-time**—Time to wait at startup to receive all VPLS information for configured route targets from other PE routers.
- **new-site-wait-time**—Time to wait to receive VPLS information from a newly configured routing instance or a new site. Effectively, it is the time to wait before a site makes an attempt to locate an unused site ID for its claim advertisement.
- **collision-detect-time**—Time to wait after issuing a claim advertisement before the PE router can start using the site ID if it does not receive a competing claim. If the PE router receives a competing claim within this time interval, it runs a collision resolution procedure. Explicitly configured site IDs always take precedence over automatically generated site IDs.
- **reclaim-wait-time**—Time to wait before attempting to claim a site ID after a collision. There are default values for all of these options, so they do not need to be explicitly configured.

To configure a VPLS automatic site ID, include the **automatic-site-id** statement at the [**edit routing-instances** *routing-instance-name* **protocols vpls site** *site-name*] hierarchy level:

```
[edit]
routing-instances {
   vpls instance 1 {
      protocols {
         vpls {
            site vpls instance 1 {
               automatic-site-id;
            }
         }
      }
   }
```

```
}
```

Option: Configuring VPLS to Use LSI Interfaces

On M Series and T Series routers, VPLS uses tunnel-based PICs to create virtual ports on **vt** interfaces. If you do not have a tunnel-based PIC installed on your M Series or T Series router, you can still configure VPLS by using label-switched interfaces (LSIs) to support the virtual ports. Use of LSI interfaces requires the use of Ethernet-based PICs installed in an Enhanced FPC.

To use LSI interfaces for VPLS instead of **vt** interfaces, include the **no-tunnel-services** statement at the **[edit routing-instances *instance-name* protocols vpls]** hierarchy level.

```
[edit routing-instances]
instance-name  {
  protocols {
    vpls {
      no-tunnel-services;
    }
  }
}
```

NOTE: The following interface types do not support the use of LSI interfaces with VPLS:

- Aggregated SONET/SDH interfaces (cannot be used as the core-facing interface)
- Channelized interfaces (cannot be used as the core-facing interface)
- ATM1 interfaces

Option: Configuring Tunnel Services on MX Series Routers

MX Series routers use Dense Port Concentrators (DPCs) with built-in physical ports, which means that you do not insert PICs on the router. Instead, you configure tunnel interfaces on one of the four Packet Forwarding Engines (PFEs) that are on each DPC.

To create tunnel interfaces on an MX Series router, include the **tunnel-services** statement at the [**edit chassis fpc *slot-number* pic *number***] hierarchy level. To configure the bandwidth for a tunnel interface, include the **bandwidth** statement at the [**edit chassis fpc *slot-number* pic *number***] hierarchy level.

The following example shows a tunnel interface with 1 Gbps of bandwidth configured on PFE 1 of the DPC installed in slot 4 of an MX Series router:

```
[edit chassis]
fpc 4;
pic 1 {
  tunnel services {
    bandwidth 1g;
  }
```

```
}
```

Once you have configured a tunnel interface on a PFE, you can treat this interface as a standard tunnel interface and proceed with a standard VPLS configuration. For more information, see the *Junos OS System Basics Configuration Guide*.

Configuring Integrated Routing and Bridging in a VPLS Instance (MX Series Routers Only)

Integrated routing and bridging (IRB) over VPLS cannot be used in conjunction with the **vlan-id all** statement. One or more Layer 2 logical interfaces must be configured inside the instance in order for IRB to function properly.

To configure IRB within a VPLS instance, include the **routing-interface** *irb-interface-name* statement at the [**edit routing-instances** *routing-instance-name* **instance-type vpls**] hierarchy level:

```
[edit]
routing-instances {
   marketing {
      instance-type vpls;
      route-distinguisher 11.11.11.11:10;
      vrf-target target:100:100;
      interface ae0.100;
      interface ae0.200;
      routing-interface irb.1234;
   }
}
```

Configuring VLAN IDs in a VPLS Instance (MX Series Routers Only)

You can configure VLAN identifiers for a VPLS instance in the following ways:

- By using the **input-vlan-map** and the **output-vlan-map** statements at the [**edit interfaces**] hierarchy level. For more information, see the *Junos OS Network Interfaces Configuration Guide* and *Junos OS Class of Service Configuration Guide*.
- By using the **vlan-id** or **vlan-tags** statements at the [**edit routing-instances** *routing-instance-name* **instance-type vpls**] hierarchy level.

The **vlan-id** and **vlan-tags** statements are used to perform the following functions:

- Translate, or normalize, the VLAN tags of received packets received into a learn VLAN identifier.
- Create multiple learning domains that each contain a VLAN identifier. A learning domain is a MAC address database to which MAC addresses are added based on the VLAN identifier.

For more information about how VLAN tags are processed and translated, see the *Junos MX Series Layer 2 Configuration Guide*.

To configure VLAN identifiers for a VPLS instance, include the **vlan-id** or **vlan-tags** statement at the [**edit routing-instances** *routing-instance-name* **instance-type vpls**] hierarchy level.

NOTE: You cannot configure VLAN mapping using the input-vlan-map and output-vlan-map statements if you configure a VLAN identifier for a VPLS instance using the vlan-id or vlan-tags statements.

```
[edit]
routing-instances {
   marketing {
      instance-type vpls;
      vlan-id 401;
      route-distinguisher 11.11.11.11:10;
      vrf-target target:100:100;
      interface ae0.100;
      interface ae0.200;
   }
}
```

Defining a VPLS Firewall Policer

You can configure filters, policers, and broadcast and unknown filters to determine which kind of traffic is allowed into and out of a VPLS domain. You can apply these filters and policers to CE-facing interfaces only.

To process traffic as it enters a VPLS domain, you can define a firewall policer and apply it to the input interface. To define policer characteristics for incoming VPLS traffic, include the **bandwidth-limit** and **burst-size-limit** statements at the **[edit firewall policer** ***policer-name*** **if-exceeding]** hierarchy level. Then, specify statements to implement the desired action (for example, **discard**) for the policed traffic at the **[edit firewall policer** ***policer-name*** **then]** hierarchy level. To apply the policer to a CE-facing interface, include the **input** or **output** statements and the name of the policer at the **[edit interfaces** ***interface-name*** **unit** ***unit-number*** **family vpls policer]** hierarchy level.

```
[edit]
interfaces {
   ge-2/1/0 {
      vlan-tagging;
      mtu 1544;
      encapsulation vlan-vpls;
      unit 0 {
         encapsulation vlan-vpls;
         vlan-id 600;
         family vpls {
            policer {
               input vpls-policer;
            }
         }
      }
   }
```

```
}
firewall {
   policer {
      vpls-policer {
         if-exceeding {
            bandwidth-limit 5m;
            burst-size-limit 1m;
         }
         then discard;
      }
   }
}
```

Defining a VPLS Firewall Filter

You can configure filters, policers, and broadcast and unknown filters to determine which kind of traffic is allowed into and out of a VPLS domain. You can apply these filters and policers to CE-facing interfaces only.

To process traffic as it exits a VPLS domain, you can define a firewall filter and apply it to the output interface. To configure match conditions for a firewall filter, include the **interface-group**, **source-mac-address**, **destination-mac-address**, **ethernet-type**, or **vlan-ethernet-type** statements at the **[edit firewall family vpls filter *filter-name* term *term-name* from]** hierarchy level. Then, implement the desired action (for example, **discard**) for the traffic at the **[edit firewall family vpls filter *filter-name* term *term-name* then]** hierarchy level. To apply the filter to a CE-facing interface, include the **input**, **output**, or **group** statements at the **[edit interfaces *interface-name* unit *unit-number* family vpls filter]** hierarchy level.

```
[edit]
interfaces {
   fe-2/1/1 {
      vlan-tagging;
      mtu 1544;
      encapsulation vlan-vpls;
      unit 0 {
         encapsulation vlan-vpls;
         vlan-id 600;
         family vpls {
            filter {
               output vpls-out-filter;
            }
         }
      }
   }
}
firewall {
   family vpls {
      filter vpls-out-filter {
         interface-specific;
         term 1 {
            from {
               source-mac-address {
                  00.10.10.10.11.18/48;
```

```
            }
          }
          then {
            count count.ce2;
            accept;
          }
        }
        term 2 {
          then accept;
        }
      }
    }
  }
```

NOTE:

- Output filters do not work for broadcast, multicast, and unknown unicast traffic.
- If an IRB interface is configured as part of a VPLS routing instance, VPLS filters might not filter packets that are destined to the IRB interface. This can be configured by installing filters that match Layer 3 fields for the the IRB interface.
- If you apply a firewall filter to discard a source MAC address, the MAC address is not deleted from the MAC address table.

Restricting Broadcast Packets in VPLS

You can configure filters, policers, and broadcast and unknown filters to determine which kind of traffic is allowed into and out of a VPLS domain. You can apply these filters and policers to CE-facing interfaces only.

To restrict the flow of broadcast and unknown unicast packets into a VPLS domain, you must create a firewall filter and apply the filter to one of the forwarding tables of the VPLS routing instance. When you apply a filter in this way, the filter processes traffic from all interfaces in the instance, including **vt** interfaces. To configure match conditions for a VPLS-based firewall filter, include the **source-mac-address**, **destination-mac-address**, **interface-group**, **ethernet-type**, or **vlan-ethernet-type** statements at the **[edit firewall family vpls filter *filter-name* term *term-name* from]** hierarchy level. Then, specify statements to activate the desired action (for example, **discard**) for the matched packets at the **[edit firewall family vpls filter *filter-name* term *term-name* then]** hierarchy level.

To apply the filter to the broadcast and unknown unicast table of a VPLS routing instance, include the **input** statement and the name of the filter at the **[edit routing-instances *instance-name* forwarding-options family vpls flood]** hierarchy level. To apply the filter to the destination MAC address table of a VPLS routing instance, include the **input** statement and the name of the filter at the **[edit routing-instances *instance-name* forwarding-options family vpls filter]** hierarchy level.

```
[edit]
firewall {
```

```
    family vpls {
      filter vpls-flood {
        term 1 {
          from {
            destination-mac-address (broadcast | multicast | unknown-unicast) {
              # The broadcast, multicast,
              # and unknown-unicast options apply to MX Series
              # routers only.
              00.90.69.dc.95.3b/48;
            }
          }
          then discard;
        }
        term 2 {
          then accept;
        }
      }
    }
  }
  routing-instances {
    green {
      forwarding-options {
        family vpls {
          (flood | filter) {
            input vpls-flood;
          }
        }
      }
    }
  }
```

When you configure VPLS, a priority filter for Spanning Tree Protocol (STP) bridge protocol data units (BPDUs) is enabled by default. This BPDU filter matches on the well-known STP MAC address of **01:80:c2:00:00:00/24** and applies high priority to this traffic.

For more information on VPLS policers and filters, see the *Junos Policy Framework Configuration Guide* and the *Junos VPNs Configuration Guide.*

Option: Enabling VPLS Class of Service

For Junos OS Release 6.2 or later, you can configure class of service (CoS) for all interfaces in the VPLS domain. CoS information is sent across the MPLS backbone and is preserved for all VPLS traffic processed by local interfaces, virtual ports, and remote interfaces.

For more information on configuring CoS, see the *Junos Class of Service Configuration Guide*.

Option: Enabling VPLS Graceful Restart

VPLS graceful restart allows you to continue forwarding VPLS traffic across the core MPLS network even if one of the routers in the forwarding path restarts. Graceful restart for VPLS functions the same way as Layer 2 VPN graceful restart. To configure graceful restart for VPLS, include the **graceful-restart** statement at the **[edit routing-options]** hierarchy level on all PE and core routers.

```
[edit]
routing-options {
   graceful-restart;
}
```

For more information on graceful restart, see the *Junos OS High Availability Configuration Guide*.

Configuring the VPLS MAC Address Timeout

You can fine-tune your VPLS domain by clearing MAC address entries from the VPLS table or modifying the default timeout interval for the VPLS table.

NOTE: On MX Series routers running Junos OS Release 8.4 and later, you can set the expiration time of entries in the MAC table only for the entire router, not for specific VPLS routing instances. To set the expiration for the entire router, include the mac-table-aging-time seconds statement at the [edit protocols l2-learning] hierarchy level. Do not include the mac-table-aging-time statement at the [edit routing-instances routing-instance-name protocols vpls] hierarchy level on MX Series routers running Junos OS Release 8.4 and later.

To clear all MAC address entries from the VPLS table, issue the **clear vpls mac-address** command. Add the **logical-system** ***logical-system-name*** option to clear entries within a logical system and include the **instance** ***instance-name*** option to clear entries in a specific VPLS instance. Use the ***mac-address*** option to remove individual MAC addresses.

To configure the VPLS table timeout interval, include the **mac-table-aging-time** statement at the **[edit routing-instances** ***instance-name*** **protocols vpls]** hierarchy level. The default interval is 300 seconds, with a minimum of 10 seconds and a maximum of 1 million seconds. As a general rule, you can configure longer values for small, stable VPLS networks and shorter values for large, dynamic VPLS networks. If no traffic is received for a specific MAC address, M Series and T Series routers wait one additional interval before automatically clearing MAC address entries from the VPLS table. MX Series routers do not wait for this interval.

```
[edit]
routing-instances {
   instance-name {
      protocols {
         vpls {
            mac-table-aging-time seconds;
         }
      }
   }
}
```

Option: Configuring VPLS Interinstance Bridging and Routing

To deliver interinstance traffic between two or more VPLS instances, or between a VPLS instance and a Layer 3 VPN routing instance, you must use a logical tunnel interface. Originally designed to interconnect logical systems, the logical tunnel interface acts as a point-to-point connection between instances. A logical tunnel interface can be generated by a Tunnel Services PIC installed on an Enhanced FPC in your routing platform, an integrated Adaptive Services Module installed in an M7i router, or a tunnel services interface configured on MX Series routers. To configure a logical tunnel interface, include the **lt-*fpc/pic/*0** statement at the **[edit interfaces]** hierarchy level. Keep in mind these rules when you connect instances:

- You need to configure both endpoints of the logical tunnel. Configure the first logical tunnel interface in the VPLS instance and the second within the instance you want to interconnect to the VPLS domain.
- Choose one of several interface encapsulation types for your logical tunnel interface peers. Your choices are Ethernet, Ethernet circuit cross-connect (CCC), Ethernet VPLS, Frame Relay, Frame Relay CCC, VLAN, VLAN CCC, and VLAN VPLS. Include one of these choices with the **encapsulation** statement at the **[edit interfaces *lt-fpc/pic/*0 unit *unit-number*]** hierarchy level.
- Depending on the encapsulation type you select, specify a corresponding data-link connection identifier (DLCI) number for Frame Relay or a VLAN identifier for VLAN encapsulations on your logical tunnel interface peers. To configure the DLCI or VLAN identifier, include the **dlci** or **vlan-id** statement at the **[edit interfaces lt-*fpc/pic/*0 unit *unit-number*]** hierarchy level.
- Your choice of protocol family for the logical tunnel interface also is determined by your selection of an encapsulation type. For Ethernet VPLS and VLAN VPLS, family **vpls** is assigned by default. For all other Ethernet and VLAN encapsulation types, include the **mpls** or **inet** statement at the **[edit interfaces lt-*fpc/pic/*0 unit *unit-number* family]** hierarchy level. For Frame Relay encapsulation types, you can configure any of the available protocol families: **ccc**, **inet**, **inet6**, **iso**, **mpls**, or **tcc**.
- Be sure to match the logical interface unit numbers of the peering logical tunnel interfaces. To configure, include the **peer-unit** statement at the **[edit interfaces lt-*fpc/pic/*0 unit *unit-number*]** hierarchy level.

```
[edit]
interfaces {
   lt-fpc/pic/0 {
      unit unit-number {
         encapsulation (ethernet | ethernet-ccc | ethernet-vpls | frame-relay |
            frame-relay-ccc | vlan | vlan-ccc | vlan-vpls);
         peer-unit number; # The logical unit number of the peering lt interface.
         dlci dlci-number;
         vlan-id vlan-number;
         family (ccc | inet | inet6 | iso | mpls | tcc);
      }
   }
}
```

```
routing-instances {
  vpls-instance-name {
    interface ge-fpc/pic/port.unit-number;
    interface lt-0/0/0.1;
    ...
    second-instance-name {
      interface at-fpc pic/port.unit-number;
      interface lt-0/0/0.2;
      ...
    }
  }
}
```

Option: Selecting Interfaces to Process VPLS Traffic

On M Series and T Series routers, the PICs that can create VPLS virtual ports dynamically from **vt** interfaces include the Tunnel Services PIC, the Link Services PIC, and the Adaptive Services PIC. On MX Series routers, logical tunnel interfaces configured by including the **tunnel-services** statement at the [**edit chassis fpc** *slot-number* **pic** *number*] hierarchy level can create VPLS virtual ports dynamically from **vt** interfaces.

By default, the Junos OS automatically and randomly selects **vt** interfaces to act as VPLS virtual ports in a round-robin fashion. However, if your routing platform contains two or more of these tunnel-enabled interfaces, you can manually select which interfaces process traffic for each VPLS domain.

You can select an interface to be the primary device responsible for VPLS traffic processing. You can also select a group of interfaces to share responsibility for VPLS traffic processing. When the primary interface is operating normally, it handles all VPLS-related tasks. If the primary device is not available, any interfaces included in the VPLS interface group assume responsibility.

To select an interface to be the primary device responsible for VPLS traffic processing, include the **primary** statement at the **[edit routing-instances** *instance-name* **protocols vpls tunnel-services]** hierarchy level. To select a group of interfaces to share responsibility for VPLS traffic processing, include the **devices** statement at the **[edit routing-instances** *instance-name* **protocols vpls tunnel-services]** hierarchy level.

```
[edit]
routing-instances {
  instance-name {
    protocols {
      vpls {
        tunnel-services {
          devices [vt-0/0/0 vt-1/0/0 vt-2/0/0];
          primary vt-0/0/0;
        }
      }
    }
  }
}
```

Option: Limiting the Number of MAC Addresses Learned on a VPLS Interface

There are three main levels where you can configure MAC address limits:

- **interface-mac-limit**—This statement allows you to specify a limit for MAC addresses at an interface level. For VPLS, you can include the **interface-mac-limit** statement at the [**edit logical-systems** *logical-system-name* **routing-instances** *routing-instance-name* **protocols vpls**], [**edit logical-systems** *logical-system-name* **routing-instances** *routing-instance-name* **protocols vpls site** *site-name* **interfaces** *interface-name*], [**edit routing-instances** *routing-instance-name* **protocols vpls**], or [**edit routing-instances** *routing-instance-name* **protocols vpls site** *site-name* **interfaces** *interface-name*] hierarchy level. For MX Series routers only, you can specify what the router does with additional MAC addresses once the MAC address limit is reached. The default behavior is for the router to flood the packet, but you can alternatively include the **packet-action drop** option to have the router drop the packets. The default MAC address table size for each interface is 1024 addresses.
- **mac-table-size**—This statement allows you to specify a limit for MAC addresses at a domain level. For VPLS, you can include the **mac-table-size** statement at the [**edit logical-systems** *logical-system-name* **routing-instances** *routing-instance-name* **protocols vpls**] or [**edit routing-instances** *routing-instance-name* **protocols vpls**] hierarchy level. The default MAC address table size for each domain is 5120 addresses.
- **global-mac-limit** (MX Series routers only)—This statement allows you to specify a limit for MAC addresses for all interfaces and all domains for the entire router. You can include the **global-mac-limit** statement at the [**edit protocols l2-learning**] hierarchy level. The default MAC address table size for the entire system is 393,215 addresses.

NOTE: If you manually configure a MAC address limit, you must ensure that values for interface limits (such as the **interface-mac-limit**) are set lower than domain limits (such as **mac-table-size**), and the domain limits are set lower than global limits (such as **global-mac-limit**). If a value for a more specific limit is set higher than a more global limit, the commit operation fails.

The range of values for the **interface-mac-limit** statement is 16 through 65,536. The output of the **show vpls statistics** command displays the results of configuring interface-level MAC address limitations.

```
[edit]
routing-instances {
   instance-name {
      protocols {
         vpls {
            interface-mac-limit number;
            site site-name {
               interface interface-name {
                  interface-mac-limit number;
               }
            }
```

```
        }
      }
    }
  }
```

Option: Optimizing VPLS Traffic Flows

To improve the performance of VPLS traffic processing in your routing platform, you can implement the following features:

- To optimize VPLS traffic flows across multiple paths, you can enable per-packet load balancing. To enable per-packet load balancing, include the **load-balance per-packet** statement at the **[edit policy-options policy-statement *policy-name* term *term-name* then]** hierarchy level and apply the policy to the forwarding table with the **export *policy-name*** statement at the **[edit routing-options forwarding-table]** hierarchy level.
- To optimize hashing of source and destination MAC addresses within VPLS traffic flows, include the **source-mac** and **destination-mac** statements at the **[edit forwarding-options hash-key family multiservice]** hierarchy level.

For more information about load balancing and hash keys, see the *Junos Policy Framework Configuration Guide*.

Option: Aggregated Interfaces for VPLS

You can configure aggregated Ethernet interfaces between CE devices and PE routers for VPLS routing instances. Traffic is load-balanced across all of the links in the aggregated interface. If one or more links in the aggregated interface fails, the traffic is switched to the remaining links.

In the example below, 0 is the interface instance number that completes the link association. This number can be from 0 through 127, for a total of 128 aggregated interfaces. The VPLS encapsulation types supported on aggregated Ethernet interfaces are **ethernet-vpls**, **vlan-vpls**, or **extended-vlan-vpls**.

```
[edit]
interfaces ae0
vlan-tagging;
encapsulation vlan-vpls;
unit 0 {
  vlan-id 100;
}
```

The aggregated Ethernet interface must also be configured for a VPLS routing instance. Use the standard VPLS routing instance configuration on aggregated Ethernet interfaces.

For more information about how to configure aggregated Ethernet interfaces, see the *Junos Network Interfaces Configuration Guide*.

Synchronizing the Routing Engine Configuration

When you configure nonstop active routing, you must also include the **commit synchronize** statement at the **[edit system]** hierarchy level so that configuration changes are synchronized on both Routing Engines:

```
[edit system]
commit synchronize;
```

If you try to commit the nonstop active routing configuration without including the **commit synchronize** statement, the commit operation fails.

If you issue the **commit synchronize** command at the **[edit]** hierarchy level on the backup Routing Engine, the Junos system software displays a warning and commits the candidate configuration.

NOTE: A newly inserted backup Routing Engine automatically synchronizes its configuration with the master Routing Engine configuration.

When you configure nonstop active routing, you can bring the backup Routing Engine online after the master Routing Engine is already running. There is no requirement to start the two Routing Engines simultaneously.

Verifying VPLS Nonstop Active Routing Operation

To see whether or not nonstop active routing is enabled, issue the **show task replication** command.

NOTE: You must issue the **show task replication** command on the master Routing Engine. This command is not supported on the backup Routing Engine.

For more information on this command, see the *Junos OS System Basics and Services Command Reference*.

Tracing VPLS Nonstop Active Routing Synchronization Events

To trace the label and logical interface association that VPLS receives from the kernel replication state, include the **nsr-synchronization** statement at the **[edit routing-options traceoptions flag]** hierarchy level. This flag also traces the Layer 2 VPN signaling state replicated from routes advertised by BGP.

```
[edit routing-options]
traceoptions {
   flag nsr-synchronization;
}
```

Option: Configuring the Spanning Tree Protocol and VPLS on MX Series Routers

If multiple routers on a customer site are connected to the same PE, you should enable the Spanning Tree Protocol on that PE. To configure RSTP or MSTP and VPLS simultaneously, include the **rstp** or **mstp** statement at the [**edit instance-type layer2-control**] hierarchy level:

```
[edit]
instance-type layer2-control;
protocols {
   rstp {
      interface interface name;
      force-version stp; # To run STP instead of RSTP
   }
}
```

The Per-VLAN Spanning Tree (PVST) protocol maintains a separate spanning-tree instance for each VLAN. To enable PVST for a specific VLAN ID, there should be a VPLS instance with that VLAN ID and all of the logical interfaces assigned to that instance should have the same matching VLAN ID. To configure PVST with VPLS, include the **vstp** statement at the [**edit instance-type layer2-control**] hierarchy level:

```
[edit]
instance-type layer2-control;
protocols {
   vstp {
      interface interface name;
      vlan vlan-id;
   }
}
```

If you want only STP to run on a device, you can configure STP by including the **force-version stp** statement at the [**edit protocols rstp**] or [**edit protocols vstp**] hierarchy level:

```
[edit]
protocols {
   rstp {
      force-version stp;
   }
}
```

For more information about the Spanning Tree Protocol (VSTP, MSTP, RSTP, or STP), see the *MX Series Solutions Guide* and the *Junos OS Routing Protocols Configuration Guide*.

Filtering Layer 2 Packets in a VPLS Instance (MX Series Routers Only)

You can match the **learn-vlan-id**, **user-vlan-id**, and **traffic-type** terms for a VPLS instance on the MX Series platform. Packets entering or exiting the VPLS instance have a single VLAN tag. This VLAN tag is the same as what was received from the network. This VLAN tag corresponds to the one VLAN ID on a singly tagged logical interface or inner VLAN tag for the doubly tagged logical interface. The VLAN ID is used to qualify learned MAC addresses.

To configure a firewall filter for a VPLS instance, specify the conditions that the packet must match at the [**edit firewall family vpls filter** ***filter-name*** **term** ***term-name*** **from**] hierarchy level. To apply a firewall filter to a VPLS routing instance, include the **input filter-name** statement at [**edit routing-instances** ***routing-instance-name*** **forwarding-options family vpls filter**] hierarchy level. For more information, see the *Junos OS Policy Framework Configuration Guide*.

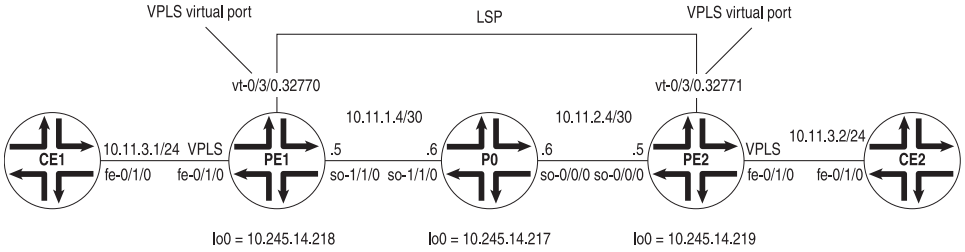
CHAPTER 3

Virtual Private LAN Service Configuration Example

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

Example: VPLS Configuration (BGP Signaling)

Figure 7: VPLS Topology Diagram



VPLS table for PE1

Interface	MAC Address	Outgoing label	Received label
fe-0/1/0	aaaa.aaaa.aaaa	n/a	n/a
vt-0/3/0.32770	bbbb.bbbb.bbbb	800000	800002

g017142

In Figure 7 on page 39, a simple VPLS topology is enabled between routers PE1 and PE2. CE routers CE1 and CE2 use Ethernet-based interfaces to connect VLAN 600 to their local PE router. The PE routers PE1 and PE2 are connected to one another by LSPs enabled across a service provider backbone running MPLS, BGP, RSVP, and OSPF.

In a VPLS routing instance named **green**, PE1 has a local interface **fe-0/1/0** and a virtual port of **vt-0/3/0.32770** (the virtual port is created dynamically on the Tunnel Services PIC when VPLS is configured). PE2 has a local interface **fe-0/1/0** and a virtual port of **vt-0/3/0.32771** in the same **green** instance. As a result, routers CE1 and CE2 send Ethernet traffic to one another as if they were physically connected to each other on a LAN.

On Router CE1, the only item you need to configure is the Fast Ethernet interface that connects to PE1. Be sure to write down the VLAN identifier and IP address, so you can match them later on CE2.

Router CE1

```
[edit]
interfaces {
  fe-0/1/0 {
    vlan-tagging; # Configure VLAN tagging for VLAN VPLS or extended VLAN VPLS.
    unit 0 {
      vlan-id 600; # The Ethernet interface on CE2 must use the same VLAN ID.
      family inet {
        address 10.11.3.1/24; # The interface on CE2 must use the same prefix.
      }
    }
  }
}
```

On Router PE1, prepare the router for VPLS by configuring BGP, MPLS, OSPF, and RSVP. (These protocols are the basis for most Layer 2 VPN-related applications, including VPLS.) Include the **signaling** statement at the **[edit protocols bgp group *group-name* family l2vpn]** hierarchy level, because VPLS uses the same infrastructure for internal BGP as Layer 2 VPNs.

NOTE: In Junos OS Release 7.3 and later, the signaling statement replaces the unicast statement at the [edit protocols bgp group *group-name* family l2vpn] hierarchy level. You must use the signaling statement if you wish to configure VPLS domains and Layer 2 VPNs simultaneously.

Next, configure VLAN tagging on the Fast Ethernet interface connected to Router CE1. Include VLAN VPLS encapsulation at both the physical and logical interface levels. Be sure to use the same VLAN ID for all Ethernet interfaces that are part of a single VPLS instance. Finally, add the Fast Ethernet interface into a VPLS routing instance and specify the site range, site ID number, and site name.

Router PE1

```
[edit]
interfaces {
  fe-0/1/0 {
    vlan-tagging;# Configure VLAN tagging for VLAN VPLS or extended VLAN VPLS.
    encapsulation vlan-vpls; # Configure VPLS encapsulation on both the
    unit 0 { # physical interface and the logical interface.
      encapsulation vlan-vpls;
      vlan-id 600;# The VLAN ID is the same one used by the CE routers.
    }
  }
  so-1/1/0 {
    unit 0 {
      family inet {
        address 10.11.1.5/30;
      }
      family mpls;
    }
  }
  lo0 {
```

```
        unit 0 {
          family inet {
            address 10.245.14.218/32;
          }
        }
      }
    }
    routing-options {
      autonomous-system 69;
      forwarding-table {
        export exp-to-fwd; # Apply a policy that selects an LSP for the VPLS instance.
      }
    }
    protocols {
      rsvp {
        interface all {
          aggregate;
        }
      }
      mpls {
        label-switched-path pe1-to-pe2 { # Configure an LSP to reach other VPLS PEs.
          to 10.245.14.219;
        }
        interface all;
      }
      bgp {
        group vpls-pe {
          type internal;
          local-address 10.245.14.218;
          family l2vpn { # VPLS uses the same infrastructure as Layer 2 VPNs
            signaling; # for internal BGP.
          }
          neighbor 10.245.14.217;
          neighbor 10.245.14.219;
        }
      }
      ospf {
        traffic-engineering;
        area 0.0.0.0 {
          interface so-1/1/0.0 {
            metric 11;
          }
          interface lo0.0 {
            passive;
          }
        }
      }
    }
    policy-options {
      policy-statement exp-to-fwd {
        term a {
          from community grn-com; # Matches the community in the VPLS instance.
          then {
            install-nexthop lsp pe1-to-pe2; # If there are multiple LSPs that exist
            accept; # between VPLS PE routers, this statement sends VPLS traffic
          } # over a specific LSP.
```

```
    }
  }
  community grn-com members target:11111:1; # Adds the instance to a BGP
} # community.
routing-instances {
  green {
    instance-type vpls; # Configure a VPLS routing instance.
    interface fe-0/1/0.0;
    route-distinguisher 10.245.14.218:1;
    vrf-target target:11111:1; # This value is important to the BGP community.
    protocols {
      vpls { # Configure a VPLS site range, site name, and site identifier.
        site-range 10;
        site greenPE1 {
          site-identifier 1;
        }
      }
    }
  }
}
```

On Router P0, configure BGP, MPLS, OSPF, and RSVP to interconnect PE1 and PE2.

Router P0

```
[edit]
interfaces {
  so-0/0/0 {
    unit 0 {
      family inet {
        address 10.11.2.6/30;
      }
      family mpls;
    }
  }
  so-1/1/0 {
    unit 0 {
      family inet {
        address 10.11.1.6/30;
      }
      family mpls;
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.245.14.217/32;
      }
    }
  }
}
protocols {
  rsvp {
    interface all {
      aggregate;
    }
  }
  mpls {
```

```
    interface all;
  }
  bgp {
    group vpls-pe {
      type internal;
      local-address 10.245.14.217;
      family l2vpn { # VPLS uses the same infrastructure as Layer 2 VPNs
        signaling; # for internal BGP.
      }
      neighbor 10.245.14.218;
      neighbor 10.245.14.219;
    }
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface so-1/1/0.0 {
        metric 11;
      }
      interface so-0/0/0.0 {
        metric 15;
      }
      interface lo0.0 {
        passive;
      }
    }
  }
}
```

On Router PE2, configure BGP, MPLS, OSPF, and RSVP to complement the configuration on PE1. Next, configure VLAN tagging on the Fast Ethernet interface connected to Router CE2. Include VLAN VPLS encapsulation at both the physical and logical interface levels. Be sure to use the same VLAN ID for all Ethernet interfaces that are part of a single VPLS instance. Finally, add the Fast Ethernet interface into a VPLS routing instance and specify the site range, site ID number, and site name.

Router PE2

```
[edit]
interfaces {
  fe-0/1/0 {
    vlan-tagging; # Configure VLAN tagging for VLAN VPLS or extended VLAN VPLS.
    encapsulation vlan-vpls; # Configure VPLS encapsulation on both the
    unit 0 { # physical interface and logical interface.
      encapsulation vlan-vpls;
      vlan-id 600;# The VLAN ID is the same one used by the CE routers.
    }
  }
  so-0/0/0 {
    unit 0 {
      family inet {
        address 10.11.2.5/30;
      }
      family mpls;
    }
  }
  lo0 {
    unit 0 {
```

```
        family inet {
          address 10.245.14.219/32;
        }
      }
    }
}
routing-options {
    autonomous-system 69;
    forwarding-table {
      export exp-to-fwd; # Apply a policy that selects an LSP for the VPLS instance.
    }
}
protocols {
    rsvp {
      interface all {
        aggregate;
      }
    }
    mpls {
      label-switched-path pe2-to-pe1 { # Configure an LSP to other VPLS PE routers.
        to 10.245.14.218;
      }
      interface all;
    }
    bgp {
      group vpls-pe {
        type internal;
        local-address 10.245.14.219;
        family l2vpn { # VPLS uses the same infrastructure as Layer 2 VPNs
          signaling; # for internal BGP.
        }
        neighbor 10.245.14.217;
        neighbor 10.245.14.218;
      }
    }
    ospf {
      traffic-engineering;
      area 0.0.0.0 {
        interface so-0/0/0.0 {
          metric 15;
        }
        interface lo0.0 {
          passive;
        }
      }
    }
}
policy-options {
    policy-statement exp-to-fwd {
      term a {
        from community grn-com; # Matches the community with the VPLS instance.
        then {
          install-nexthop lsp pe2-to-pe1; # If there are multiple LSPs that exist
          accept; # between VPLS PE routers, this statement sends VPLS traffic
        } # over a specific LSP.
      }
```

```
    }
    community grn-com members target:11111:1; # This adds the instance into a BGP
      community.
  }
  routing-instances {
    green {
      instance-type vpls; # Configure a VPLS routing instance.
      interface fe-0/1/0.0;
      route-distinguisher 10.245.14.219:1;
      vrf-target target:11111:1; # This value is important for the BGP community.
      protocols {
        vpls { # Configure a VPLS site range, site name, and site identifier.
          site-range 10;
          site greenPE2 {
            site-identifier 2;
          }
        }
      }
    }
  }
```

On Router CE2, complete your VPLS network by configuring the Fast Ethernet interface that connects to PE2. Use the same VLAN identifier and IP address prefix used on Router CE1.

Router CE2

```
[edit]
interfaces {
  fe-0/1/0 {
    vlan-tagging; # Configure VLAN tagging for VLAN VPLS or extended VLAN VPLS.
    unit 0 {
      vlan-id 600; # The Ethernet interface on CE1 must use the same VLAN ID.
      family inet {
        address 10.11.3.2/24; # The interface on CE1 must use the same prefix.
      }
    }
  }
}
```

Verifying Your Work

To verify proper operation of VPLS, use the following commands:

- **clear vpls mac-address instance *instance-name***
- **show interfaces terse**
- **show route forwarding-table family mpls**
- **show route forwarding-table family vpls (destination | extensive | matching | table)**
- **show route instance (detail)**
- **show system statistics vpls**
- **show vpls connections**
- **show vpls statistics**

The following section shows the output of these commands on Router PE1 as a result of the configuration example:

```
user@PE1> show interfaces terse
Interface               Admin Link Proto Local                 Remote
so-1/1/0                up    up
so-1/1/0.0              up    up   inet  10.11.1.5/30
                                  mpls
so-1/1/1                up    up
so-1/1/2                up    up
so-1/1/3                up    up
fe-0/1/0                up    up
fe-0/1/0.0              up    up   vpls   # This is the local Fast Ethernet
# interface.
fe-0/1/1                up    up
fe-0/1/2                up    up
fe-0/1/3                up    up
gr-0/3/0                up    up
ip-0/3/0                up    up
mt-0/3/0                up    up
pd-0/3/0                up    up
pe-0/3/0                up    up
vt-0/3/0                up    up
vt-0/3/0.32770          up    up  # This is the dynamically generated virtual
port.
dsc                     up    up
fxp0                    up    up
fxp0.0                  up    up   inet  192.186.14.218/24
fxp1                    up    up
fxp1.0                  up    up   tnp   4
gre                     up    up
ipip                    up    up
lo0                     up    up
lo0.0                   up    up   inet  10.245.14.218       --> 0/0
                                        127.0.0.1           --> 0/0
                                  inet6 fe80::2a0:a5ff:fe28:13e0
                                        feee::10:245:14:218
lsi                     up    up
mtun                    up    up
pimd                    up    up
pime                    up    up
tap                     up    up


user@PE1> show system statistics vpls
vpls:
        0 total packets received
        0 with size smaller than minimum
        0 with incorrect version number
        0 packets for this host
        0 packets with no logical interface
        0 packets with no family
        0 packets with no route table
        0 packets with no auxiliary table
        0 packets with no corefacing entry
        0 packets with no CE-facing entry
         6 mac route learning requests # This indicates that VPLS is working.
        6 mac routes learnt
        0 mac routes aged
        0 mac routes moved
```

To display VPLS source and destination MAC address accounting information, use the **destination**, **extensive**, **matching**, or **table** option with the **show route forwarding-table family vpls** command. When you analyze the display output, keep in mind the following:

- VPLS MAC address accounting is handled on a per-MAC address basis for each VPLS instance. All information is retrieved from MAC address entries in the MAC address table. VPLS MAC address accounting is performed only on local CE routers.
- The VPLS counters for source and destination MAC addresses increment continuously until the oldest MAC address entries are removed from the memory buffer, either when the entries time out or if the VPLS instance is restarted.

```
user@PE1> show route forwarding-table family vpls extensive

Routing table: green.vpls [Index 2]
VPLS:

Destination:  default
  Route type: dynamic             Route reference: 0
  Flags: sent to PFE
  Next-hop type: flood                 Index: 353      Reference: 1

Destination:  default
  Route type: permanent           Route reference: 0
  Flags: none
  Next-hop type: discard               Index: 298      Reference: 1

Destination:  fe-0/1/0.0
  Route type: dynamic             Route reference: 0
  Flags: sent to PFE
  Next-hop type: flood                 Index: 355      Reference: 1

Destination:  bb:bb:bb:bb:bb:bb/48 # This MAC address belongs to remote
CE2.
  Route type: dynamic             Route reference: 0
  Flags: sent to PFE, prefix load balance
  Next-hop type: indirect              Index: 351      Reference: 4
  Next-hop type: Push 800000, Push 100002(top)
  Next-hop interface: so-1/1/0.0

Destination:  aa:aa:aa:aa:aa:aa/48 # This MAC address belongs to local
CE1.
  Route type: dynamic             Route reference: 0
  Flags: sent to PFE, prefix load balance
  Next-hop type: unicast               Index: 354      Reference: 2
  Next-hop interface: fe-0/1/0.0

user@PE1> show route forwarding-table family vpls

Routing table: green.vpls
VPLS:
Destination        Type RtRef Next hop          Type Index NhRef Netif
default            dynm     0                   flood  353     1
default            perm     0                   dscd   298     1
fe-0/1/0.0         dynm     0                   flood  355     1
bb:bb:bb:bb:bb:bb/48 # This MAC address belongs to remote CE2.
                   dynm     0                   indr   351     4
                                                Push 800000, Push
100002(top)
so-1/1/0.0
```

```
aa:aa:aa:aa:aa:aa/48 # This MAC address belongs to local CE1.
                 dynm     0                  ucst   354     2 fe-0/1/0.0
```

```
user@PE1> show route forwarding-table family mpls

Routing table: mpls
MPLS:
Destination        Type RtRef Next hop          Type Index NhRef Netif
default            perm     0                   dscd    19     1
0                  user     0                   recv    18     3
1                  user     0                   recv    18     3
2                  user     0                   recv    18     3
100000             user     0 10.11.1.6         swap  100001     so-1/1/0.0
800002             user     0                     Pop
vt-0/3/0.32770
vt-0/3/0.32770 (VPLS)
                   user     0                   indr   351     4
                                                Push 800000, Push
100002(top) so-1/1/0.0
```

```
user@PE1> show route instance green detail

green:
  Router ID: 0.0.0.0
  Type: vpls              State: Active
  Interfaces:
    fe-0/1/0.0  # This is the local Fast Ethernet interface.
   vt-0/3/0.32770  # This is the dynamically generated VPLS virtual port.

  Route-distinguisher: 10.245.14.218:1
  Vrf-import: [ __vrf-import-green-internal__ ]
  Vrf-export: [ __vrf-export-green-internal__ ]
  Vrf-import-target: [ target:11111:1 ]
  Vrf-export-target: [ target:11111:1 ]
  Tables:
    green.l2vpn.0          : 2 routes (2 active, 0 holddown, 0 hidden)
```

```
user@PE1> show vpls connections

L2VPN Connections:
Legend for connection status (St)
OR -- out of range              WE -- intf encaps != instance encaps
EI -- encapsulation invalid     Dn -- down
EM -- encapsulation mismatch    VC-Dn -- Virtual circuit down
CM -- control-word mismatch     -> -- only outbound conn is up
CN -- circuit not present       <- -- only inbound  conn is up
OL -- no outgoing label         Up -- operational
NC -- intf encaps not CCC/TCC   XX -- unknown
NP -- interface not present
Legend for interface status
Up -- operational
Dn -- down
Instance: green
Local site: greenPE1 (1)
    connection-site           Type  St     Time last up          # Up
trans
    2                         rmt   Up     Jan 24 06:26:49 2003
  1
      Local interface: vt-0/3/0.32770, Status: Up, Encapsulation: VPLS
```

```
      Remote PE: 10.245.14.219, Negotiated control-word: No
      Incoming label: 800002, Outgoing label: 800000
```

```
user@PE1> show system statistics vpls

vpls:
        0 total packets received
        0 with size smaller than minimum
        0 with incorrect version number
        0 packets for this host
        0 packets with no logical interface
        0 packets with no family
        0 packets with no route table
        0 packets with no auxiliary table
        0 packets with no corefacing entry
        0 packets with no CE-facing entry
        7 mac route learning requests
        7 mac routes learnt
        0 mac routes aged
        0 mac routes moved
```

```
user@PE1> show route instance green detail

green:
  Router ID: 0.0.0.0
  Type: vpls              State: Active
   Interfaces:
    fe-0/1/0.0
    vt-0/3/0.32770
  Route-distinguisher: 10.245.14.218:1
  Vrf-import: [ __vrf-import-green-internal__ ]
  Vrf-export: [ __vrf-export-green-internal__ ]
  Vrf-import-target: [ target:11111:1 ]
  Vrf-export-target: [ target:11111:1 ]
  Tables:
    green.l2vpn.0          : 2 routes (2 active, 0 holddown, 0 hidden)
```

```
user@PE1> show vpls statistics

Layer-2 VPN Statistics:
Instance: green
   Local interface: fe-0/1/0.0, Index: 351
   Remote provider edge router: 10.245.14.219
     Multicast packets:                   363
     Multicast bytes  :                 30956
     Flood packets    :                     0
     Flood bytes      :                     0
   Local interface: vt-0/3/0.32770, Index: 354
   Remote provider edge router: 10.245.14.219
     Multicast packets:                   135
     Multicast bytes  :                 12014
     Flood packets    :                   135
     Flood bytes      :                 12014
```

To clear all MAC address entries for a VPLS instance from the VPLS table, issue the **clear vpls mac-address instance** ***instance-name*** command. Add the **logical-system** ***logical-system-name*** option to clear entries in a VPLS instance within a logical system. Use the ***mac-address*** option to remove individual MAC addresses.

Example: VPLS Configuration (BGP and LDP Interworking)

Figure 8: Topology for VPLS Configuration Example

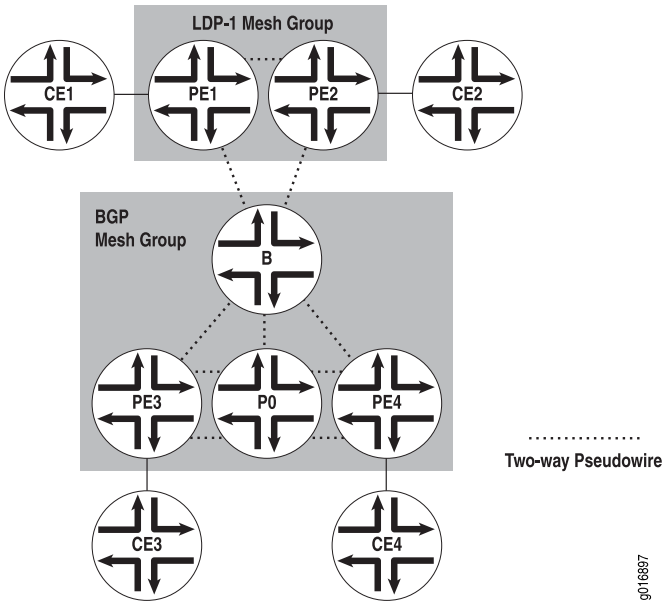


Figure 8 on page 50, shows two VPLS mesh groups: LDP-1 and the default BGP mesh group. The VPLS instance is named **vi** in the configuration. Table 1 on page 50 shows the addresses for the router interfaces in the example topology.

Table 1: Router Interface Addresses for VPLS Configuration Example

Router	Interface	Address
CE1	fe-0/0/3 (link to Router PE1)	10.12.31.1
	loopback	10.12.53.1
CE2	fe-0/0/1 (link to Router PE2)	10.12.31.2
	loopback	10.12.53.2
PE1	t1-1/1/1 (link to Router PE2)	10.12.100.17
	t1-0/1/0 (link to Router B)	10.12.100.2
	loopback	10.255.170.106
PE2	t1-0/1/1 (link to Router PE1)	10.12.100.18
	t1–0/1/3 (link to Router B)	10.12.100.6
	loopback	10.255.170.104

Table 1: Router Interface Addresses for VPLS Configuration Example *(continued)*

Router	Interface	Address
B	t1-0/1/2 (link to Router PE1)	10.12.100.1
	t1-0/1/3 (link to Router PE2)	10.12.100.5
	so-0/2/2 (link to Router PE3)	10.12.100.9
	fe-0/0/3 (link to Router PE4)	10.12.100.13
	loopback	10.255.170.98
PE3	s0-0/2/1 (link to Router B)	10.12.100.10
	so-0/2/2 (link to Router P0)	10.12.100.21
	loopback	10.255.170.96
P0	so-0/2/1 (link to Router PE3)	10.12.100.22
	t1-0/1/3 (link to Router PE4)	10.12.100.25
	loopback	10.255.170.100
PE4	fe-0/0/3 (link to Router B)	10.12.100.14
	t1-0/1/3 (link to Router P0)	10.12.100.26
	loopback	10.255.170.102
CE3	ge-1/2/1 (link to PE3)	10.12.31.3
	loopback	10.12.53.3
CE4	fe-0/0/2 (link to PE4)	10.12.31.4
	loopback	10.12.53.4

On Router CE3, the only item you need to configure is the Gigabit Ethernet interface that connects to PE3.

Router CE3

```
[edit]
interfaces {
   ge-1/2/1 {
      unit 0 {
         family inet {
            address 10.12.31.1/24;
         }
      }
```

```
    }
  }
```

On Router PE3, prepare the router for VPLS by configuring BGP, MPLS, OSPF, and LDP. (These protocols are the basis for most Layer 2 VPN-related applications, including VPLS.) Include the **signaling** statement at the **[edit protocols bgp group *group-name* family l2vpn]** hierarchy level, because VPLS uses the same infrastructure for internal BGP as Layer 2 VPNs.

NOTE: In Junos OS Release 7.3 and later, the signaling statement replaces the unicast statement at the [edit protocols bgp group *group-name* family l2vpn] hierarchy level. You must use the signaling statement if you wish to configure VPLS domains and Layer 2 VPNs simultaneously.

Next, configure VLAN tagging on the Gigabit Ethernet interface connected to Router CE3. Finally, add the Gigabit Ethernet interface into a VPLS routing instance and specify the site range, site ID number, and site name.

Router PE3

```
[edit]
interfaces {
  so-0/2/1 {
    unit 0 {
      family inet {
        address 10.12.100.10/30;
      }
      family mpls;
    }
  so-0/2/2 {
    unit 0 {
      family inet {
        address 10.12.100.21/30;
      }
      family mpls;
    }
  }
    ge-1/3/1 {
      encapsulation ethernet-vpls;
      unit 0 {
        family vpls;
      }
    }
  }
}
protocols {
  mpls {
    interface all;
  }
  bgp {
    log-updown;
    group int {
      type internal;
      local-address 10.255.170.96;
      family l2vpn {
```

```
            signaling;
          }
          neighbor 10.255.170.98;
          neighbor 10.255.170.102;
        }
      }
      ospf {
        area 0.0.0.0 {
          interface so-0/2/1.0;
          interface so-0/2/2.0;
          interface lo0.0 {
            passive;
          }
        }
      }
      ldp {
        interface so-0/2/1.0;
        interface so-0/2/2.0;
      }
    }
    routing-instances {
      v1 {
        instance-type vpls;
        interface ge-1/3/1.0;
        route-distinguisher 10.255.170.96:1;
        vrf-target target:1:2;
        protocols {
          vpls {
            site-range 10;
            site 1 {
              site-identifier 3;
            }
          }
        }
      }
```

On Router P0, configure MPLS, OSPF, and LDP to interconnect PE3 and PE4.

Router P0

```
[edit]
interfaces {
  t1-0/1/3 {
    unit 0 {
      family inet {
        address 10.12.100.25/30;
      }
      family mpls;
    }
  so-0/2/1 {
    unit 0 {
      family inet {
        address 10.12.100.22/30;
      }
      family mpls;
    }
  }
}
```

```
protocols {
    mpls {
        interface all;
    }
    ospf {
        area 0.0.0.0 {
            interface so-0/2/1.0;
            interface t1-0/1/3.0;
            interface lo0.0 {
                passive;
            }
        }
    }
    ldp {
        interface t1-0/1/3.0;
        interface so-0/2/1.0;
    }
}
```

On Router PE4, configure BGP, MPLS, OSPF, and LDP to complement the configuration on PE3. Next, configure VLAN tagging on the Fast Ethernet interface connected to Router CE4. Include VLAN VPLS encapsulation at both the physical and logical interface levels. Finally, add the Fast Ethernet interface into a VPLS routing instance and specify the site range, site ID number, and site name.

Router PE4

```
[edit]
interfaces {
    fe-0/0/2 {
    encapsulation ethernet-vpls;
        unit 0 {
            family vpls;
        }
    fe-0/0/3 {
        unit 0 {
            family inet {
                address 10.12.100.14/30;
            }
            family mpls;
        }
    }
    t1-0/1/3 {
        unit 0 {
            family inet {
                address 10.12.100.26/30;
            }
            family mpls;
        }
    }
}
protocols {
    mpls {
        interface all;
    }
    bgp {
        log-updown;
        group int {
```

```
        type internal;
        local-address 10.255.170.102;
        family l2vpn {
          signaling;
        }
        neighbor 10.255.170.96;
        neighbor 10.255.170.98;
      }
      }
    }
    ospf {
      area 0.0.0.0 {
        interface fe-0/0/3.0;
        interface t1-0/1/3.0;
        interface lo0.0 {
          passive;
        }
      }
    }
    ldp {
      interface fe-0/0/3.0;
      interface t1-0/1/3.0;
      interface lo0.0;
    }
}
```

On Router CE4, configure the Fast Ethernet interface that connects to PE4.

Router CE4

```
[edit]
interfaces {
  fe-0/0/2 {
    unit 0 {
      family inet {
        address 10.12.31.4/24;
      }
    }
  }
}
```

On Router B, the area border router, configure the interfaces. Next, configure BGP, MPLS, OSPF, and LDP. Be sure to include the loopback interface in the LDP configuration by including the **interface lo0.0** statement at the [**edit protocols ldp**] hierarchy level. For BGP, include the **signaling** statement at the [**edit bgp group *group-name* family l2vpn**] hierarchy level. Last, configure the VPLS instance with both BGP and LDP signaling. Configure the LDP-1 mesh group by including the **mesh-group ldp1** statement at the [**edit routing-instances v1 protocols vpls**] hierarchy level.

Router B

```
[edit]
interfaces {
  fe-0/0/3 {
    unit 0 {
      family inet {
        address 10.12.100.13/30;
      }
      family mpls;
    }
```

```
    }
    t1-0/1/2 {
      unit 0 {
        family inet {
          address 10.12.100.1/30;
        }
        family mpls;
      }
    }
    t1-0/1/3 {
      unit 0 {
        family inet {
          address 10.12.100.5/30;
        }
        family mpls;
      }
    }
    so-0/2/2 {
      unit 0 {
        family inet {
          address 10.12.100.9/30;
        }
        family mpls;
      }
    }
}
protocols {
    mpls {
      interface all;
    }
    bgp {
      log-updown;
      group int {
        type internal;
        local-address 10.255.170.98;
        family l2vpn {
          signaling;
        }
        neighbor 10.255.170.96;
        neighbor 10.255.170.102;
      }
    }
    ospf {
      area 0.0.0.0 {
        interface t1-0/1/2.0;
        interface t1-0/1/3.0;
        interface so-0/2/2.0;
        interface fe-0/0/3.0;
        interface lo0.0 {
          passive;
        }
      }
    }
    ldp {
      interface fe-0/0/3.0;
      interface t1-0/1/2.0;
```

```
        interface t1-0/1/3.0;
        interface so-0/2/2.0;
        interface lo0.0;
    }
}
routing-instances {
    v1 {
        instance-type vpls;
        route-distinguisher 10.255.170.98:1;
        vrf-target target:1:2;
        protocols {
            vpls {
                site-range 10;
                site 1 {
                    site-identifier 1;
                }
                vpls-id 101;
                mesh-group ldp-1 {
                    neighbor 10.255.170.106;
                    neighbor 10.255.170.104;
                }
            }
        }
    }
}
```

Finally, configure the LDP PE routers. On Router PE1, prepare the router for VPLS by configuring LDP, MPLS, and OSPF. Next, configure VPLS encapsulation on the Fast Ethernet interface connected to CE1. Finally, add the Fast Ethernet interface to the routing instance, specifying the VPLS ID and the neighboring routers' loopback addresses.

Router PE1

```
[edit]
interfaces {
    fe-0/0/3 {
        encapsulation ethernet-vpls;
        unit 0 {
            family vpls;
        }
    }
    t1-0/1/0 {
        unit 0 {
            family inet {
                address 10.12.100.2/30;
            }
            family mpls;
        }
    }
    t1-1/1/1 {
        unit 0 {
            family inet {
                address 10.12.100.17/30;
            }
            family mpls;
        }
    }
}
```

```
protocols {
    mpls {
        interface all;
    }
    ospf {
        area 0.0.0.0 {
            interface t1-0/1/0.0;
            interface t1-1/1/1.0;
            interface lo0.0 {
                passive;
            }
        }
    }
    ldp {
        interface t1-0/1/0.0;
        interface t1-1/1/1.0;
        interface lo0.0;
    }
}
routing-instances {
    v1 {
        instance-type vpls;
        interface fe-0/0/3.0;
        protocols {
            vpls {
                vpls-id 101;
                neighbor 10.255.170.98;
                neighbor 10.255.170.104;
            }
        }
    }
}
```

Next, configure the Fast Ethernet interface on Router CE1 that connects to Router PE1.

Router CE1

```
[edit]
interfaces {
    fe-0/0/3 {
        unit 0 {
            family inet {
                address 10.12.31.1/24;
            }
        }
    }
}
```

On Router PE2, prepare the router for VPLS by configuring LDP, MPLS, and OSPF. Next, configure VPLS encapsulation on the Fast Ethernet interface connected to Router CE1. Finally, add the Fast Ethernet interface to the routing instance, specifying the VPLS ID and the neighboring routers' loopback addresses.

Router PE2

```
[edit]
interfaces {
    t1-0/1/1 {
        unit 0 {
            family inet {
```

```
            address 10.12.100.18/30;
          }
          family mpls;
        }
      t1-0/1/3 {
        unit 0 {
          family inet {
            address 10.12.100.6/30;
          }
          family mpls;
        }
      }
      fe-1/0/2 {
        encapsulation ethernet-vpls;
        unit 0 {
          family vpls;
        }
      }
    }
    protocols {
      mpls {
        interface all;
      }
      ospf {
        area 0.0.0.0 {
          interface t1-0/1/3.0;
          interface t1-0/1/1.0;
          interface lo0.0 {
            passive;
          }
        }
      }
      ldp {
        interface t1-0/1/1.0;
        interface t1-0/1/3.0;
        interface lo0.0;
      }
    }
    routing-instances {
      v1 {
        instance-type vpls;
        interface fe-1/0/2.0;
        protocols {
          vpls {
            vpls-id 101;
            neighbor 10.255.170.98;
            neighbor 10.255.170.106;
          }
        }
      }
    }
```

Finally, on Router CE2 configure the Fast Ethernet interface connected to PE2:

Router CE2

```
[edit]
interfaces {
```

```
    fe-0/0/1 {
      unit 0 {
        family inet {
          address 10.12.31.2/24;
        }
      }
    }
  }
```

Verifying Your Work

To verify proper operation of VPLS, use the following commands:

- **show bgp summary**
- **show ldp neighbor**
- **show vpls connections**
- **show route forwarding-table family vpls (destination | extensive | matching | table)**
- **show interfaces vt* terse**
- **show vpls flood extensive**
- **show vpls statistics**

The following section shows the output of some of these commands on Router B as a result of the configuration example.

Use the **show bgp summary** command to verify BGP signaling for VPLS is up.

```
user@PB> show bgp summary
Groups: 1 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l2vpn.0            2          2          0          0          0          0
Peer               AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Damped...
10.255.170.96    65000        124        125       0       0       54:26 Establ
  bgp.l2vpn.0: 1/1/0
  v1.l2vpn.0: 1/1/0
10.255.170.102   65000        122        124       0       0       54:18 Establ
  bgp.l2vpn.0: 1/1/0
  v1.l2vpn.0: 1/1/0
```

Use the **show ldp neighbors** command to verify that LDP signaling for VPLS is up.

```
user@B> show ldp neighbors
Address            Interface          Label space ID         Hold time
10.255.170.104     lo0.0              10.255.170.104:0         41
10.255.170.106     lo0.0              10.255.170.106:0         38
10.12.100.14       fe-0/0/3.0         10.255.170.102:0         12
10.12.100.10       so-0/2/2.0         10.255.170.96:0          14
10.12.100.2        t1-0/1/2.0         10.255.170.106:0         14
10.12.100.6        t1-0/1/3.0         10.255.170.104:0         13
```

To verify that the VPLS connections are up, use the **show vpls connections** command.

```
user@B>show vpls connections
Layer-2 VPN connections:
```

```
Legend for connection status (St)
EI -- encapsulation invalid      NC -- interface encapsulation not CCC/TCC/VPLS
EM -- encapsulation mismatch     WE -- interface and instance encaps not same
VC-Dn -- Virtual circuit down    NP -- interface hardware not present
CM -- control-word mismatch      -> -- only outbound connection is up
CN -- circuit not provisioned    <- -- only inbound connection is up
OR -- out of range               Up -- operational
OL -- no outgoing label          Dn -- down
LD -- local site signaled down   CF -- call admission control failure
RD -- remote site signaled down  SC -- local and remote site ID collision LN --
local site not designated  LM -- local site ID not minimum designated RN -- remote
 site not designated RM -- remote site ID not minimum designated XX -- unknown
connection status  IL -- no incoming label
MM -- MTU mismatch               MI -- Mesh-Group ID not availble

Legend for interface status
Up -- operational
Dn -- down

Instance: v1
BGP-VPLS State
  Local site: 1 (1)
    connection-site           Type  St     Time last up          # Up trans
    3                         rmt   Up     Jan 22 16:38:47 2008           1
      Local interface: vt-0/3/0.1048834, Status: Up, Encapsulation: VPLS
        Description: Intf - vpls v1 local site 1 remote site 3
      Remote PE: 10.255.170.96, Negotiated control-word: No
      Incoming label: 800258, Outgoing label: 800000
    4                         rmt   Up     Jan 22 16:38:54 2008           1
      Local interface: vt-0/3/0.1048835, Status: Up, Encapsulation: VPLS
        Description: Intf - vpls v1 local site 1 remote site 4
      Remote PE: 10.255.170.102, Negotiated control-word: No
      Incoming label: 800259, Outgoing label: 800000 LDP-VPLS State
VPLS-id: 101
  Mesh-group connections: m1
    Neighbor                  Type  St     Time last up          # Up trans
    10.255.170.104(vpls-id 101) rmt Up     Jan 22 16:38:40 2008           1
      Local interface: vt-0/3/0.1048833, Status: Up, Encapsulation: ETHERNET
        Description: Intf - vpls v1 neighbor 10.255.170.104 vpls-id 101
      Remote PE: 10.255.170.104, Negotiated control-word: No
      Incoming label: 800001, Outgoing label: 800000
    10.255.170.106(vpls-id 101) rmt Up     Jan 22 16:38:39 2008           1
      Local interface: vt-0/3/0.1048832, Status: Up, Encapsulation: ETHERNET
        Description: Intf - vpls v1 neighbor 10.255.170.106 vpls-id 101
      Remote PE: 10.255.170.106, Negotiated control-word: No
      Incoming label: 800000, Outgoing label: 800000
```

To display VPLS routes (MAC addresses) in the vpls forwarding table, use the **show route forwarding-table family vpls** command.

```
user@B> show route forwarding-table family vpls
Routing table: v1.vpls
VPLS:
Destination        Type RtRef Next hop           Type Index NhRef Netif
default            perm     0                    rjct   540     1
vt-0/3/0.1048832   user     0                    comp   587     3
vt-0/3/0.1048833   user     0                    comp   587     3
vt-0/3/0.1048834   user     0                    comp   589     3
vt-0/3/0.1048835   user     0                    comp   589     3
00:17:cb:c2:10:01/48
                   dynm     0                    indr 262143     4
```

```
                                                Push 800000   580     2
t1-0/1/3.0
00:17:cb:c2:10:02/48
                   dynm     0                   indr 262145     4
                              10.12.100.14      Push 800000   594     2
fe-0/0/3.0
00:17:cb:c2:10:03/48
                   dynm     0                   indr 262142     4
                                                Push 800000   576     2
t1-0/1/2.0
00:17:cb:c2:10:bd/48
                   dynm     0                   indr 262144     4
                                                Push 800000   585     2
so-0/2/2.0
```

To display VPLS source and destination MAC address accounting information, use the **destination**, **extensive**, **matching**, or **table** option with the **show route forwarding-table family vpls** command. When you analyze the display output, keep in mind the following:

- VPLS MAC address accounting is handled on a per-MAC address basis for each VPLS instance. All information is retrieved from MAC address entries in the MAC address table. VPLS MAC address accounting is performed only on local CE routers.
- The VPLS counters for source and destination MAC addresses increment continuously until the oldest MAC address entries are removed from the memory buffer, either when the entries time out or if the VPLS instance is restarted.

To display status information about Virtual Loopback Tunnel interfaces in the VPLS instance, use the **show interfaces vt* terse** command.

```
user@B> show interfaces vt* terse
Interface               Admin Link Proto    Local                 Remote
vt-0/3/0                up    up
vt-0/3/0.1048832        up    up   vpls
vt-0/3/0.1048833        up    up   vpls
vt-0/3/0.1048834        up    up   vpls
vt-0/3/0.1048835        up    up   vpls
```

To display VPLS route information related to the flood process, use the **show vpls flood extensive** command.

```
user@B> show vpls flood extensive
Name: v1
CEs: 0
VEs: 4
  Flood route prefix: 0x4a/32
  Flood route type: IFF_FLOOD
  Flood route owner: vt-0/3/0.1048834
  Flood group name: __ves__
  Flood group index: 0
  Nexthop type: comp
  Nexthop index: 589
    Flooding to:
    Name             Type          NhType          Index
    m1               Group          comp            588
        Composition: flood-to-all
        Flooding to:
        Name             Type          NhType          Index
        vt-0/3/0.1048832 VE             indr            262142
        vt-0/3/0.1048833 VE             indr            262143
```

```
  Flood route prefix: 0x4b/32
  Flood route type: IFF_FLOOD
  Flood route owner: vt-0/3/0.1048835
  Flood group name: __ves__
  Flood group index: 0
  Nexthop type: comp
  Nexthop index: 589
    Flooding to:
    Name             Type          NhType          Index
    m1               Group          comp            588
        Composition: flood-to-all
        Flooding to:
        Name             Type          NhType          Index
        vt-0/3/0.1048832 VE             indr           262142
        vt-0/3/0.1048833 VE             indr           262143

  Flood route prefix: 0x48/32
  Flood route type: IFF_FLOOD
  Flood route owner: vt-0/3/0.1048832
  Flood group name: m1
  Flood group index: 2
  Nexthop type: comp
  Nexthop index: 587
    Flooding to:
    Name             Type          NhType          Index
    __ves__          Group          comp            586
        Composition: flood-to-all
        Flooding to:
        Name             Type          NhType          Index
        vt-0/3/0.1048834 VE             indr           262144
        vt-0/3/0.1048835 VE             indr           262145

  Flood route prefix: 0x49/32
  Flood route type: IFF_FLOOD
  Flood route owner: vt-0/3/0.1048833
  Flood group name: m1
  Flood group index: 2
  Nexthop type: comp
  Nexthop index: 587
    Flooding to:
    Name             Type          NhType          Index
    __ves__          Group          comp            586
        Composition: flood-to-all
        Flooding to:
        Name             Type          NhType          Index
        vt-0/3/0.1048834 VE             indr           262144
        vt-0/3/0.1048835 VE             indr           262145
```

To view packet flow statistics for the VPLS instance, use the **show vpls statistics** command:

```
user@B> show vpls statistics
Instance: v1
   Local interface: vt-0/3/0.1048832, Index: 72
   Remote PE: 10.255.170.106
     Multicast packets:                    6
     Multicast bytes  :                  360
     Flooded packets  :                   16
     Flooded bytes    :                 1188
     Current MAC count:                    1
```

```
Local interface: vt-0/3/0.1048833, Index: 73
Remote PE: 10.255.170.104
  Multicast packets:                     4
  Multicast bytes  :                   240
  Flooded packets  :                     6
  Flooded bytes    :                   398
  Current MAC count:                     1
Local interface: vt-0/3/0.1048834, Index: 74
Remote PE: 10.255.170.96
  Multicast packets:                     2
  Multicast bytes  :                   120
  Flooded packets  :                     4
  Flooded bytes    :                   278
  Current MAC count:                     1
Local interface: vt-0/3/0.1048835, Index: 75
Remote PE: 10.255.170.102
  Multicast packets:                     1
  Multicast bytes  :                    60
  Flooded packets  :                     2
  Flooded bytes    :                   158
  Current MAC count:                     1
```

Example: Configuring Inter-AS VPLS with MAC Processing at the ASBR

This example describes how to configure inter-AS Virtual Private LAN Service (VPLS) with MAC processing between BGP-signaled VPLS and LDP-signaled VPLS. This feature is described in RFC 4761 as multi-AS VPLS option E or method E.

This example is organized in the following sections:

Requirements

To support inter-AS VPLS between BGP-signaled VPLS and LDP-signaled VPLS, your network must meet the following hardware and software requirements:

- MX Series or M320 routers for the ASBRs.
- Junos OS Release 9.3 or higher.
- Gigabit Ethernet or 10-Gigabit Ethernet interfaces.

Overview and Topology

VPLS is a key enabler for delivering multipoint Ethernet service. Major service providers have implemented IP and MPLS backbones and offer VPLS services to large enterprises. Growing demand requires the VPLS network to scale to support many VPLS customers with multiple sites spread across geographically dispersed regions. BGP-signaled VPLS signaling offers scaling advantages over LDP-signaled VPLS. In some environments there is a need for BGP-signaled VPLS to interoperate with existing LDP-signaled VPLS.

This example shows one way to configure BGP-signaled VPLS interworking with an existing LDP-signaled VPLS network.

The advantages of the configuration are:

- You can interconnect customer sites that are spread across different autonomous systems (ASs).
- LDP-signaled VPLS and BGP-signaled VPLS interworking is supported.
- Because the ASBR supports MAC operations, customer sites can be connected directly to the ASBR.
- The inter-AS link is not restricted to Ethernet interfaces.
- Additional configuration for multihoming is relatively straightforward.

Traffic from the interworking virtual private LAN services is switched at the ASBR. The ASBR does all the data plane operations: flooding, MAC learning, aging, and MAC forwarding for each AS to switch traffic among any customer facing interfaces and between the fully meshed pseudowires in the AS. A single pseudowire is created between the ASBRs across the inter-AS link and the ASBRs forward traffic from the pseudowires in each AS to the peer ASBR.

Each ASBR performs VPLS operations within its own AS and performs VPLS operations with the ASBR in the other AS. The ASBR treats the other AS as a BGP-signaled VPLS site. To establish VPLS pseudowires, VPLS NLRI messages are exchanged across the EBGP sessions on the inter-AS links between the ASBRs.

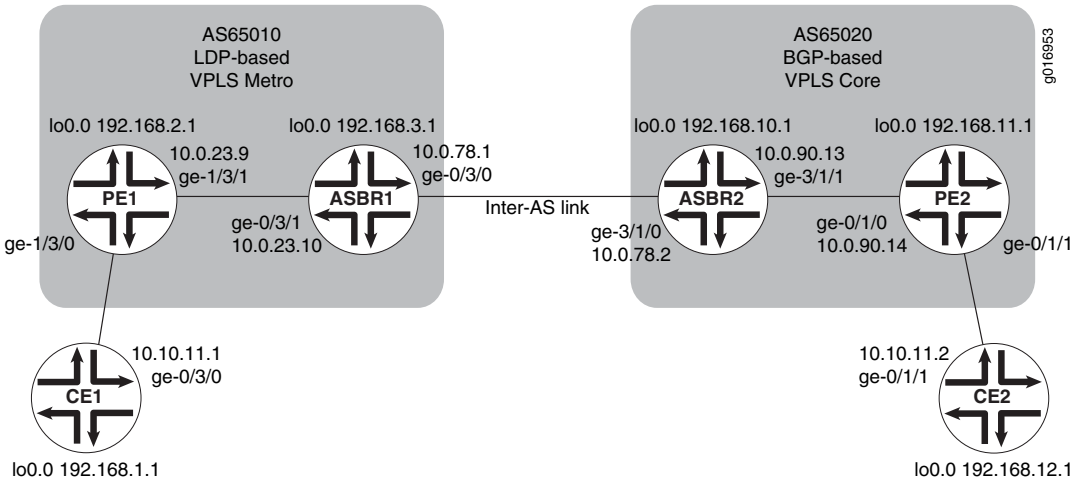
The sample metro network is configured for LDP-signaled VPLS. The core network is configured for BGP-signaled VPLS.

The first part of the example shows the basic configuration steps to configure the logical interfaces, OSPF, internal BGP, LDP, and MPLS. This part of the configuration is the same as other VPLS configurations for LDP-signaled VPLS and BGP-signaled VPLS.

The unique part of the example is configured in the VPLS routing instances, external BGP, and the policy that populates the BGP route table with routes learned from direct routes and OSPF routes. Additional details about the configuration statements are included in the step-by-step procedure.

Figure 9 on page 66 shows the topology used in this example.

Figure 9: Inter-AS VPLS with MAC Operations Example Topology



Configuration

To configure inter-AS VPLS between BGP-signaled VPLS and LDP-signaled VPLS, perform these tasks.

NOTE: In any configuration session it is a good practice to periodically use the commit check command to verify that the configuration can be committed.

- Configuring Interfaces on page 66
- Configuring OSPF on page 68
- Configuring the Internal BGP Peer Group on page 69
- Configuring LDP on page 71
- Configuring MPLS on page 72
- Configuring the External BGP Peer Group Between the Loopback Interfaces on page 72
- Configuring the External BGP Peer Group Between the Inter-AS Link Interfaces on page 73
- Configuring the VPLS Routing Instances on page 77

Configuring Interfaces

Step-by-Step Procedure

To configure interfaces:

1. On each router, configure an IP address on the loopback logical interface 0 (lo0.0):

   user@CE1# set interfaces lo0 unit 0 family inet address 192.168.1.1/32 primary

   user@PE1# set interfaces lo0 unit 0 family inet address 192.168.2.1/32 primary

   user@ASBR1# set interfaces lo0 unit 0 family inet address 192.168.3.1/32 primary

   user@ASBR2# set interfaces lo0 unit 0 family inet address 192.168.10.1/32 primary

```
user@PE2# set interfaces lo0 unit 0 family inet address 192.168.11.1/32 primary

user@CE2# set interfaces lo0 unit 0 family inet address 192.168.12.1/32 primary
```

2. On each router, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

3. On each router, display the interface information for **lo0** and verify that the correct IP address is configured:

```
user@host> show interfaces lo0


Physical interface: lo0, Enabled, Physical link is Up
  Interface index: 6, SNMP ifIndex: 6
  Type: Loopback, MTU: Unlimited
  Device flags   : Present Running Loopback
  Interface flags: SNMP-Traps
  Link flags     : None
  Last flapped   : Never
    Input packets : 0
    Output packets: 0

  Logical interface lo0.0 (Index 75) (SNMP ifIndex 16)
    Flags: SNMP-Traps Encapsulation: Unspecified
    Input packets : 0
    Output packets: 0
    Protocol inet, MTU: Unlimited
      Flags: None
      Addresses
        Local: 127.0.0.1
      Addresses, Flags: Primary Is-Default Is-Primary
        Local: 192.168.3.1
Logical interface lo0.16384 (Index 64) (SNMP ifIndex 21)
    Flags: SNMP-Traps Encapsulation: Unspecified
    Input packets : 0
    Output packets: 0
    Protocol inet, MTU: Unlimited
      Flags: None
      Addresses
        Local: 127.0.0.1

  Logical interface lo0.16385 (Index 65) (SNMP ifIndex 22)
    Flags: SNMP-Traps Encapsulation: Unspecified
    Input packets : 0
    Output packets: 0
    Protocol inet, MTU: Unlimited
      Flags: None
```

In the example above notice that the primary **lo0** local address for the **inet** protocol family on Router ASBR1 is **192:168:3:1**.

4. On each router, configure an IP address and protocol family on the Gigabit Ethernet interfaces. Specify the **inet** protocol family.

```
user@CE1# set interfaces ge-0/3/0 unit 0 family inet address 10.10.11.1/24

user@PE1# set interfaces ge-1/3/1 unit 0 family inet address 10.0.23.9/30

user@ASBR1# set interfaces ge-0/3/1 unit 0 family inet address 10.0.23.10/30
user@ASBR1# set interfaces ge-0/3/0 unit 0 family inet address 10.0.78.1/30

user@ASBR2# set interfaces ge-3/1/0 unit 0 family inet address 10.0.78.2/30
user@ASBR2# set interfaces ge-3/1/1 unit 0 family inet address 10.0.90.13/30

user@PE2# set interfaces ge-0/1/0 unit 0 family inet address 10.0.90.14/30

user@CE2# set interfaces ge-0/1/1 unit 0 family inet address 10.10.11.2/24
```

5. On each router, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

6. Display information for Gigabit Ethernet interfaces and verify that the IP address and protocol family are configured correctly.

```
user@ASBR2> show interfaces ge-* terse

Interface               Admin Link Proto    Local             Remote
ge-3/1/0                up    up
ge-3/1/0.0              up    up   inet   10.0.78.2/30
                                   multiservice
ge-3/1/1                up    up
ge-3/1/1.0              up    up   inet   10.0.90.13/30
                                   multiservice
ge-3/1/2                up    down
ge-3/1/3                up    down
```

Configuring OSPF

Step-by-Step Procedure

To configure OSPF:

1. On the PE and ASBR routers, configure the provider instance of OSPF. Configure OSPF traffic engineering support. Specify area 0.0.0.1 in the LDP-signaled VPLS network and area 0.0.0.0 in the BGP-signaled network. Specify the Gigabit Ethernet logical interfaces between the PE and ASBR routers. Specify **lo0.0** as a passive interface.

```
user@PE1# set protocols ospf traffic-engineering
user@PE1# set protocols ospf area 0.0.0.1 interface ge-1/3/1.0
user@PE1# set protocols ospf area 0.0.0.1 interface lo0.0 passive
```

```
user@ASBR1# set protocols ospf traffic-engineering
user@ASBR1# set protocols ospf area 0.0.0.1 interface ge-0/3/1.0
user@ASBR1# set protocols ospf area 0.0.0.1 interface lo0.0 passive

user@ASBR2# set protocols ospf traffic-engineering
user@ASBR2# set protocols ospf area 0.0.0.0 interface ge-3/1/1.0
user@ASBR2# set protocols ospf area 0.0.0.0 interface lo0.0 passive

user@PE2# set protocols ospf traffic-engineering
user@PE2# set protocols ospf area 0.0.0.0 interface ge-0/1/0.0
user@PE2# set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

2. On each router, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

3. Display OSPF neighbor information and verify that the PE routers form adjacencies with the ASBR router in the same area. Verify that the neighbor state is **Full**.

```
user@host> show ospf neighbor

Address          Interface         State     ID              Pri  Dead
10.0.23.10       ge-1/3/1.0        Full      192.168.3.1     128  31
```

Configuring the Internal BGP Peer Group

Step-by-Step Procedure

The purpose of configuring an internal BGP peer group is to create a full mesh of BGP LSPs among the PE routers in the BGP-signaled AS, including the ASBR routers.

To configure the internal BGP peer group:

1. The purpose of this step is to create a full mesh of IBGP peers between the PE routers, including the ASBR routers, within the BGP-signaled AS.

   On Router ASBR2, configure internal BGP. Specify the BGP type as **internal**. Specify the local address as the local **lo0** IP address.

   Specify the **inet** protocol family. Specify the **labeled-unicast** statement and the **resolve-vpn** option. The **labeled-unicast** statement causes the router to advertise labeled routes out of the IPv4 inet.0 route table and places labeled routes into the inet.0 route table. The **resolve-vpn** option puts labeled routes in the MPLS inet.3 route table The inet.3 route table is used to resolve routes for the PE router located in the other AS.

   Specify the **l2vpn** family to indicate to the router that this is a VPLS. Specify the **signaling** option to configure BGP as the signaling protocol. This enables BGP to carry Layer 2 VPLS NLRI messages for this peer group.

Specify the **lo0** interface IP address of the PE as the neighbor. Configure an autonomous system identifier.

```
user@ASBR2# set protocols bgp group core-ibgp type internal
user@ASBR2# set protocols bgp group core-ibgp local-address 192.168.10.1
user@ASBR2# set protocols bgp group core-ibgp family inet labeled-unicast
  resolve-vpn
user@ASBR2# set protocols bgp group core-ibgp family l2vpn signaling
user@ASBR2# set protocols bgp group core-ibgp neighbor 192.168.11.1
user@ASBR2# set routing-options autonomous-system 0.65020
```

2. On Router PE2, configure internal BGP. Specify the BGP type as **internal**. Specify the local address as the local **lo0** IP address.

   Specify the **l2vpn** family to indicate this is a VPLS. Specify the **signaling** option to configure BGP as the signaling protocol. This enables BGP to carry Layer 2 VPLS NLRI messages.

   Specify the **lo0** interface IP address of Router ASBR2 as the neighbor. Configure an autonomous system identifier.

   ```
   user@PE2# set protocols bgp group core-ibgp type internal
   user@PE2# set protocols bgp group core-ibgp local-address 192.168.11.1
   user@PE2# set protocols bgp group core-ibgp family l2vpn signaling
   user@PE2# set protocols bgp group core-ibgp neighbor 192.168.10.1
   user@PE2# set routing-options autonomous-system 0.65020
   ```

3. On each router, commit the configuration:

   ```
   user@host> commit check

   configuration check succeeds

   user@host> commit

   commit complete
   ```

4. On Router PE2 and Router ASBR2, display BGP neighbor information and verify that the peer connection state is **Established**.

   ```
   user@ASBR2> show bgp neighbor

   Peer: 192.168.11.1+49443 AS 65020 Local: 192.168.10.1+179 AS 65020
     Type: Internal    State: Established    Flags: ImportEval Sync
     Last State: OpenConfirm   Last Event: RecvKeepAlive
     Last Error: None
     Options: Preference LocalAddress AddressFamily Rib-group Refresh
     Address families configured: l2vpn-signaling inet-labeled-unicast
     Local Address: 192.168.10.1 Holdtime: 90 Preference: 170
     Number of flaps: 0
     Peer ID: 192.168.11.1     Local ID: 192.168.10.1     Active Holdtime: 90
     Keepalive Interval: 30         Peer index: 0

   ...
   ```

Configuring LDP

Step-by-Step Procedure

To configure LDP:

1. On the PE and ASBR routers, configure LDP with the Gigabit Ethernet interfaces between the PE and ASBR routers, and between the two ASBR routers. To support LDP-signaled VPLS, additionally configure LDP with the **lo0.0** interface on Router PE1 and Router ASBR1:

```
user@PE1# set protocols ldp interface ge-1/3/1.0
user@PE1# set protocols ldp interface lo0.0

user@ASBR1# set protocols ldp interface ge-0/3/1.0
user@ASBR1# set protocols ldp interface ge-0/3/0.0
user@ASBR1# set protocols ldp interface lo0.0

user@ASBR2# set protocols ldp interface ge-3/1/0.0
user@ASBR2# set protocols ldp interface ge-3/1/1.0

user@PE2# set protocols ldp interface ge-0/1/0.0
```

NOTE: The configuration of LDP signaling between the ASBR routers is not required for Inter-AS VPLS. It is included here for reference only and might be used in LDP environments.

2. On each router, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

3. Display LDP configuration information and verify that the correct interfaces are configured. LDP operation can be verified after MPLS is configured.

```
user@ASBR1> show configuration protocols ldp

interface ge-0/3/0.0;
interface ge-0/3/1.0;
interface lo0.0;
```

The preceding example is from ASBR1.

Configuring MPLS

Step-by-Step Procedure

To configure MPLS:

1. On the PE and ASBR routers, configure MPLS. Enable MPLS on the logical interfaces. Add the Gigabit Ethernet interfaces to the MPLS protocol. This adds entries to the MPLS forwarding table.

```
user@pe1# set protocols mpls interface ge-1/3/1.0
user@pe1# set interfaces ge-1/3/1 unit 0 family mpls

user@ASBR1# set protocols mpls interface ge-0/3/1.0
user@ASBR1# set protocols mpls interface ge-0/3/0.0
user@ASBR1# set interfaces ge-0/3/1 unit 0 family mpls
user@ASBR1# set interfaces ge-0/3/0 unit 0 family mpls

user@ASBR2# set protocols mpls interface ge-3/1/0.0
user@ASBR2# set protocols mpls interface ge-3/1/1.0
user@ASBR2# set interfaces ge-3/1/0 unit 0 family mpls
user@ASBR2# set interfaces ge-3/1/1 unit 0 family mpls

user@pe2# set protocols mpls interface ge-0/1/0.0
user@pe2# set interfaces ge-0/1/0 unit 0 family mpls
```

2. On each router, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

3. On the PE and ASBR routers, display LDP neighbor information and verify that the directly connected LDP neighbors are listed:

```
user@ASBR1> show ldp neighbor

Address            Interface          Label space ID         Hold time
192.168.2.1        lo0.0              192.168.2.1:0            44
10.0.78.2          ge-0/3/0.0         192.168.10.1:0           13
10.0.23.9          ge-0/3/1.0         192.168.2.1:0            11
```

The preceding example is from ASBR1.

Configuring the External BGP Peer Group Between the Loopback Interfaces

Step-by-Step Procedure

To configure the external BGP (EBGP) peer group between the loopback interfaces:

1. On Router ASBR1 and Router PE1, configure an autonomous system identifier:

```
user@PE1# set routing-options autonomous-system 0.65010

user@ASBR1# set routing-options autonomous-system 0.65010
```

2. On Router ASBR1, configure an external BGP peer group for the loopback interfaces. Specify the **external** BGP group type. Include the **multihop** statement. Specify the local address as the local **lo0** IP address. Configure the **l2vpn** family for BGP signaling. Configure the peer AS as the core AS number. Specify the **lo0** IP address of Router ASBR2 as the neighbor.

```
user@ASBR1# set protocols bgp group vpls-core type external
user@ASBR1# set protocols bgp group vpls-core multihop
user@ASBR1# set protocols bgp group vpls-core local-address 192.168.3.1
user@ASBR1# set protocols bgp group vpls-core family l2vpn signaling
user@ASBR1# set protocols bgp group vpls-core peer-as 65020
user@ASBR1# set protocols bgp group vpls-core neighbor 192.168.10.1
```

3. On Router ASBR2, configure an external BGP peer group for the loopback interfaces. Specify the **external** BGP group type. Include the **multihop** statement. The **multihop** statement is needed because the EBGP neighbors are in different ASs. Specify the local address as the local **lo0** IP address. Configure the **l2vpn** family for BGP signaling. Configure the peer AS as the metro AS number. Specify the **lo0** IP address of Router ASBR1 as the neighbor.

```
user@ASBR2# set protocols bgp group vpls-metro type external
user@ASBR2# set protocols bgp group vpls-metro multihop
user@ASBR2# set protocols bgp group vpls-metro local-address 192.168.10.1
user@ASBR2# set protocols bgp group vpls-metro family l2vpn signaling
user@ASBR2# set protocols bgp group vpls-metro peer-as 65010
user@ASBR2# set protocols bgp group vpls-metro neighbor 192.168.3.1
```

4. On each router, commit the configuration:

```
user@host> commit
```

Configuring the External BGP Peer Group Between the Inter-AS Link Interfaces

Step-by-Step Procedure

The purpose of configuring external BGP peer groups between the inter-AS link interfaces is to create a full mesh of BGP LSPs among the ASBR routers. To configure the external BGP peer group between the inter-AS link interfaces:

1. On Router ASBR1, configure a policy to export OSPF and direct routes, including the **lo0** address of the PE routers, into BGP for the establishment of label-switched paths (LSPs):

```
user@ASBR1# set policy-options policy-statement loopback term term1 from
  protocol ospf
user@ASBR1# set policy-options policy-statement loopback term term1 from
  protocol direct
user@ASBR1# set policy-options policy-statement loopback term term1 from
  route-filter 192.168.0.0/16 longer
user@ASBR1# set policy-options policy-statement loopback term term1 then accept
```

2. On Router ASBR1, configure an external BGP peer group for the inter-AS link. Specify the **external** BGP group type. Specify the local inter-AS link IP address as the local address. Configure the **inet** family and include the **labeled-unicast** and **resolve-vpn** statements. The **labeled-unicast** statement advertises labeled routes out of the IPv4 inet.0 route table and places labeled routes into the inet.0 route table. The **resolve-vpn** option stores labeled routes in the MPLS **inet.3** route table.

Include the **export** statement and specify the policy you created. Configure the peer AS as the core AS number. Specify the inter-AS link IP address of Router ASBR2 as the neighbor.

```
user@ASBR1# set protocols bgp group metro-core type external
user@ASBR1# set protocols bgp group metro-core local-address 10.0.78.1
user@ASBR1# set protocols bgp group metro-core family inet labeled-unicast
  resolve-vpn
user@ASBR1# set protocols bgp group metro-core export loopback
user@ASBR1# set protocols bgp group metro-core peer-as 65020
user@ASBR1# set protocols bgp group metro-core neighbor 10.0.78.2
```

3. On Router ASBR2, configure a policy to export OSPF and direct routes, including the **lo0** address, into BGP for the establishment of LSPs:

```
user@ASBR2# set policy-options policy-statement loopback term term1 from
  protocol ospf
user@ASBR2# set policy-options policy-statement loopback term term1 from
  protocol direct
user@ASBR2# set policy-options policy-statement loopback term term1 from
  route-filter 192.168.0.0/16 longer
user@ASBR2# set policy-options policy-statement loopback term term1 then accept
```

4. On Router ASBR2, configure an external BGP peer group for the inter-AS link. Specify the **external** BGP group type. Specify the local inter-AS link IP address as the local address. Configure the **inet** family and include the **labeled-unicast** and **resolve-vpn** statements. Include the **export** statement and specify the policy you created. Configure the peer AS as the core AS number. Specify the inter-AS link IP address of Router ASBR1 as the neighbor.

```
user@ASBR2# set protocols bgp group core-metro type external
user@ASBR2# set protocols bgp group core-metro local-address 10.0.78.2
user@ASBR2# set protocols bgp group core-metro family inet labeled-unicast
  resolve-vpn
user@ASBR2# set protocols bgp group core-metro export loopback
user@ASBR2# set protocols bgp group core-metro peer-as 65010
user@ASBR2# set protocols bgp group core-metro neighbor 10.0.78.1
```

5. On each router, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

6. On Router ASBR1, display the BGP neighbors. Verify that the first peer is the IP address of the Gigabit Ethernet interface of Router ASBR2. Verify that the second peer is the IP address of the **lo0** interface of Router ASBR2. Also verify that the state of each peer is **Established**. Notice that on Router ASBR1 the NLRI advertised by Router ASBR2 the inter-AS link peer is **inet-labeled-unicast** and the NLRI advertised by Router ASBR2 the loopback interface peer is **l2vpn-signaling**.

```
user@ASBR1> show bgp neighbor
```

```
Peer: 10.0.78.2+65473 AS 65020 Local: 10.0.78.1+179 AS 65010
  Type: External    State: Established    Flags: Sync
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: Cease
  Export: [ loopback ]
  Options: Preference LocalAddress AddressFamily PeerAS Rib-group Refresh
  Address families configured: inet-labeled-unicast
  Local Address: 10.0.78.1 Holdtime: 90 Preference: 170
  Number of flaps: 3
  Last flap event: Stop
  Error: 'Cease' Sent: 1 Recv: 2
  Peer ID: 192.168.10.1     Local ID: 192.168.3.1      Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 0
  BFD: disabled, down
  Local Interface: ge-0/3/0.0
  NLRI for restart configured on peer: inet-labeled-unicast
  NLRI advertised by peer: inet-labeled-unicast
  NLRI for this session: inet-labeled-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-labeled-unicast
  NLRI that restart is negotiated for: inet-labeled-unicast
  NLRI of received end-of-rib markers: inet-labeled-unicast
  NLRI of all end-of-rib markers sent: inet-labeled-unicast
  Peer supports 4 byte AS extension (peer-as 65020)
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              2
    Received prefixes:            3
    Accepted prefixes:            3
    Suppressed due to damping:    0
    Advertised prefixes:          3
  Last traffic (seconds): Received 8    Sent 3    Checked 60
  Input messages:  Total 8713   Updates 3       Refreshes 0     Octets 165688
  Output messages: Total 8745   Updates 2       Refreshes 0     Octets 166315
  Output Queue[0]: 0

Peer: 192.168.10.1+51234 AS 65020 Local: 192.168.3.1+179 AS 65010
  Type: External    State: Established    Flags: Sync
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: Cease
  Options: Multihop Preference LocalAddress AddressFamily PeerAS Rib-group Refresh
  Address families configured: l2vpn-signaling
  Local Address: 192.168.3.1 Holdtime: 90 Preference: 170
  Number of flaps: 3
  Last flap event: Stop
  Error: 'Cease' Sent: 1 Recv: 2
  Peer ID: 192.168.10.1     Local ID: 192.168.3.1      Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 0
  BFD: disabled, down
  NLRI for restart configured on peer: l2vpn-signaling
  NLRI advertised by peer: l2vpn-signaling
  NLRI for this session: l2vpn-signaling
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: l2vpn-signaling
```

```
  NLRI that restart is negotiated for: l2vpn-signaling
  NLRI of received end-of-rib markers: l2vpn-signaling
  NLRI of all end-of-rib markers sent: l2vpn-signaling
  Peer supports 4 byte AS extension (peer-as 65020)
  Table bgp.l2vpn.0 Bit: 20000
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
    Send state: in sync
    Active prefixes:              1
    Received prefixes:            1
    Accepted prefixes:            1
    Suppressed due to damping:    0
    Advertised prefixes:          1
  Table inter-as.l2vpn.0
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
    Send state: not advertising
    Active prefixes:              1
    Received prefixes:            1
    Accepted prefixes:            1
    Suppressed due to damping:    0
  Last traffic (seconds): Received 19   Sent 18   Checked 42
  Input messages:  Total 8712   Updates 3       Refreshes 0     Octets 165715
  Output messages: Total 8744   Updates 2       Refreshes 0     Octets 166342
  Output Queue[1]: 0
  Output Queue[2]: 0
```

7. On Router ASBR2, display the BGP summary. Notice that the first peer is the IP address of the Gigabit Ethernet interface of Router ASBR1, the second peer is the IP address of the **lo0** interface of Router ASBR1, and the third peer is the **lo0** interface of Router PE2. Verify that the state of each peer is **Established**.

```
user@ASBR2> show bgp summary

Groups: 3 Peers: 3 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 3          2          0          0          0          0
bgp.l2vpn.0            2          2          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
10.0.78.1             65010       8781       8748       0       2 2d 17:54:56 Establ
  inet.0: 2/3/3/0
192.168.3.1           65010       8780       8747       0       2 2d 17:54:54 Establ
  bgp.l2vpn.0: 1/1/1/0
  inter-as.l2vpn.0: 1/1/1/0
192.168.11.1          65020       8809       8763       0       1 2d 17:59:22 Establ
  bgp.l2vpn.0: 1/1/1/0
  inter-as.l2vpn.0: 1/1/1/0
```

8. On Router PE2, display the BGP group. Verify that the peer is the IP address of the **lo0** interface of Router ASBR2. Verify that the number of established peer sessions is 1.

```
user@PE1> show bgp group

Group Type: Internal    AS: 65020                Local AS: 65020
  Name: core-ibgp       Index: 1                 Flags: Export Eval
  Holdtime: 0
  Total peers: 1        Established: 1
  192.168.10.1+179
```

```
  bgp.l2vpn.0: 1/1/1/0
  inter-as.l2vpn.0: 1/1/1/0

Groups: 1  Peers: 1    External: 0    Internal: 1    Down peers: 0   Flaps: 7
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l2vpn.0            1          1          0          0          0          0
inte.l2vpn.0           1          1          0          0          0          0
```

Configuring the VPLS Routing Instances

Step-by-Step Procedure

To configure the VPLS routing instances:

1. On Router PE1, configure the VPLS routing instance. To enable a VPLS instance, specify the **vpls** instance type. Configure VPLS on the CE-facing Gigabit Ethernet interface. Configure the CE-facing interface to use **ethernet-vpls** encapsulation.

   ```
   user@PE1# set routing-instances metro instance-type vpls
   user@PE1# set routing-instances metro interface ge-1/3/0.0
   ```

2. On Router PE1, configure the VPLS protocol within the routing instance. To uniquely identify the virtual circuit, configure the VPLS identifier. The VPLS identifier uniquely identifies each VPLS in the router. Configure the same VPLS ID on all the routers for a given VPLS.

   Specify the IP address of the **lo0** interface on Router ASBR2 as the neighbor.

   Configure the CE-facing interface to use **ethernet-vpls** encapsulation and the **vpls** protocol family.

   ```
   user@PE1# set routing-instances metro protocols vpls vpls-id 101
   user@PE1# set routing-instances metro protocols vpls neighbor 192.168.3.1
   user@PE1# set interfaces ge-1/3/0 encapsulation ethernet-vpls
   user@PE1# set interfaces ge-1/3/0 unit 0 family vpls
   ```

3. On Router ASBR1, configure the VPLS routing instance. To enable a VPLS instance, specify the **vpls** instance type. Configure a route distinguisher and a VRF target. The **vrf-target** statement causes default VRF import and export policies to be generated that accept and tag routes with the specified target community.

   NOTE: A route distinguisher allows the router to distinguish between two identical IP prefixes used as VPN routes. Configure a different route distinguisher on each ASBR router.

   NOTE: You must configure the same VRF target on both ASBR routers.

   ```
   user@ASBR1# set routing-instances inter-as instance-type vpls
   user@ASBR1# set routing-instances inter-as route-distinguisher 65010:1
   user@ASBR1# set routing-instances inter-as vrf-target target:2:1
   ```

4. On Router ASBR1, configure the VPLS protocol within the routing instance.

Configure the VPLS identifier. Specify the IP address of the **lo0** interface on Router PE1 as the neighbor.

```
user@ASBR1# set routing-instances inter-as protocols vpls vpls-id 101
user@ASBR1# set routing-instances inter-as protocols vpls neighbor 192.168.2.1
```

NOTE: The VPLS identifier uniquely identifies each LDP-signaled VPLS in the router. Configure the same VPLS ID on Router PE1 and Router ASBR1.

5. On Router ASBR1, configure the VPLS site within the routing instance. Configure the site identifier as required by the protocol to establish the EBGP pseudowire. As a best practice for more complex topologies involving multihoming, configure a site preference.

```
user@ASBR1# set routing-instances inter-as protocols vpls site ASBR-metro
  site-identifier 1
user@ASBR1# set routing-instances inter-as protocols vpls site ASBR-metro
  site-preference 10000
```

6. On Router ASBR1, configure the VPLS mesh group **peer-as** statement within the routing instance to specify which ASs belong to this AS mesh group. Configure the peer AS for the mesh group as **all**.

   This statement enables the router to establish a single pseudowire between the ASBR routers. VPLS NLRI messages are exchanged across the EBGP sessions on the inter-AS links between the ASBR routers. All autonomous systems are in one mesh group.

```
user@ASBR1# set routing-instances inter-as protocols vpls mesh-group metro
  peer-as all
```

7. On ASBR2, configure the VPLS routing instance. To enable a VPLS instance, specify the **vpls** instance type. Configure a route distinguisher and a VRF target. The **vrf-target** statement causes default VRF import and export policies to be generated that accept and tag routes with the specified target community.

NOTE: A route distinguisher allows the router to distinguish between two identical IP prefixes used as VPN routes. Configure a different route distinguisher on each ASBR router.

NOTE: You must configure the same VRF target community on both ASBR routers.

```
user@ASBR2# set routing-instances inter-as instance-type vpls
user@ASBR2# set routing-instances inter-as route-distinguisher 65020:1
user@ASBR2# set routing-instances inter-as vrf-target target:2:1
```

8. On Router ASBR2, configure the VPLS site within the routing instance. Configure the site identifier as required by the protocol.

   user@ASBR2# set routing-instances inter-as protocols vpls site ASBR-core site-identifier 2

9. On Router ASBR2, configure the VPLS mesh group within the routing instance to specify which VPLS PEs belong to this AS mesh group. Configure the peer AS for the mesh group as **all**.

   This statement enables the router to establish a single pseudowire between the ASBR routers. VPLS NLRI messages are exchanged across the EBGP sessions on the inter-AS links between the ASBR routers. All autonomous systems are in one mesh group.

   user@ASBR1# set routing-instances inter-as protocols vpls mesh-group core peer-as all

10. On Router PE2, configure the VPLS routing instance. To enable a VPLS instance, specify the **vpls** instance type. Configure VPLS on the CE-facing Gigabit Ethernet interface. Configure a route distinguisher and a VRF target.

    user@PE2# set routing-instances inter-as instance-type vpls
    user@PE2# set routing-instances inter-as interface ge-0/1/1.0
    user@PE2# set routing-instances inter-as route-distinguisher 65020:1
    user@PE2# set routing-instances inter-as vrf-target target:2:1

11. On Router PE2, configure the VPLS site within the routing instance. Configure the site identifier as required by the protocol.

    Configure the CE-facing interface to use **ethernet-vpls** encapsulation and the **vpls** protocol family.

    user@PE2# set routing-instances inter-as protocols vpls site PE2 site-identifier 3
    user@PE2# set interfaces ge-0/1/1 encapsulation ethernet-vpls
    user@PE2# set interfaces ge-0/1/1 unit 0 family vpls

12. On each router, commit the configuration:

    ```
    user@host> commit check

    configuration check succeeds

    user@host> commit

    commit complete
    ```

13. On the PE routers, display the CE-facing Gigabit Ethernet interface information and verify that the encapsulation is configured correctly:

```
user@host> show interfaces ge-1/3/0

Address            Interface          Label space ID         Hold time
10.0.23.10         ge-1/3/1.0         192.168.3.1:0            11

Physical interface: ge-1/3/0, Enabled, Physical link is Up
  Interface index: 147, SNMP ifIndex: 145
  Link-level type: Ethernet, MTU: 1514, Speed: 1000mbps, MAC-REWRITE Error: None,
  Loopback: Disabled, Source filtering: Disabled, Flow control: Enabled,
```

```
Auto-negotiation: Enabled, Remote fault: Online
Device flags   : Present Running
Interface flags: SNMP-Traps Internal: 0x4000
Link flags     : None
CoS queues     : 4 supported, 4 maximum usable queues
Schedulers     : 256
Current address: 00:12:1e:ee:34:db, Hardware address: 00:12:1e:ee:34:db
Last flapped   : 2008-08-27 19:02:52 PDT (5d 22:32 ago)
Input rate     : 0 bps (0 pps)
Output rate    : 0 bps (0 pps)
Ingress rate at Packet Forwarding Engine      : 0 bps (0 pps)
Ingress drop rate at Packet Forwarding Engine : 0 bps (0 pps)
Active alarms  : None
Active defects : None

Logical interface ge-1/3/0.0 (Index 84) (SNMP ifIndex 146)
  Flags: SNMP-Traps Encapsulation: ENET2
  Input packets : 0
  Output packets: 1
  Protocol inet, MTU: 1500
    Flags: None
    Addresses, Flags: Is-Preferred Is-Primary
      Destination: 10.10.11/24, Local: 10.10.11.11, Broadcast: 10.10.11.255
```

Results This section describes commands you can use to test the operation of the VPLS.

1. To verify the VPLS connections have been established, enter the **show vpls connections** command on Router PE 1.

```
user@PE1> show vpls connections

Layer-2 VPN connections:

Legend for connection status (St)
EI -- encapsulation invalid      NC -- interface encapsulation not CCC/TCC/VPLS
EM -- encapsulation mismatch     WE -- interface and instance encaps not same
VC-Dn -- Virtual circuit down    NP -- interface hardware not present
CM -- control-word mismatch      -> -- only outbound connection is up
CN -- circuit not provisioned    <- -- only inbound connection is up
OR -- out of range               Up -- operational
OL -- no outgoing label          Dn -- down
LD -- local site signaled down   CF -- call admission control failure
RD -- remote site signaled down  SC -- local and remote site ID collision
LN -- local site not designated  LM -- local site ID not minimum designated
RN -- remote site not designated RM -- remote site ID not minimum designated
XX -- unknown connection status  IL -- no incoming label
MM -- MTU mismatch               MI -- Mesh-Group ID not availble
BK -- Backup connection          ST -- Standby connection

Legend for interface status
Up -- operational
Dn -- down

Instance: metro
  VPLS-id: 101
    Neighbor                Type  St     Time last up          # Up trans
    192.168.3.1(vpls-id 101)  rmt   Up     Sep  9 14:05:18 2008           1
      Remote PE: 192.168.3.1, Negotiated control-word: No
      Incoming label: 800001, Outgoing label: 800000
      Local interface: vt-1/2/0.1048576, Status: Up, Encapsulation: ETHERNET
        Description: Intf - vpls metro neighbor 192.168.3.1 vpls-id 101
```

In the display from Router PE1, verify that the neighbor is the **lo0** address of Router ASBR1 and that the status is **Up**.

2. To verify the VPLS connections have been established, enter the **show vpls connections** command on Router ASBR 1.

```
user@ASBR1> show vpls connections

...
Instance: inter-as
  BGP-VPLS State
  Mesh-group connections: metro
    Neighbor         Local-site   Remote-site   St      Time last up
    192.168.10.1     1            2             Up      Sep  8 20:16:28 2008
      Incoming label: 800257, Outgoing label: 800000
      Local interface: vt-1/2/0.1049088, Status: Up, Encapsulation: VPLS
  LDP-VPLS State
  VPLS-id: 101
  Mesh-group connections: __ves__
    Neighbor                Type  St     Time last up          # Up trans
    192.168.2.1(vpls-id 101)  rmt   Up     Sep  9 14:05:22 2008           1
      Remote PE: 192.168.2.1, Negotiated control-word: No
      Incoming label: 800000, Outgoing label: 800001
```

```
    Local interface: vt-0/1/0.1049089, Status: Up, Encapsulation: ETHERNET
      Description: Intf - vpls inter-as neighbor 192.168.2.1 vpls-id 101
```

In the display from Router ASBR1, verify that the neighbor is the **lo0** address of Router PE1 and that the status is **Up**.

3. To verify the VPLS connections have been established, enter the **show vpls connections** command on Router ASBR2.

```
user@ASBR2> show vpls connections

...
Instance: inter-as
  BGP-VPLS State
  Mesh-group connections: __ves__
    Neighbor         Local-site   Remote-site   St      Time last up
    192.168.11.1     2            3             Up      Sep 11 15:18:23 2008
      Incoming label: 800002, Outgoing label: 800001
      Local interface: vt-4/0/0.1048839, Status: Up, Encapsulation: VPLS
  Mesh-group connections: core
    Neighbor         Local-site   Remote-site   St      Time last up
    192.168.3.1      2            1             Up      Sep  8 20:16:28 2008
      Incoming label: 800000, Outgoing label: 800257
      Local interface: vt-4/0/0.1048834, Status: Up, Encapsulation: VPLS
```

In the display from Router ASBR2, verify that the neighbor is the **lo0** address of Router PE2 and that the status is **Up**.

4. To verify the VPLS connections have been established, enter the **show vpls connections** command on Router PE2.

```
user@PE2> show vpls connections

...
Instance: inter-as
  Local site: PE2 (3)
    connection-site           Type  St     Time last up          # Up trans
    2                         rmt   Up     Sep  8 20:16:28 2008           1
      Remote PE: 192.168.10.1, Negotiated control-word: No
      Incoming label: 800001, Outgoing label: 800002
      Local interface: vt-0/3/0.1048832, Status: Up, Encapsulation: VPLS
        Description: Intf - vpls inter-as local site 3 remote site 2
```

In the display from Router PE2, verify that the remote PE is the **lo0** address of Router ASBR2 and that the status is **Up**.

5. To verify that the CE routers can send and receive traffic across the VPLS, use the **ping** command.

```
user@CE1> ping 10.10.11.2

PING 10.10.11.2 (10.10.11.2): 56 data bytes
64 bytes from 10.10.11.2: icmp_seq=0 ttl=64 time=1.369 ms
64 bytes from 10.10.11.2: icmp_seq=1 ttl=64 time=1.360 ms
64 bytes from 10.10.11.2: icmp_seq=2 ttl=64 time=1.333 ms
^C


user@CE2> ping 10.10.11.1

PING 10.10.11.1 (10.10.11.1): 56 data bytes
64 bytes from 10.10.11.1: icmp_seq=0 ttl=64 time=6.209 ms
```



```
64 bytes from 10.10.11.1: icmp_seq=1 ttl=64 time=1.347 ms
64 bytes from 10.10.11.1: icmp_seq=2 ttl=64 time=1.324 ms
^C
```

If Router CE1 can send traffic to and receive traffic from Router CE2 and Router CE2 can send traffic to and receive traffic from Router CE1, the VPLS is performing correctly.

6. To display the configuration for Router CE1, use the **show configuration** command.

   For your reference, the relevant sample configuration for Router CE1 follows.

```
interfaces {
  lo0 {
    unit 0 {
      family inet {
        address 192.168.1.1/32 {
          primary;
        }
        address 127.0.0.1/32;
      }
    }
  }
  ge-0/3/0 {
    unit 0 {
      family inet {
        address 10.10.11.1/24;
      }
    }
  }
}
```

7. To display the configuration for Router PE1, use the **show configuration** command.

   For your reference, the relevant sample configuration for Router PE1 follows.

```
interfaces {
  lo0 {
    unit 0 {
      family inet {
        address 192.168.2.1/32 {
          primary;
        }
        address 127.0.0.1/32;
      }
    }
  }
  ge-1/3/0 {
    encapsulation ethernet-vpls;
    unit 0 {
      family vpls;
    }
  }
  ge-1/3/1 {
    unit 0 {
      family inet {
        address 10.0.23.9/30;
      }
      family mpls;
```

```
    }
  }
}
routing-options {
  autonomous-system 0.65010;
}
protocols {
  mpls {
    interface ge-1/3/1.0;
  }
  ospf {
    traffic-engineering;
    area 0.0.0.1 {
      interface ge-1/3/1.0;
      interface lo0.0 {
        passive;
      }
    }
  }
  ldp {
    interface ge-1/3/1.0;
    interface lo0.0;
  }
}
routing-instances {
  metro {
    instance-type vpls;
    interface ge-1/3/0.0;
    protocols {
      vpls {
        vpls-id 101;
        neighbor 192.168.3.1;
      }
    }
  }
}
```

8. To display the configuration for Router ASBR1, use the **show configuration** command.

   For your reference, the relevant sample configuration for Router ASBR1 follows.

```
interfaces {
  lo0 {
    unit 0 {
      family inet {
        address 192.168.3.1/32 {
          primary;
        }
        address 127.0.0.1/32;
      }
    }
  }
  ge-0/3/0 {
    unit 0 {
      family inet {
        address 10.0.78.1/30;
      }
```

```
        family mpls;
      }
    }
    ge-0/3/1 {
      unit 0 {
        family inet {
          address 10.0.23.10/30;
        }
        family mpls;
      }
    }
}
routing-options {
    autonomous-system 0.65010;
}
protocols {
    mpls {
      interface ge-0/3/1.0;
      interface ge-0/3/0.0;
    }
    bgp {
      group vpls-core {
        type external;
        multihop;
        local-address 192.168.3.1;
        family l2vpn {
          signaling;
        }
        peer-as 65020;
        neighbor 192.168.10.1;
      }
      group metro-core {
        type external;
        local-address 10.0.78.1;
        family inet {
          labeled-unicast {
            resolve-vpn;
          }
        }
        export loopback;
        peer-as 65020;
        neighbor 10.0.78.2;
      }
    }
    ospf {
      traffic-engineering;
      area 0.0.0.1 {
        interface ge-0/3/1.0;
        interface lo0.0 {
          passive;
        }
      }
    }
    ldp {
      interface ge-0/3/0.0;
      interface ge-0/3/1.0;
```

```
        interface lo0.0;
    }
}
policy-options {
    policy-statement loopback {
        term term1 {
            from {
                protocol [ ospf direct ];
                inactive: route-filter 10.0.0.0/8 longer;
                route-filter 192.168.0.0/16 longer;
            }
            then accept;
        }
    }
}
routing-instances {
    inter-as {
        instance-type vpls;
        route-distinguisher 65010:1;
        vrf-target target:2:1;
        protocols {
            vpls {
                site ASBR-metro {
                    site-identifier 1;
                    site-preference 10000;
                }
                vpls-id 101;
                neighbor 192.168.2.1;
                mesh-group metro {
                    peer-as {
                        all;
                    }
                }
            }
        }
    }
}
```

9. To display the configuration for Router ASBR2, use the **show configuration** command.

   For your reference, the relevant sample configuration for Router ASBR2 follows.

```
interfaces {
    lo0 {
        unit 0 {
            family inet {
                address 192.168.10.1/32 {
                    primary;
                }
                address 127.0.0.1/32;
            }
        }
    }
    ge-3/1/0 {
        unit 0 {
            family inet {
                address 10.0.78.2/30;
```



```
        }
        family mpls;
      }
    }
    ge-3/1/1 {
      unit 0 {
        family inet {
          address 10.0.90.13/30;
        }
        family mpls;
      }
    }
}
routing-options {
    autonomous-system 0.65020;
}
protocols {
    mpls {
      interface ge-3/1/0.0;
      interface ge-3/1/1.0;
    }
    bgp {
      group core-ibgp {
        type internal;
        local-address 192.168.10.1;
        family inet {
          labeled-unicast {
            resolve-vpn;
          }
        }
        family l2vpn {
          signaling;
        }
        neighbor 192.168.11.1;
      }
      group vpls-metro {
        type external;
        multihop;
        local-address 192.168.10.1;
        family l2vpn {
          signaling;
        }
        peer-as 65010;
        neighbor 192.168.3.1;
      }
      group core-metro {
        type external;
        local-address 10.0.78.2;
        family inet {
          labeled-unicast {
            resolve-vpn;
          }
        }
        export loopback;
        peer-as 65010;
        neighbor 10.0.78.1;
```

```
        }
      }
      ospf {
        traffic-engineering;
        area 0.0.0.0 {
          interface ge-3/1/1.0;
          interface lo0.0 {
            passive;
          }
        }
      }
      ldp {
        interface ge-3/1/0.0;
        interface ge-3/1/1.0;
      }
    }
    policy-options {
      policy-statement loopback {
        term term1 {
          from {
            protocol [ ospf direct ];
            route-filter 192.168.0.0/16 longer;
          }
          then accept;
        }
      }
    }
    routing-instances {
      inter-as {
        instance-type vpls;
        route-distinguisher 65020:1;
        vrf-target target:2:1;
        protocols {
          vpls {
            site ASBR-core {
              site-identifier 2;
            }
            mesh-group core {
              peer-as {
                all;
              }
            }
          }
        }
      }
    }
```

10. To display the configuration for Router PE2, use the **show configuration** command.

    For your reference, the relevant sample configuration for Router PE2 follows.

```
    interfaces {
      lo0 {
        unit 0 {
          family inet {
            address 192.168.11.1/32 {
              primary;
```

```
            }
            address 127.0.0.1/32;
          }
        }
      }
      ge-0/1/0 {
        unit 0 {
          family inet {
            address 10.0.90.14/30;
          }
          family mpls;
        }
      }
      ge-0/1/1 {
        encapsulation ethernet-vpls;
        unit 0 {
          family vpls;
        }
      }
    }
    routing-options {
      autonomous-system 0.65020;
    }
    protocols {
      mpls {
        interface ge-0/1/0.0;
      }
      bgp {
        group core-ibgp {
          type internal;
          local-address 192.168.11.1;
          family l2vpn {
            signaling;
          }
          neighbor 192.168.10.1;
        }
      }
      ospf {
        traffic-engineering;
        area 0.0.0.0 {
          interface ge-0/1/0.0;
          interface lo0.0 {
            passive;
          }
        }
      }
      ldp {
        interface ge-0/1/0.0;
      }
    }
    routing-instances {
      inter-as {
        instance-type vpls;
        interface ge-0/1/1.0;
        route-distinguisher 65020:1;
        vrf-target target:2:1;
```

```
        protocols {
            vpls {
                site PE2 {
                    site-identifier 3;
                }
            }
        }
    }
}
```

11. To display the configuration for Router CE2, use the **show configuration** command.

    For your reference, the relevant sample configuration for Router CE2 follows.

```
interfaces {
    lo0 {
        unit 0 {
            family inet {
                address 192.168.12.1/32 {
                    primary;
                }
                address 127.0.0.1/32;
            }
        }
    }
    ge-0/1/1 {
        unit 0 {
            family inet {
                address 10.10.11.2/24;
            }
        }
    }
}
```

Related Documentation

- Introduction

For More Information

For additional information about VPLS, see the following:

- *Junos VPNs Configuration Guide*
- *Junos Network Interfaces Configuration Guide*
- *Junos Class of Service Configuration Guide*
- *Junos Routing Protocols Configuration Guide*
- *Junos OS Routing Protocols and Policies Command Reference*
- *Junos OS Interfaces Command Reference*
- RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*

- RFC 4762, *Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling*
- RFC 4761, *Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling*

PART 2

Index

Index