VPLS Feature Guide for EX9200 Switches

Release
16.2

Modified: 2016-11-02

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

VPLS Feature Guide for EX9200 Switches
16.2

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

Table of Contents

About the Documentation xiii
Documentation and Release Notes xiii
Supported Platforms xiii
Using the Examples in This Manual xiii
Merging a Full Example xiv
Merging a Snippet xiv
Documentation Conventions xv
Documentation Feedback xvii
Requesting Technical Support xvii
Self-Help Online Tools and Resources xvii
Opening a Case with JTAC xviii

Part 1 Overview

Chapter 1 **Introduction to VPLS and Supported Standards 3**

Introduction to VPLS 3
Supported VPLS Standards 4

Chapter 2 **VPLS Configuration Overview 5**

Introduction to Configuring VPLS 5
Configuring an Ethernet Switch as the CE Device for VPLS 6

Part 2 Configuring VPLS

Chapter 3 **Configuring Signaling Protocols for VPLS 9**

VPLS Routing and Virtual Ports 9
BGP Signaling for VPLS PE Routers Overview 11
Interoperability Between BGP Signaling and LDP Signaling in VPLS 12
LDP-Signaled and BGP-Signaled PE Router Topology 12
Flooding Unknown Packets Across Mesh Groups 14
Unicast Packet Forwarding 14
BGP Route Reflectors for VPLS 14
Configuring Interoperability Between BGP Signaling and LDP Signaling in VPLS 15
LDP BGP Interworking Platform Support 16
Configuring FEC 128 VPLS Mesh Groups for LDP BGP Interworking 16
Configuring FEC 129 VPLS Mesh Groups for LDP BGP Interworking 17
Configuring Switching Between Pseudowires Using VPLS Mesh Groups 17
Configuring Integrated Routing and Bridging Support for LDP BGP Interworking with VPLS 18

Configuring Inter-AS VPLS with MAC Processing at the ASBR 18
Inter-AS VPLS with MAC Operations Configuration Summary 18
Configuring the ASBRs for Inter-AS VPLS 19

Chapter 4 Assigning Routing Instances to VPLS 21

Configuring VPLS Routing Instances 21
Configuring BGP Signaling for VPLS 23
Configuring the VPLS Site Name and Site Identifier 23
Configuring Automatic Site Identifiers for VPLS 24
Configuring the Site Range 25
Configuring the VPLS Site Interfaces 27
Configuring the VPLS Site Preference 27
Configuring LDP Signaling for VPLS 28
Configuring LDP Signaling for the VPLS Routing Instance 29
Configuring LDP Signaling on the Router 30
Configuring VPLS Routing Instance and VPLS Interface Connectivity 31
Configuring the VPLS Encapsulation Type 31
Configuring the MPLS Routing Table to Leak Routes a Nondefault Routing Instance 32
Configuring the VPLS MAC Table Timeout Interval 32
Configuring the Size of the VPLS MAC Address Table 33
Limiting the Number of MAC Addresses Learned from an Interface 34
Removing Addresses from the MAC Address Database 35
Configuring VPLS Fast Reroute Priority 36
Specifying the VT Interfaces Used by VPLS Routing Instances 38
Understanding PIM Snooping for VPLS 38
VPLS Label Blocks Operation 40
Elements of Network Layer Reachability Information 40
Requirements for NLRI Elements 41
How Labels are Used in Label Blocks 41
Label Block Composition 41
Label Blocks in Junos OS 42
VPLS Label Block Structure 42
Configuring the Label Block Size for VPLS 44
PE Router Mesh Groups for VPLS Routing Instances 44

Chapter 5 Associating Interfaces with VPLS 47

Configuring Interfaces for VPLS Routing 47
Configuring the VPLS Interface Name 48
Configuring VPLS Interface Encapsulation 48
Enabling VLAN Tagging 51
Configuring VLAN IDs for Logical Interfaces 51
Enabling VLANs for Hub and Spoke VPLS Networks 52
Configuring Aggregated Ethernet Interfaces for VPLS 52
VPLS and Aggregated Ethernet Interfaces 54
Configuring VPLS Without a Tunnel Services PIC 55

Chapter 6 Configuring Pseudowires 57

VPLS Path Selection Process for PE Routers 57
Configuring Static Pseudowires for VPLS 59

Chapter 7 Configuring Multihoming 61
VPLS Multihoming Overview 61
VPLS Multihoming Reactions to Network Failures 63
BGP and VPLS Path Selection for Multihomed PE Routers 64
Configuring VPLS Multihoming (FEC 128) 66
VPLS Multihomed Site Configuration 67
Specifying an Interface as the Active Interface 68
Configuring Multihoming on the PE Router 68
VPLS Single-Homed Site Configuration 68
Chapter 8 Configuring Point-to-Multipoint LSPs 71
Flooding Unknown Traffic Using Point-to-Multipoint LSPs in VPLS 71
Configuring Static Point-to-Multipoint Flooding LSPs 73
Configuring Dynamic Point-to-Multipoint Flooding LSPs 73
Configuring Dynamic Point-to-Multipoint Flooding LSPs with the Default Template 73
Configuring Dynamic Point-to-Multipoint Flooding LSPs with a Preconfigured Template 74
Mapping VPLS Traffic to Specific LSPs 75
Chapter 9 Configuring BGP Path Selection for Layer 2 VPNs 77
Enabling BGP Path Selection for Layer 2 VPNs and VPLS 78
Chapter 10 Configuring Load Balancing and Performance 81
Configuring VPLS Load Balancing 81
Chapter 11 Configuring Class of Service and Firewall Filters in VPLS 85
Configuring EXP-Based Traffic Classification for VPLS 85
Configuring Firewall Filters and Policers for VPLS 85
Configuring a VPLS Filter 86
Configuring an Interface-Specific Counter for VPLS 87
Configuring an Action for the VPLS Filter 87
Configuring VPLS FTFs 87
Changing Precedence for Spanning-Tree BPDU Packets 88
Applying a VPLS Filter to an Interface 88
Applying a VPLS Filter to a VPLS Routing Instance 88
Configuring a Filter for Flooded Traffic 89
Configuring a VPLS Policer 89
Firewall Filter Match Conditions for VPLS Traffic 90
Chapter 12 Monitoring and Tracing VPLS 103
Tracing VPLS Traffic and Operations 103
Part 3 Configuration Statements and Operational Commands
Chapter 13 Configuration Statements 107
active-interface (VPLS Multihoming) 109
any (VPLS Multihoming) 110
automatic-site-id 111
best-site 112
bfd-liveness-detection (Layer 2 VPN and VPLS) 113

connectivity-type . . . 114
encapsulation (Physical Interface) . . . 115
encapsulation-type (Layer 2 VPNs) . . . 121
family multiservice . . . 123
fast-reroute-priority . . . 126
identifier (VPLS Multihoming for FEC 129) . . . 127
interface (Routing Instances) . . . 128
interface (VPLS Multihoming for FEC 129) . . . 129
interface (VPLS Routing Instances) . . . 130
interface-mac-limit (VPLS) . . . 131
l2vpn-id . . . 132
label-block-size . . . 133
label-switched-path-template (Multicast) . . . 134
local-switching (VPLS) . . . 135
mac-flush . . . 136
mac-table-aging-time . . . 138
mac-table-size . . . 139
mesh-group (Protocols VPLS) . . . 140
multi-homing (VPLS Multihoming for FEC 128) . . . 141
multi-homing (VPLS Multihoming for FEC 129) . . . 142
neighbor (Protocols VPLS) . . . 143
no-tunnel-services . . . 145
peer-active (VPLS Multihoming for FEC 129) . . . 146
peer-as (VPLS) . . . 147
ping-interval . . . 148
preference (Interface-Level Preference for VPLS Multihoming for FEC 129) . . . 149
preference (Site-Level Preference for VPLS Multihoming for FEC 129) . . . 150
primary (VPLS Multihoming) . . . 151
rsvp-te (Routing Instances Provider Tunnel) . . . 152
site (VPLS Multihoming for FEC 128) . . . 153
site (VPLS Multihoming for FEC 129) . . . 154
site-identifier (VPLS) . . . 155
site-preference . . . 156
site-range . . . 157
static (Protocols VPLS) . . . 158
template . . . 159
traceoptions (Protocols VPLS) . . . 160
tunnel-services (Routing Instances VPLS) . . . 162
vlan-id . . . 163
vlan-id-list (Interface in VPLS) . . . 163
vlan-tagging . . . 164
vpls (Interfaces) . . . 165
vpls (Routing Instance) . . . 166
vpls-id . . . 168

Chapter 14 Operational Commands 169

Operational-Mode Commands 169
Overview of Junos OS CLI Operational Mode Commands 169
CLI Command Categories 169
Commonly Used Operational Mode Commands 170
Example: Running Operational Mode Commands on Logical Systems 172
Example: Viewing BGP Trace Files on Logical Systems 173
Example: Configuring System Logging on Logical Systems 178

List of Figures

Part 2 Configuring VPLS

Chapter 3 **Configuring Signaling Protocols for VPLS 9**

Figure 1: Flooding a Packet with an Unknown Destination to All PE Routers in the VPLS Instance 10

Figure 2: BGP and LDP Signaling for a VPLS Routing Instance 13

Chapter 4 **Assigning Routing Instances to VPLS 21**

Figure 3: VPLS Label Block Structure 42

Figure 4: Label Mapping Example 43

Chapter 7 **Configuring Multihoming 61**

Figure 5: CE Device Multihomed to Two PE Routers 61

Chapter 8 **Configuring Point-to-Multipoint LSPs 71**

Figure 6: Flooding Unknown VPLS Traffic Using Ingress Replication 71

Figure 7: Flooding Unknown VPLS Traffic Using a Point-to-Multipoint LSP 71

List of Tables

About the Documentation xiii

Table 1: Notice Icons xv
Table 2: Text and Syntax Conventions xv

Part 2 Configuring VPLS

Chapter 4 Assigning Routing Instances to VPLS 21

Table 3: NLRI Elements 40

Chapter 5 Associating Interfaces with VPLS 47

Table 4: VLAN ID Range by Interface Type 51

Chapter 11 Configuring Class of Service and Firewall Filters in VPLS 85

Table 5: Firewall Filter Match Conditions for VPLS Traffic 91

Part 3 Configuration Statements and Operational Commands

Chapter 14 Operational Commands 169

Table 6: Commonly Used Operational Mode Commands 170

About the Documentation

- Documentation and Release Notes on page xiii
- Supported Platforms on page xiii
- Using the Examples in This Manual on page xiii
- Documentation Conventions on page xv
- Documentation Feedback on page xvii
- Requesting Technical Support on page xvii

Documentation and Release Notes

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

If the information in the latest release notes differs from the information in the documentation, follow the product Release Notes.

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at http://www.juniper.net/books.

Supported Platforms

For the features described in this document, the following platforms are supported:

- EX Series

Using the Examples in This Manual

If you want to use the examples in this manual, you can use the **load merge** or the **load merge relative** command. These commands cause the software to merge the incoming configuration into the current candidate configuration. The example does not become active until you commit the candidate configuration.

If the example configuration contains the top level of the hierarchy (or multiple hierarchies), the example is a *full example*. In this case, use the **load merge** command.

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

Merging a Full Example

To merge a full example, follow these steps:

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

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

   ```
   system {
     scripts {
       commit {
         file ex-script.xsl;
       }
     }
   }
   interfaces {
     fxp0 {
       disable;
       unit 0 {
         family inet {
           address 10.0.0.1/24;
         }
       }
     }
   }
   ```

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

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

Merging a Snippet

To merge a snippet, follow these steps:

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

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

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

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

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

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

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

For more information about the **load** command, see CLI Explorer.

Documentation Conventions

Table 1 on page xv defines notice icons used in this guide.

Table 1: Notice Icons

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

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

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the **configure** command: user@host> **configure**

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

Convention	Description	Examples
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host>` **show chassis alarms** `No alarms currently active`
Italic text like this	• Introduces or emphasizes important new terms. • Identifies guide names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS CLI User Guide* • RFC 1997, *BGP Communities Attribute*
Italic text like this	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	• To configure a stub area, include the **stub** statement at the **[edit protocols ospf area area-id]** hierarchy level. • The console port is labeled **CONSOLE**.
< > (angle brackets)	Encloses optional keywords or variables.	**stub <default-metric** ***metric*****>;**
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	**broadcast | multicast** **(*string1* | *string2* | *string3*)**
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	**rsvp { # Required for dynamic MPLS only**
[ ] (square brackets)	Encloses a variable for which you can substitute one or more values.	**community name members [** ***community-ids*** **]**

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

; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
GUI Conventions		
Bold text like this	Represents graphical user interface (GUI) items you click or select.	• In the Logical Interfaces box, select **All Interfaces**. • To cancel the configuration, click **Cancel**.

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

Convention	Description	Examples
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select **Protocols>Ospf**.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can provide feedback by using either of the following methods:

- Online feedback rating system—On any page of the Juniper Networks TechLibrary site at http://www.juniper.net/techpubs/index.html, simply click the stars to rate the content, and use the pop-up form to provide us with information about your experience. Alternately, you can use the online feedback form at http://www.juniper.net/techpubs/feedback/.
- E-mail—Send your comments to **techpubs-comments@juniper.net**. Include the document or topic name, URL or page number, and software version (if applicable).

Requesting Technical Support

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

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit http://www.juniper.net/support/warranty/.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

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

- Find CSC offerings: http://www.juniper.net/customers/support/
- Search for known bugs: http://www2.juniper.net/kb/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/

- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: http://kb.juniper.net/InfoCenter/
- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

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

Opening a Case with JTAC

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

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

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

PART 1

Overview

- Introduction to VPLS and Supported Standards on page 3
- VPLS Configuration Overview on page 5

CHAPTER 1

Introduction to VPLS and Supported Standards

- Introduction to VPLS on page 3
- Supported VPLS Standards on page 4

Introduction to VPLS

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

NOTE: In ACX Series routers, VPLS configuration is supported only on ACX5048 and ACX5096 routers.

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

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

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

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

Supported VPLS Standards

Junos OS substantially supports the following Internet RFCs and draft, which define standards for virtual private LAN service (VPLS).

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

  FEC 128, FEC 129, control bit **0**, the Ethernet pseudowire type **0x0005**, and the Ethernet tagged mode pseudowire type **0x0004** are supported.
- RFC 6391, *Flow-Aware Transport of Pseudowires over an MPLS Packet Switched Network*
- RFC 6790, *The Use of Entropy Labels in MPLS Forwarding*
- Internet draft draft-kompella-l2vpn-vpls-multihoming, *Multi-homing in BGP-based Virtual Private LAN Service*

Related Documentation

- *Supported Carrier-of-Carriers and Interprovider VPN Standards*
- *Supported VPWS Standards*
- *Supported Layer 2 VPN Standards*
- *Supported Layer 3 VPN Standards*
- *Supported Multicast VPN Standards*
- *Accessing Standards Documents on the Internet*

CHAPTER 2

VPLS Configuration Overview

- Introduction to Configuring VPLS on page 5
- Configuring an Ethernet Switch as the CE Device for VPLS on page 6

Introduction to Configuring VPLS

Virtual private LAN service (VPLS) allows you to provide a point-to-multipoint LAN between a set of sites in a virtual private network (VPN).

NOTE: In ACX Series routers, VPLS configuration is supported only on ACX5048 and ACX5096 routers.

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

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

Each VPLS is configured under a routing instance of type **vpls**. A **vpls** routing instance can transparently carry Ethernet traffic across the service provider's network. As with other routing instances, all logical interfaces belonging to a VPLS routing instance are listed under that instance.

In addition to VPLS routing instance configuration, you must configure MPLS label-switched paths (LSPs) between the PE routers, IBGP sessions between the PE routers, and an interior gateway protocol (IGP) on the PE and provider (P) routers.

By default, VPLS is disabled.

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

Configuring an Ethernet Switch as the CE Device for VPLS

For VPLS configurations, the CE device does not necessarily need to be a router. You can link the PE routers directly to Ethernet switches. However, there are a few configuration issues to be aware of:

- When you configure VPLS routing instances and establish two or more connections between a CE Ethernet switch and a PE router, you must enable the Spanning Tree Protocol (STP) on the switch to prevent loops.
- The Junos OS allows standard Bridge Protocol Data Unit (BPDU) frames to pass through emulated Layer 2 connections, such as those configured with Layer 2 VPNs, Layer 2 circuits, and VPLS instances. However, CE Ethernet switches that generate proprietary BPDU frames might not be able to run STP across Juniper Networks routing platforms configured for these emulated Layer 2 connections.

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

PART 2

Configuring VPLS

- Configuring Signaling Protocols for VPLS on page 9
- Assigning Routing Instances to VPLS on page 21
- Associating Interfaces with VPLS on page 47
- Configuring Pseudowires on page 57
- Configuring Multihoming on page 61
- Configuring Point-to-Multipoint LSPs on page 71
- Configuring BGP Path Selection for Layer 2 VPNs on page 77
- Configuring Load Balancing and Performance on page 81
- Configuring Class of Service and Firewall Filters in VPLS on page 85
- Monitoring and Tracing VPLS on page 103

CHAPTER 3

Configuring Signaling Protocols for VPLS

- VPLS Routing and Virtual Ports on page 9
- BGP Signaling for VPLS PE Routers Overview on page 11
- Interoperability Between BGP Signaling and LDP Signaling in VPLS on page 12
- BGP Route Reflectors for VPLS on page 14
- Configuring Interoperability Between BGP Signaling and LDP Signaling in VPLS on page 15

VPLS Routing and Virtual Ports

Because VPLS carries Ethernet traffic across a service provider network, it must mimic an Ethernet network in some ways. When a PE router configured with a VPLS routing instance receives a packet from a CE device, it first determines whether it has the destination of the VPLS packet in the appropriate routing table. If it does, it forwards the packet to the appropriate PE router or CE device. If it does not, it broadcasts the packet to all other PE routers and CE devices that are members of that VPLS routing instance. In both cases, the CE device receiving the packet must be different from the one sending the packet.

NOTE: In the VPLS documentation, the term *router* is used to refer to any device that provides routing functions.

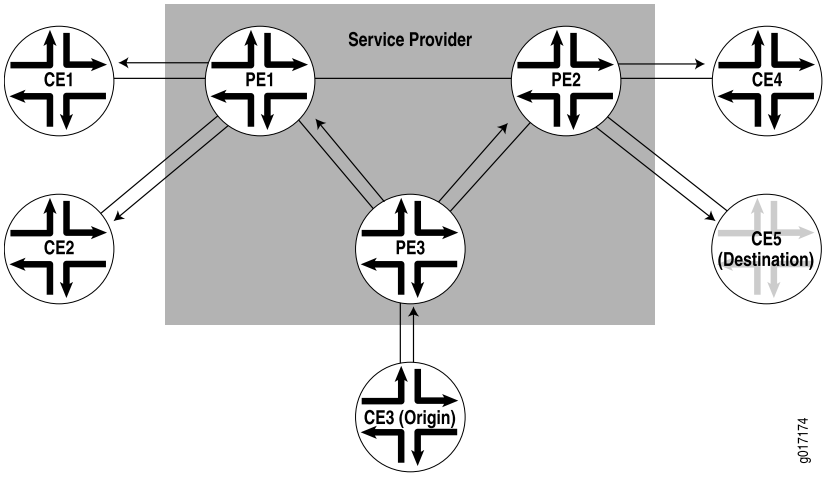
When a PE router receives a packet from another PE router, it first determines whether it has the destination of the VPLS packet in the appropriate routing table. If it does, the PE router either forwards the packet or drops it depending on whether the destination is a local or remote CE device:

- If the destination is a local CE device, the PE router forwards the packet to it.
- If the destination is a remote CE device (connected to another PE router), the PE router discards the packet.

If the PE router cannot determine the destination of the VPLS packet, it floods the packet to all attached CE devices.

This process is illustrated in Figure 1 on page 10.

Figure 1: Flooding a Packet with an Unknown Destination to All PE Routers in the VPLS Instance



VPLS can be directly connected to an Ethernet switch. Layer 2 information gathered by an Ethernet switch (for example, media access control [MAC] addresses and interface ports) is included in the VPLS routing instance table. However, instead of all VPLS interfaces being physical switch ports, the router allows remote traffic for a VPLS instance to be delivered across an MPLS LSP and arrive on a virtual port. The virtual port emulates a local, physical port. Traffic can be learned, forwarded, or flooded to the virtual port in almost the same way as traffic is sent to a local port.

The VPLS routing table learns MAC address and interface information for both physical and virtual ports. The main difference between a physical port and a virtual port is that the router captures additional information from the virtual port, an outgoing MPLS label used to reach the remote site and an incoming MPLS label for VPLS traffic received from the remote site. The virtual port is generated dynamically on a Tunnel Services Physical Interface Card (PIC) when you configure VPLS on the router.

You can also configure VPLS without a Tunnel Services PIC. To do so, you use a label-switched interface (LSI) to provide VPLS functionality. An LSI MPLS label is used as the inner label for VPLS. This label maps to a VPLS routing instance. On the PE router, the LSI label is stripped and then mapped to a logical LSI interface. The Layer 2 Ethernet frame is then forwarded using the LSI interface to the correct VPLS routing instance.

One restriction on flooding behavior in VPLS is that traffic received from remote PE routers is never forwarded to other PE routers. This restriction helps prevent loops in the core network. However, if a CE Ethernet switch has two or more connections to the same PE router, you must enable the Spanning Tree Protocol (STP) on the CE switch to prevent loops. STP is supported on MX Series routers and EX Series switches only.

The Junos OS allows standard Bridge Protocol Data Unit (BPDU) frames to pass through emulated Layer 2 connections, such as those configured with Layer 2 VPNs, Layer 2 circuits, and VPLS routing instances. However, CE Ethernet switches that generate proprietary BPDU frames might not be able to run STP across Juniper Networks routing platforms configured for these emulated Layer 2 connections.

NOTE: Under certain circumstances, VPLS provider routers might duplicate an Internet Control Message Protocol (ICMP) reply from a CE router when a PE router has to flood an ICMP request because the destination MAC address has not yet been learned. The duplicate ICMP reply can be triggered when a CE router with promiscuous mode enabled is connected to a PE router. The PE router automatically floods the promiscuous mode–enabled CE router, which then returns the ICMP request to the VPLS provider routers. The VPLS provider routers consider the ICMP request to be new and flood the request again, creating a duplicate ping reply.

BGP Signaling for VPLS PE Routers Overview

BGP can autonomously signal pseudowires between the PE routers participating in the same virtual private LAN service (VPLS) network. As PE routers are added to and removed from the VPLS network, BGP can signal pseudowires to new PE routers and tear down old pseudowires to old PE routers. Each PE router only needs to be configured with the identity of the VPLS routing instance. Each PE router does not need to be configured with the identities of all of the PE routers that are or might become a part of the VPLS network.

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

When you configure BGP for signaling in a VPLS network, customer sites can be either single-homed to a single PE router or multihomed to two or more PE routers. Multihoming provides redundancy for the connection between the customer site and the service provider's network.

You can either configure all of the PE routers in the VPLS network as a full mesh or you can use BGP route reflectors. For full mesh configurations, each PE router needs to be able to create a bidirectional pseudowire to each of the other PE routers participating in the VPLS network.

Related Documentation

- VPLS Multihoming Overview on page 61
- VPLS Path Selection Process for PE Routers on page 57

Interoperability Between BGP Signaling and LDP Signaling in VPLS

You can configure a VPLS routing instance where some of the PE routers use BGP for signaling and some use LDP for signaling.

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

The following concepts form the basis of the configuration needed to include both BGP-signaled and LDP-signaled PE routers in a VPLS routing instance:

- PE router mesh group—Consists of a set of routers participating in a VPLS routing instance that share the same signaling protocol, either BGP or LDP, and are also fully meshed. Each VPLS routing instance can have just one BGP mesh group. However, you can configure multiple LDP mesh groups for each routing instance.
- Border router—A PE router that must be reachable by all of the other PE routers participating in a VPLS routing instance, whether they are LDP-signaled or BGP-signaled. Bidirectional pseudowires are created between the border router and all of these PE routers. The border router is aware of the composition of each PE mesh group configured as a part of the VPLS routing instance. It can also have direct connections to local CE routers, allowing it to act as a typical PE router in a VPLS routing instance.

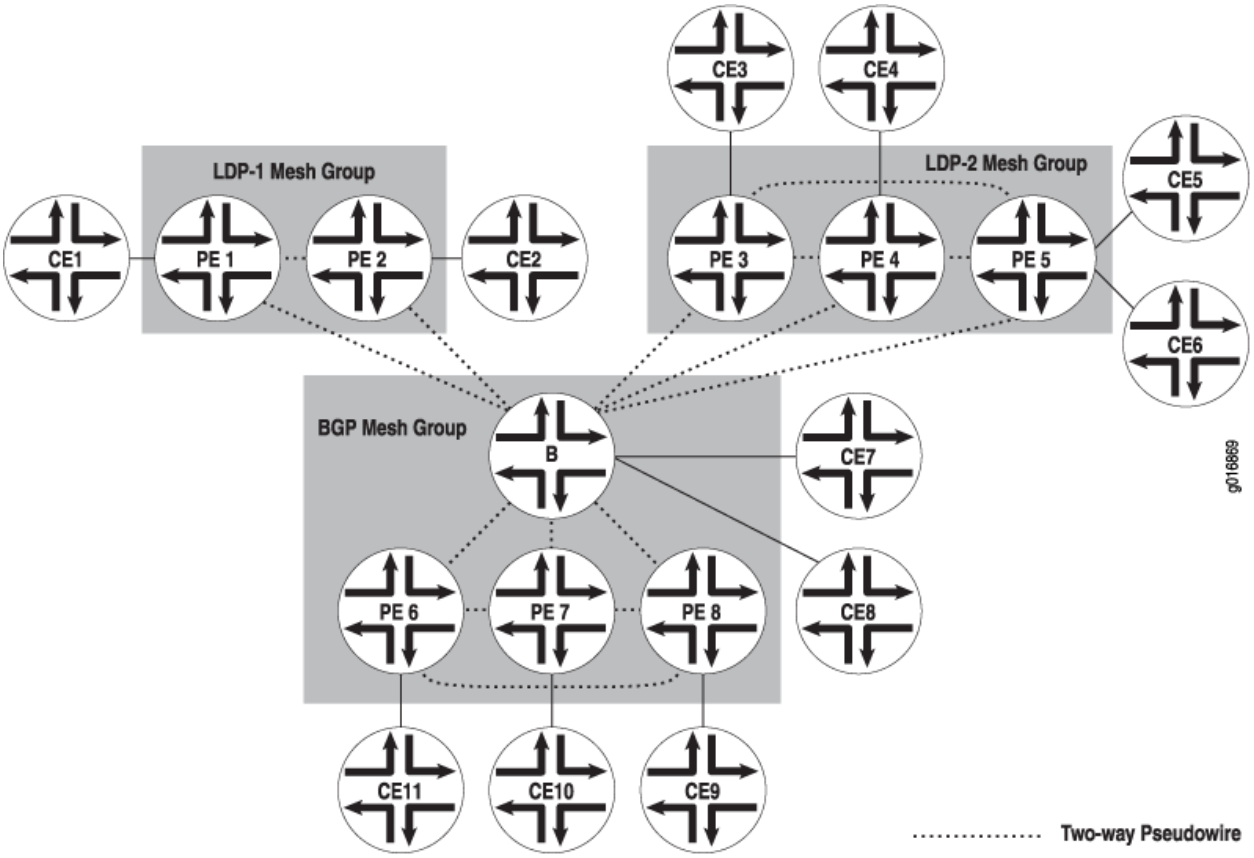
The following sections describe how the LDP-signaled and BGP-signaled PE routers function when configured to interoperate within a VPLS routing instance:

- LDP-Signaled and BGP-Signaled PE Router Topology on page 12
- Flooding Unknown Packets Across Mesh Groups on page 14
- Unicast Packet Forwarding on page 14

LDP-Signaled and BGP-Signaled PE Router Topology

Figure 2 on page 13 illustrates a topology for a VPLS routing instance configured to support both BGP and LDP signaling. Router B is the border router. Routers PE1 and PE2 are in the LDP-signaled mesh group LDP-1. Routers PE3, PE4, and PE5 are in the LDP-signaled mesh group LDP-2. Routers PE6, PE7, PE8, and router B (the border router) are in the BGP-signaled mesh group. The border router also acts as a standard VPLS PE router (having local connections to CE routers). All of the PE routers shown are within the same VPLS routing instance.

Figure 2: BGP and LDP Signaling for a VPLS Routing Instance



Two-way pseudowires are established between the PE routers in each mesh group and between each PE router in the VPLS routing instance and the border router. In Figure 2 on page 13, two-way pseudowires are established between routers PE1 and PE2 in mesh group LDP-1, routers PE3, PE4, and PE5 in mesh group LDP-2, and routers PE6, PE7, and PE8 in the BGP mesh group. Routers PE1 through PE8 also all have two-way pseudowires to the Border router. Based on this topology, the LDP-signaled routers are able to interoperate with the BGP-signaled routers. Both the LDP-signaled and BGP-signaled PE routers can logically function within a single VPLS routing instance.

NOTE: The following features are not supported for VPLS routing instances configured with both BGP and LDP signaling:

- Point-to-multipoint LSPs
- Integrated routing and bridging
- IGMP snooping

Flooding Unknown Packets Across Mesh Groups

Broadcast, multicast, and unicast packets of unknown origin received from a PE router are flooded to all local CE routers. They are also flooded to all of the PE routers in the VPLS routing instance except the PE routers that are a part of the originating PE router mesh group.

For example, if a multicast packet is received by the border router in Figure 2 on page 13, it is flooded to the two local CE routers. It is also flooded to routers PE1 and PE2 in the LDP-1 mesh group and to routers PE3, PE4, and PE5 in the LDP-2 mesh group. However, the packet is not flooded to routers PE6, PE7, and PE8 in the BGP mesh group.

Unicast Packet Forwarding

The PE border router is made aware of the composition of each PE router mesh group. From the data plane, each PE router mesh group is viewed as a virtual pseudowire LAN. The border router is configured to interconnect all of the PE router mesh groups belonging to a single VPLS routing instance. To interconnect the mesh groups, a common MAC table is created on the border router.

Unicast packets originating within a mesh group are dropped if the destination is another PE router within the same mesh group. However, if the destination MAC address of the unicast packet is a PE router located in a different mesh group, the packet is forwarded to that PE router.

BGP Route Reflectors for VPLS

In large networks, it might be necessary to configure BGP route reflectors to reduce the control plane workload for the routers participating in the VPLS network. BGP route reflectors can help to reduce the workload of the network control plane in the following ways.

- Making it unnecessary to configure all of the VPLS PE routers in a full mesh.
- Limiting the total volume of BGP VPLS messages exchanged within the network by transmitting messages to interested routers only (instead of all of the BGP routers in the network)
- Reducing the network signaling load whenever another BGP router is added to or removed from the network

The basic solution to these problems is to deploy a small group of BGP route reflectors that are in a full mesh with one another. Each of the VPLS PE routers is configured to have a BGP session with one or more of the route reflectors, making it unnecessary to maintain a full mesh of BGP sessions between all of the PE routers.

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

This type of configuration only affects the control plane of the VPLS network (how routers signal and tear down pseudowires to one another in the network). The actual data plane state and forwarding paths for the VPLS traffic are not modified by the route reflectors. Effectively, the VPLS pseudowires should take the same paths across the network whether or not you have configured route reflectors. For a description of how VPLS selects the best path to a PE router, see "VPLS Path Selection Process for PE Routers" on page 57.

The MAC addresses themselves are not exchanged or processed in any way by BGP. Each VPLS PE router performs all MAC address learning and aging individually. BGP's only function relative to VPLS is to exchange messages related to automatic discovery of PE routers being added to and removed from the VPLS network and the MPLS label exchange needed to signal a pseudowire from one PE router to another.

Related Documentation

- VPLS Path Selection Process for PE Routers on page 57
- *Example: Configuring a Route Reflector*
- *Example: NG-VPLS Using Point-to-Multipoint LSPs*
- *Example: Next-Generation VPLS for Multicast with Multihoming*

Configuring Interoperability Between BGP Signaling and LDP Signaling in VPLS

A single VPLS routing instance can encompass one set of PE routers that use BGP for signaling and another set of PE routers that use LDP for signaling. Within each set, all of the PE routers are fully meshed in both the control and data planes and have a bidirectional pseudowire to each of the other routers in the set. However, the BGP-signaled routers cannot be directly connected to the LDP-signaled routers. To be able to manage the two separate sets of PE routers in a single VPLS routing instance, a border PE router must be configured to interconnect the two sets of routers.

The VPLS RFCs and Internet drafts require that all of the PE routers participating in a single VPLS routing instance must be fully meshed in the data plane. In the control plane, each fully meshed set of PE routers in a VPLS routing instance is called a PE router mesh group. The border PE router must be reachable by and have bidirectional pseudowires to all of the PE routers that are a part of the VPLS routing instance, both the LDP-signaled and BGP-signaled routers.

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

For LDP BGP interworking to function, LDP-signaled routers can be configured with forwarding equivalence class (FEC) 128 or FEC 129.

The following sections describe how to configure BGP LDP interworking for VPLS:

- LDP BGP Interworking Platform Support on page 16
- Configuring FEC 128 VPLS Mesh Groups for LDP BGP Interworking on page 16
- Configuring FEC 129 VPLS Mesh Groups for LDP BGP Interworking on page 17

- Configuring Switching Between Pseudowires Using VPLS Mesh Groups on page 17
- Configuring Integrated Routing and Bridging Support for LDP BGP Interworking with VPLS on page 18
- Configuring Inter-AS VPLS with MAC Processing at the ASBR on page 18

LDP BGP Interworking Platform Support

LDP BGP interworking is supported on the following Juniper Networks routers and routing platforms:

- M7i
- M10i
- M40e
- M120
- M320
- MX Series routers
- T Series routers
- TX Matrix routers
- EX Series switches

Configuring FEC 128 VPLS Mesh Groups for LDP BGP Interworking

To configure FEC 128 LDP BGP interworking for VPLS, include the **mesh-group** statement in the VPLS routing instance configuration of the PE border router:

```
mesh-group mesh-group-name {
    local-switching;
    mac-flush [ explicit-mac-flush-message-options ];
    neighbor address;
    peer-as all;
    vpls-id number;
}
```

You can include this statement at the following hierarchy levels:

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

Using the **neighbor** statement, configure each PE router that is a part of the mesh group. You must separate the LDP-signaled routers and the BGP-signaled routers into their own respective mesh groups. The LDP-signaled routers can be divided into multiple mesh groups. The BGP-signaled routers must be configured within a single mesh group for each routing instance.

Configuring FEC 129 VPLS Mesh Groups for LDP BGP Interworking

Configuration for a mesh group for FEC 129 is very similiar to the configuration for FEC 128.

Note the following differences for FEC 129:

- Each user-defined mesh group must have a unique route distinguisher. Do not use the route distinguisher that is defined for the default mesh group at the **[edit routing-intances]** hierarchy level.
- Each user-defined mesh group must have its own import and export route target.
- Each user-defined mesh group can have a unique Layer 2 VPN ID. By default, all the mesh groups that are configured for the a VPLS routing-instance use the same Layer 2 VPN ID, the one that you configure at the **[edit routing-instances]** hierarchy level.

Configuring Switching Between Pseudowires Using VPLS Mesh Groups

To configure switching between Layer 2 circuit pseudowires using VPLS mesh groups, you can do either of the following:

- Configure a mesh group for each Layer 2 circuit pseudowire terminating at a VPLS routing instance. The Junos OS can support up to 16 mesh groups on MX Series routers and up to 128 on M Series and T Series routers. However, two mesh groups are created by default, one for the CE routers and one for the PE routers. Therefore, the maximum number of user-defined mesh groups is 14 for MX Series routers and 126 for M Series and T Series routers.
- Configure a single mesh group, terminate all the Layer 2 circuit pseudowires into it, and enable local switching between the pseudowires by including the **local-switching** statement at the **[edit routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name*]** hierarchy level. By default, you cannot configure local switching for mesh groups (except for the CE mesh group) because all of the VPLS PE routers must be configured in a full mesh. However, local switching is useful if you are terminating Layer 2 circuit pseudowires in a mesh group configured for an LDP signaled VPLS routing instance.

NOTE: **Do not include the local-switching statement on PE routers configured in a full mesh VPLS network.**

To terminate multiple pseudowires at a single VPLS mesh group, include the **local-switching** statement:

```
local-switching;
```

You can include this statement at the following hierarchy levels:

- **[edit routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name*]**

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

Configuring Integrated Routing and Bridging Support for LDP BGP Interworking with VPLS

Beginning with Junos OS Release 9.4, you can configure an integrated routing and bridging (IRB) interface on a router that functions as an autonomous system border router (ASBR) in an inter-AS VPLS environment between BGP-signaled VPLS and LDP-signaled VPLS. Previously, IRB interfaces were supported only on Provider Edge (PE) routers.

To configure a IRB support for LDP BGP Interworking with VPLS, include the **routing-interface *interface-name*** statement.

You can include this statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-routers *logical-router-name* routing-instances *routing-instance-name*]

Configuring Inter-AS VPLS with MAC Processing at the ASBR

Inter-AS VPLS with MAC processing at the ASBR enables you to interconnect customer sites that are located in different ASs. In addition, you can configure the ASs with different signaling protocols. You can configure one of the ASs with BGP-signaled VPLS and the other with LDP-signaled VPLS. For more information about how to configure LDP-signaled and BGP signaled VPLS, see "Configuring Interoperability Between BGP Signaling and LDP Signaling in VPLS" on page 15.

For inter-AS VPLS to function properly, you need to configure IBGP peering between the PE routers, including the ASBRs in each AS, just as you do for a typical VPLS configuration. You also need to configure EBGP peering between the ASBRs in the separate ASs. The EBGP peering is needed between the ASBRs only. The link between the ASBR routers does not have to be Ethernet. You can also connect a CE router directly to one of the ASBRs, meaning you do not have to have a PE router between the ASBR and the CE router.

The configuration for the connection between the ASBRs makes inter-AS VPLS with MAC operations unique. The other elements of the configuration are described in other sections of this manual. An extensive configuration example for inter-AS VPLS with MAC operations is provided in the *Junos OS, Release 16.1*.

The following sections describe how to configure inter-AS VPLS with MAC operations:

- Inter-AS VPLS with MAC Operations Configuration Summary on page 18
- Configuring the ASBRs for Inter-AS VPLS on page 19

Inter-AS VPLS with MAC Operations Configuration Summary

This section provides a summary of all of the elements which must be configured to enable inter-AS VPLS with MAC operations. These procedures are described in detail later in this chapter and in other parts of the*Junos OS VPNs Library for Routing Devices*.

The following lists all of major elements of an inter-AS VPLS with MAC operations configuration:

- Configure IBGP between all of the routers within each AS, including the ASBRs.
- Configure EBGP between the ASBRs in the separated ASs. The EBGP configuration includes the configuration that interconnects the ASs.
- Configure a full mesh of LSPs between the ASBRs.
- Configure a VPLS routing instance encompassing the ASBR routers. The ASBRs are VPLS peers and are linked by a single pseudowire. Multihoming between ASs is not supported. A full mesh of pseudowires is needed between the ASBR routers in all of the interconnected ASs.
- Configure the VPLS routing instances using either BGP signaling or LDP signaling. LDP BGP interworking is supported for inter-AS VPLS with MAC operations, so it is possible to interconnect the BGP-signaled VPLS routing instances with the LDP-signaled VPLS routing instances.
- Configure a single VPLS mesh group for all of the ASBRs interconnected using inter-AS VPLS.

Configuring the ASBRs for Inter-AS VPLS

This section describes the configuration on the ASBRs needed to enable inter-AS VPLS with MAC operations.

On each ASBR, you need to configure a VPLS mesh group within the VPLS routing instance which needs to include all of the PE routers within the AS, in addition to the ASBR. You need to configure the same mesh group for each of the ASs you want to interconnect using inter-AS VPLS. The mesh group name should be identical on each AS. You also must include the **peer-as all** statement. This statement enables the router to establish a single pseudowire to each of the other ASBRs.

To configure the mesh group on each ASBR, include the **mesh-group** and **peer-as all** statements:

```
mesh-group mesh-group-name {
  peer-as all;
}
```

You can include these statements at the following hierarchy levels:

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

Related Documentation

- *Example: Configuring BGP Autodiscovery for LDP VPLS*
- *Example: Configuring BGP Autodiscovery for LDP VPLS with User-Defined Mesh Groups*

CHAPTER 4

Assigning Routing Instances to VPLS

- Configuring VPLS Routing Instances on page 21
- Configuring VPLS Fast Reroute Priority on page 36
- Specifying the VT Interfaces Used by VPLS Routing Instances on page 38
- Understanding PIM Snooping for VPLS on page 38
- VPLS Label Blocks Operation on page 40
- Configuring the Label Block Size for VPLS on page 44
- PE Router Mesh Groups for VPLS Routing Instances on page 44

Configuring VPLS Routing Instances

To configure a VPLS routing instance, include the **vpls** statement:

```
vpls {
  active-interface {
    any;
    primary interface-name;
  }
  connectivity-type (ce | irb | permanent);
  control-word;
  encapsulation-type encapsulation-type;
  interface-mac-limit limit;
  import-labeled-routes [ routing-instance-name ];
  label-block-size size;
  mac-table-aging-time time;
  mac-table-size size;
  neighbor neighbor-id;
  no-control-word;
  no-tunnel-services;
  site site-name {
    active-interface {
      any;
      primary interface-name;
    }
    interface interface-name {
      interface-mac-limit limit;
    }
    mesh-group mesh-group-name;
    multi-homing;
```

```
    site-identifier identifier;
    site-preference preference-value {
      backup;
      primary;
    }
  }
  site-range number;
  traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
  }
  tunnel-services {
    devices device-names;
    primary primary-device-name;
  }
  vpls-id vpls-id;
}
```

You can include this statement at the following hierarchy levels:

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

NOTE: ACX Series routers do not support the **[edit logical-systems]** hierarchy.

NOTE: You cannot configure a routing protocol (OSPF, RIP, IS-IS or BGP) inside a VPLS routing instance (**instance-type vpls**). The Junos CLI disallows this configuration.

The configuration for the VPLS routing instance statements is explained in the following sections:

- Configuring BGP Signaling for VPLS on page 23
- Configuring LDP Signaling for VPLS on page 28
- Configuring VPLS Routing Instance and VPLS Interface Connectivity on page 31
- Configuring the VPLS Encapsulation Type on page 31
- Configuring the MPLS Routing Table to Leak Routes a Nondefault Routing Instance on page 32
- Configuring the VPLS MAC Table Timeout Interval on page 32
- Configuring the Size of the VPLS MAC Address Table on page 33
- Limiting the Number of MAC Addresses Learned from an Interface on page 34
- Removing Addresses from the MAC Address Database on page 35

Configuring BGP Signaling for VPLS

You can configure BGP signaling for the VPLS routing instance. BGP is used to signal the pseudowires linking each of the PE routers participating in the VPLS routing instance. The pseudowires carry VPLS traffic across the service provider's network between the VPLS sites.

NOTE: You cannot configure both BGP signaling and LDP signaling for the same VPLS routing instance. If you attempt to configure the statements that enable BGP signaling for the VPLS routing instance (the **site**, **site-identifier**, and **site-range** statements) and the statements that enable LDP signaling for the same instance (the **neighbor** and **vpls-id** statements), the commit operation fails.

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

Configure BGP signaling for the VPLS routing instance by completing the steps in the following sections:

- Configuring the VPLS Site Name and Site Identifier on page 23
- Configuring Automatic Site Identifiers for VPLS on page 24
- Configuring the Site Range on page 25
- Configuring the VPLS Site Interfaces on page 27
- Configuring the VPLS Site Preference on page 27

Configuring the VPLS Site Name and Site Identifier

When you configure BGP signaling for the VPLS routing instance, on each PE router you must configure each VPLS site that has a connection to the PE router. All the Layer 2 circuits provisioned for a VPLS site are listed as the set of logical interfaces (using the **interface** statement) within the **site** statement.

You must configure a site name and site identifier for each VPLS site.

To configure the site name and the site identifier, include the **site** and the **site-identifier** statements:

```
site site-name {
    interface interface-name {
        interface-mac-limit limit;
    }
    site-identifier identifier;
}
```

The numerical identifier can be any number from 1 through 65,534 that uniquely identifies the local VPLS site.

You can include these statements at the following hierarchy levels:

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

Configuring Automatic Site Identifiers for VPLS

When you enable automatic site identifiers, the Junos OS automatically assigns site identifiers to VPLS sites. To configure automatic site identifiers for a VPLS routing instance, include the **automatic-site-id** statement:

```
automatic-site-id {
    collision-detect-time seconds;
    new-site-wait-time seconds;
    reclaim-wait-time minimum seconds maximum seconds;
    startup-wait-time seconds;
}
```

You can include this statement at the following hierarchy levels:

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

NOTE: ACX Series routers do not support the **[edit logical-systems]** hierarchy.

The **automatic-site-id** statement includes a number of options that control different delays in network layer reachability information (NLRI) advertisements. All of these options are configured with default values. See the statement summary for the **automatic-site-id** statement for more information.

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

- **collision-detect-time**—The time in seconds to wait after a claim advertisement is sent to the other routers in a VPLS instance before a PE router can begin using a site identifier. If the PE router receives a competing claim advertisement for the same site identifier during this time period, it initiates the collision resolution procedure for site identifiers.
- **new-site-wait-time**—The time in seconds to wait to receive VPLS information for a newly configured routing instance or a new site. This time interval is also applied whenever the automatic site identifier feature is activated on a VPLS routing instance other than at startup. Effectively, this timer indicates how long to wait before an attempt is made to allocate a site identifier. This timer is also triggered whenever a VPLS routing instance is enabled.

- **reclaim-wait-time**—The time to wait before attempting to claim a site identifier after a collision. A collision occurs whenever an attempt is made to claim a site identifier by two separate VPLS sites.
- **startup-wait-time**—The time in seconds to wait at startup to receive all the VPLS information for the route targets configured on the other PE routers included in the VPLS routing instance.

Configuring the Site Range

When you enable BGP signaling for each VPLS routing instance, you can optionally configure the site range. The site range specifies an upper limit on the maximum site identifier that can be accepted to allow a pseudowire to be brought up. You must specify a value from 1 through 65,534. The default value is 65,534. We recommend using the default. Pseudowires cannot be established to sites with site identifiers greater than the configured site range. If you issue the **show vpls connections** command, such sites are displayed as OR (out of range).

To configure the site range, include the **site-range** statement:

```
site-range number;
```

You can include this statement at the following hierarchy levels:

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

NOTE: ACX Series routers do not support the **[edit logical-systems]** hierarchy.

There are networks that require that the site range be configured using a value smaller than the local site identifier, for example, a hub-and-spoke VPLS with multihomed sites. For this type of network, you need to allow pseudowires to be established between the spoke routers and the hub router. However, you also need to prevent pseudowires from being established between spoke routers directly. Due to the multihoming requirement of spoke sites, Layer 2 VPN NRLIs need to be accepted from other spoke routers (at least from spokes with the same site identifier as the locally configured sites) to determine the status of local spoke routers (active or not active) based on the local preference included in the NRLIs received from the other spoke routers.

This type of VPLS network can be implemented by, for example, numbering hub sites with identifiers 1 through 8 and spoke sites with identifiers 9 and larger. You can then configure a site range of 8 on each of the spoke sites. Although the spoke sites accept NRLIs and install them in the Layer 2 VPN routing tables (allowing the multihomed sites to determine the status of the local site), the spoke sites cannot establish pseudowires directly to the other spoke sites due to the configured site range.

The following configurations illustrate this concept. The configurations are for the VPLS routing instances on three routers, two spoke routers and one hub router:

Router 1—spoke:

```
routing-instance hub-and-spoke {
    no-local-switching;
    protocols {
        vpls {
            site-range 8;
            no-tunnel-services;
            site spoke-9 {
                site-identifier 9 {
                    multi-homing;
                    site-preference primary;
                }
            }
            site spoke-10 {
                site-identifier 10 {
                    multi-homing;
                    site-preference backup;
                }
            }
        }
    }
}
```

Router 2—spoke:

```
routing-instance hub-and-spoke {
    no-local-switching;
    protocols {
        vpls {
            site-range 8;
            no-tunnel-services;
            site spoke-9 {
                site-identifier 9 {
                    multi-homing;
                    site-preference backup;
                }
            }
            site spoke-10 {
                site-identifier 10 {
                    multi-homing;
                    site-preference primary;
                }
            }
        }
    }
}
```

Hub—router 3:

```
routing-instance hub-and-spoke {
    no-local-switching;
    protocols {
        vpls {
            no-tunnel-services;
            site hub {
                site-identifier 1;
```

```
        }
      }
    }
  }
```

Configuring the VPLS Site Interfaces

You must configure an interface for each of the pseudowires you specify for the VPLS site.

To configure an interface for the VPLS site, include the **interface** statement:

```
interface interface-name {
  interface-mac-limit limit;
}
```

You can include these statements at the following hierarchy levels:

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

You can also configure a limit on the number of MAC addresses that can be learned from the specified interface. For more information, see "Limiting the Number of MAC Addresses Learned from an Interface" on page 34.

Configuring the VPLS Site Preference

You can specify the local preference value advertised for a particular VPLS site. The site preference value is specified using the **site-preference** statement configured at the **[edit routing-instances** *routing-instance-name* **protocols vpls site** *site-name***]** hierarchy level. By configuring the **site-preference** statement, a value configured for the **local-preference** statement at the **[edit protocols bgp]** hierarchy level is ignored by the VPLS routing instance. However, you can change the site preference value for VPLS routes exported to other routers by configuring an export policy. When a PE router receives multiple advertisements with the same VPLS edge (VE) device identifier, the advertisement with the highest local preference value is preferred.

To configure the VPLS site preference, include the **site-preference** statement:

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

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

For a list of hierarchy levels at which you can include the **site-preference** statement, see the statement summary section for this statement.

Configuring LDP Signaling for VPLS

You can configure LDP as the signaling protocol for a VPLS routing instance. This functionality is described in RFC 4762, *Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling*.

The Junos OS software does not support all of RFC 4762. When enabling LDP signaling for a VPLS routing instance, network engineers should be aware that only the following values are supported:

- FEC—**128** or **129**
- Control bit—**0**
- Ethernet pseudowire type—**0x0005**
- Ethernet tagged mode pseudowire type—**0x0004**

LDP signaled VPLS supports the Virtual Circuit Connectivity Verification (VCCV) Type Length Value (TLV) for pseudowire label mapping, label database display, and LDP trace. When you enable LDP signaling for a pseudowire, LDP advertises the VCCV capabilities to the neighboring routers. VCCV provides a control channel for a pseudowire and includes both operations and management functions (for example, connectivity verification). This control channel is established between the pseudowire's ingress and egress devices. Once established, connectivity verification messages can be sent over the VCCV control channel.

The Junos OS software supports the following VCCV capabilities for LDP signaled VPLS (defined in RFC 5085 Section 8.1):

- VCCV connectivity check types:
  - Router Alert Label
  - MPLS pseudowire label with TTL=1
- VCCV connectivity verification type:
  - LSP ping

If the peer device also advertises VCCV parameters during pseudowire setup, the Junos OS software selects the set of common advertised parameters to use as the method for performing VCCV OAM on the pseudowire.

The locally advertised and peer advertised VCCV parameters can be viewed using the **show ldp database** command as show here:

```
user@host> show ldp database l2circuit extensive
Input label database, 10.255.245.198:0--10.255.245.194:0
  Label     Prefix
 299872     L2CKT CtrlWord PPP VC 100
            MTU: 4470
            VCCV Control Channel types:
                MPLS router alert label
                MPLS PW label with TTL=1
```

```
                VCCV Control Verification types:
                    LSP ping
    Label       Prefix
                State: Active
                Age: 19:23:08
```

Be aware of the following behavior with regard to TLVs when configuring LDP-signaled VPLS in a network with equipment from other vendors:

- When a Juniper Network's device receives a TLV with an empty address, LDP accepts the TLV.
- When a MAC address is withdrawn, LDP specifies a zero address (0.0.0.0) for the AddressList.

To enable LDP signaling for the set of PE routers participating in the same VPLS routing instance, you need to use the **vpls-id** statement configured at the **[edit routing-instances *routing-instance-name* protocols vpls]** hierarchy level to configure the same VPLS identifier on each of the PE routers. The VPLS identifier must be globally unique. When each VPLS routing instance (domain) has a unique VPLS identifier, it is possible to configure multiple VPLS routing instances between a given pair of PE routers.

LDP signaling requires that you configure a full-mesh LDP session between the PE routers in the same VPLS routing instance. Neighboring PE routers are statically configured. Tunnels are created between the neighboring PE routers to aggregate traffic from one PE router to another. Pseudowires are then signaled to demultiplex traffic between VPLS routing instances. These PE routers exchange the pseudowire label, the MPLS label that acts as the VPLS pseudowire demultiplexer field, by using LDP forwarding equivalence classes (FECs). Tunnels based on both MPLS and generic routing encapsulation (GRE) are supported.

NOTE: **You cannot configure both BGP signaling and LDP signaling for the same VPLS routing instance. If you attempt to configure the statements that enable BGP signaling for the VPLS routing instance (the site, site-identifier, and site-range statements), and the statements that enable LDP signaling for the same instance, neighbor and vpls-id, the commit operation fails.**

To enable LDP signaling for the VPLS routing instance, complete the steps in the following sections:

- Configuring LDP Signaling for the VPLS Routing Instance on page 29
- Configuring LDP Signaling on the Router on page 30

Configuring LDP Signaling for the VPLS Routing Instance

To configure the VPLS routing instance to use LDP signaling, you must configure the same VPLS identifier on each PE router participating in the instance. Specify the VPLS identifier with the **vpls-id** statement:

```
vpls-id vpls-id;
```

You can include this statement at the following hierarchy levels:

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

NOTE: ACX Series routers do not support the **[edit logical-systems]** hierarchy.

To configure the VPLS routing instance to use LDP signaling, you also must include the **neighbor** statement to specify each of the neighboring PE routers that are a part of this VPLS domain:

```
neighbor neighbor-id;
```

You can include this statement at the following hierarchy levels:

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

NOTE: ACX Series routers do not support the **[edit logical-systems]** hierarchy.

Configuring LDP Signaling on the Router

To enable LDP signaling, you need to configure LDP on each PE router participating in the VPLS routing instance. A minimal configuration is to enable LDP on the loopback interface, which includes the router identifier (**router-id**), on the PE router using the **interface** statement:

```
interface interface-name;
```

You can include this statement at the following hierarchy levels:

- **[edit protocols ldp]**
- **[edit logical-systems** *logical-system-name* **protocols ldp]**

NOTE: ACX Series routers do not support the **[edit logical-systems]** hierarchy.

You can enable LDP on all the interfaces on the router using the **all** option for the **interfaces** statement. For more information about how to configure LDP, see the *Junos OS MPLS Applications Configuration Guide*.

Configuring VPLS Routing Instance and VPLS Interface Connectivity

You can configure the VPLS routing instance to take down or maintain its VPLS connections depending on the status of the interfaces configured for the VPLS routing instance. By default, the VPLS connection is taken down whenever a customer-facing interface configured for the VPLS routing instance fails. This behavior can be explicitly configured by specifying the **ce** option for the **connectivity-type** statement:

```
connectivity-type ce;
```

You can alternatively specify that the VPLS connection remain up so long as an Integrated Routing and Bridging (IRB) interface is configured for the VPLS routing instance by specifying the **irb** option for the **connectivity-type** statement:

```
connectivity-type irb;
```

To ensure that the VPLS connection remain up until explicitly taken down, specify the **permanent** option for the **connectivity-type** statement:

```
connectivity-type permanent;
```

This option is reserved for use in configuring Layer 2 Wholesale subscriber networks. See the *Broadband Subscriber Management Solutions Guide* for details about configuring a Layer 2 Wholesale network.

You can include this statement at the following hierarchy levels:

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

NOTE: ACX Series routers do not support the **[edit logical-systems]** hierarchy.

Configuring the VPLS Encapsulation Type

You can specify a VPLS encapsulation type for the pseudowires established between VPLS neighbors. The encapsulation type is carried in the LDP-signaling messages exchanged between VPLS neighbors when pseudowires are created. You might need to alter the encapsulation type depending on what other vendors' equipment is deployed within your network.

VPLS effectively provides a bridge between Ethernet networks. As a consequence, only two encapsulation types are available:

- **ethernet**—Ethernet
- **ethernet-vlan**—Ethernet virtual LAN (VLAN)

If you do not specify an encapsulation type for the VPLS routing instance or the VPLS neighbor, **ethernet** is used.

To specify an encapsulation type for the VPLS routing instance, include the **encapsulation-type** statement:

```
encapsulation-type (ethernet | ethernet-vlan);
```

You can include this statement at the following hierarchy levels:

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

You can also specify an encapsulation type for a specific VPLS neighbor by including the **encapsulation-type** statement at the following hierarchy levels:

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

Configuring the MPLS Routing Table to Leak Routes a Nondefault Routing Instance

You can specify one or more nondefault routing instances where you want MPLS routes to be leaked from the mpls.0 path routing table in the master routing instance. This capability is useful in an L2VPN/VPLS configuration when the remote PE router is learned from the IGP in a nondefault routing instance, because L2VPN/VPLS installs ingress-labeled routes only in the master mpls.0 table.

By default, routes in the mpls.0 routing table in the master routing instance are not leaked to the corresponding routing tables in nondefault routing instances. When L2VPN/VPLS traffic is received on the core-facing interface in a nondefault routing instance, the router performs a lookup in the table that corresponds to that interface, *routing-instance-name*.mpls.0. Because the routes are not leaked by default, then no routes are found in the *routing-instance-name*.mpls.0 routing table and all the incoming traffic is dropped.

To leak MPLS routes to a nondefault routing instance, include the **import-labeled-routes** statement and specify one or more routing instances where the routes need to be leaked:

```
import-labeled-routes [ routing-instance-name ];
```

You can include this statement at the following hierarchy levels:

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

Configuring the VPLS MAC Table Timeout Interval

You can modify the timeout interval for the VPLS table. We recommend you that configure longer values for small, stable VPLS networks and shorter values for large, dynamic VPLS networks. If the VPLS table does not receive any updates during the timeout interval, the

router waits one additional interval before automatically clearing the MAC address entries from the VPLS table.

To modify the timeout interval for the VPLS table, include the **mac-table-aging-time** statement:

mac-table-aging-time *seconds*;

You can include this statement at the following hierarchy levels:

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

NOTE: The **mac-table-aging-time** statement is not available on ACX Series and MX Series routers.

NOTE: ACX Series routers do not support the **[edit logical-systems]** hierarchy.

Configuring the Size of the VPLS MAC Address Table

You can modify the size of the VPLS media access control (MAC) address table. The default table size is 512 MAC addresses, the minimum is 16 addresses, and the maximum is 65,536 addresses.

NOTE: T4000 routers with Type 5 FPCs support up to 262,143 MAC addresses per VPLS routing instance. To enable the improved VPLS MAC address learning limit (that is, 262,143 MAC addresses), you must Include the **enhanced-mode** statement at the **[edit chassis network-services]** hierarchy level, reboot the router, and then modify the size of the VPLS MAC address table.

If the MAC table limit is reached, new MAC addresses can no longer be added to the table. Eventually the oldest MAC addresses are removed from the MAC address table automatically. This frees space in the table, allowing new entries to be added. However, as long as the table is full, new MAC addresses are dropped.

To change the VPLS MAC table size for each VPLS or VPN routing instance, include the **mac-table-size** statement:

mac-table-size *size*;

You can include this statement at the following hierarchy levels:

- **[edit routing-instances *routing-instance-name* protocols vpls]**

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

NOTE: ACX Series routers do not support the **[edit logical-systems]** hierarchy.

When you include the **mac-table-size** statement, the affected interfaces include all interfaces within the VPLS routing instance, including the local interfaces, the LSI interfaces, and the VT interfaces.

Limiting the Number of MAC Addresses Learned from an Interface

You can configure a limit on the number of MAC addresses learned by a VPLS routing instance using the **mac-table-size** statement. If the MAC table limit is reached, new MAC addresses can no longer be added to the table. Eventually the oldest MAC addresses are removed from the MAC address table automatically. This frees space in the table, allowing new entries to be added. However, as long as the table is full, new MAC addresses are dropped.

Because this limit applies to each VPLS routing instance, the MAC addresses of a single interface can consume all the available space in the table, preventing the routing instance from acquiring addresses from other interfaces.

You can limit the number of MAC addresses learned from each interface configured for a VPLS routing instance. To do so, include the **interface-mac-limit** statement:

```
interface-mac-limit limit;
```

You can include this statement at the following hierarchy levels:

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

The **interface-mac-limit** statement affects the local interfaces only (the interfaces facing CE devices).

Configuring the **interface-mac-limit** statement at the **[edit routing-instances *routing-instance-name* protocols vpls]** hierarchy level causes the same limit to be applied to all of the interfaces configured for that specific routing instance.

NOTE: Starting with Junos OS Release 12.3R4, if you do not configure the parameter to limit the number of MAC addresses to be learned by a VPLS instance, the default value is not effective. Instead, if you do not include the interface-mac-limit option at the **[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls site *site-name* interfaces *interface-name*]**, hierarchy level, this setting is not present in the configuration with the default value of 1024 addresses. If you upgrade a router running a Junos OS release earlier than Release 12.3R4 to Release 12.3R4 or later, you must configure the **interface-mac-limit** option with a valid value for it to be saved in the configuration.

You can also limit the number of MAC addresses learned by a specific interface configured for a VPLS routing instance. This gives you the ability to limit particular interfaces that you expect might generate a lot of MAC addresses.

To limit the number of MAC addresses learned by a specific interface, include the **interface-mac-limit** statement at the following hierarchy levels:

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

NOTE: ACX Series routers do not support the **[edit logical-systems]** hierarchy.

The MAC limit configured for an individual interface at this hierarchy level overrides any value configured at the **[edit routing-instances *routing-instance-name* protocols vpls]** hierarchy level. Also, the MAC limit configured using the **mac-table-size** statement can override the limit configured using the **interface-mac-limit** statement.

The MAC address limit applies to customer-facing interfaces only.

Removing Addresses from the MAC Address Database

You can enable MAC flush processing for the VPLS routing instance or for the mesh group under a VPLS routing instance. MAC flush processing removes MAC addresses from the MAC address database that have been learned dynamically. With the dynamically learned MAC addresses removed, MAC address convergence requires less time to complete.

You can clear dynamically learned MAC addresses from the MAC address database by including the **mac-flush** statement:

```
mac-flush [ explicit-mac-flush-message-options ];
```

To clear dynamically learned MAC addresses globally across all devices participating in the routing instance, you can include the statement at the following hierarchy levels:

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

To clear the MAC addresses on the routers in a specific mesh group, you can include the statement at the following hierarchy levels:

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

NOTE: ACX Series routers do not support the **mesh-group** statement.

NOTE: ACX Series routers do not support the **[edit logical-systems]** hierarchy.

For certain cases where MAC flush processing is not initiated by default, you can also specify ***explicit-mac-flush-message-options*** to additionally configure the router to send explicit MAC flush messages under specific conditions. For a list of the explicit MAC flush message options you can include with this statement, see the summary section for this statement.

Related Documentation

- *Configuring Improved VPLS MAC Address Learning on T4000 Routers with Type 5 FPCs*
- *enhanced-mode*
- *show ldp database*

Configuring VPLS Fast Reroute Priority

When a path is rerouted after a link failure by using the MPLS fast reroute feature, the affected next hops are repaired by switching them from the active label switched path (LSP) to the standby LSP. To specify the order in which the next hops are repaired and traffic convergence is restored for VPLS routing instances after a fast reroute event, you can use the **fast-reroute-priority** statement to configure **high**, **medium**, or **low** fast reroute priority for a VPLS routing instance. By default, the fast reroute priority for a VPLS routing instance is **low**.

Next hops are repaired and known unicast, unknown unicast, broadcast, and multicast traffic for VPLS routing instances are restored in the following order, based on the fast reroute priority configuration:

1. Next hops for high-priority VPLS routing instances are repaired.

2. Next hops for medium-priority VPLS routing instances are repaired.
3. Next hops for low-priority VPLS routing instances are repaired.

Because next hops for VPLS routing instances configured with **high** fast reroute priority are repaired first, the traffic traversing high-priority VPLS instances is restored faster than the traffic for VPLS instances configured with **medium** or **low** fast reroute priority. The ability to prioritize specific VPLS routing instances for faster convergence and traffic restoration enables service providers to offer differentiated service levels to their customers.

Within a particular fast reroute priority level (**high**, **medium**, or **low**), no particular order for traffic restoration of VPLS routing instances is followed.

NOTE: VPLS fast reroute priority is not supported on J Series routers.

To configure **high**, **medium**, or **low** fast reroute priority for a VPLS routing instance, include the **fast-reroute-priority** statement:

```
fast-reroute-priority (high | medium | low);
```

You can include this statement at the following hierarchy levels:

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

You can configure fast reroute priority only for routing instances with the **instance-type** set to **vpls**. If you attempt to configure fast reroute priority for a routing instance with an **instance-type** other than **vpls**, the router displays a warning message and the configuration fails.

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

The following example snippet shows configuration of **high** fast reroute priority for a VPLS routing instance named **test-vpls**:

```
test-vpls {
    instance-type vpls;
    forwarding-options {
        fast-reroute-priority high;
    }
}
```

To display the fast reroute priority setting configured for a VPLS routing instance, use the **show route instance detail** operational command. For information about using this command, see the CLI Explorer.

Specifying the VT Interfaces Used by VPLS Routing Instances

By default, the Junos OS automatically selects one of the virtual tunnel (VT) interfaces available to the router for de-encapsulating traffic from a remote site. The Junos OS cycles through the currently available VT interfaces, regularly updating the list of available VT interfaces as new remote sites are discovered and new connections are brought up. However, you can also explicitly configure which VT interfaces will receive the VPLS traffic.

By including the **tunnel-services** statement at the **[edit routing-instances *routing-instance-name* protocols vpls]** hierarchy level, you can specify that traffic for particular VPLS routing instances be forwarded to specific VT interfaces. Doing so allows you to load-balance VPLS traffic among all the available VT interfaces on the router.

The **tunnel-services** statement includes the following options:

- **devices**—Specifies the VT interfaces acceptable for use by the VPLS routing instance. If you do not configure this option, all VT interfaces available to the router can be used for de-encapsulating traffic for this instance.
- **primary**—Specifies the primary VT interface to be used by the VPLS routing instance. The VT interface specified is used to de-encapsulate all VPLS traffic from the MPLS core network for this routing instance. If the VT interface specified is unavailable, then one of the other acceptable VT interfaces (specified in the devices option) is used for handling the VPLS traffic. If you do not configure this option, any acceptable VT interface can be used to de-encapsulate VPLS traffic from the core.

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

To specify that traffic for a particular VPLS routing instance be forwarded to specific VT interfaces, include the **tunnel-services** statement:

```
tunnel-services {
    devices device-names;
    primary primary-device-name;
}
```

These statements can be configured at the following hierarchy levels:

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

Understanding PIM Snooping for VPLS

There are two ways to direct PIM control packets:

- By the use of PIM snooping

- By the use of PIM proxying

PIM snooping configures a device to examine and operate only on PIM hello and join/prune packets. A PIM snooping device snoops PIM hello and join/prune packets on each interface to find interested multicast receivers and populates the multicast forwarding tree with this information. PIM snooping differs from PIM proxying in that both PIM hello and join/prune packets are transparently flooded in the VPLS as opposed to the flooding of only hello packets in the case of PIM proxying. PIM snooping is configured on PE routers connected through pseudowires. PIM snooping ensures that no new PIM packets are generated in the VPLS, with the exception of PIM messages sent through LDP on pseudowires.

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

A device that supports PIM snooping snoops hello packets received on attachment circuits. It does not introduce latency in the VPLS core when it forwards PIM join/prune packets.

To configure PIM snooping on a PE router, use the **pim-snooping** statement at the **[edit routing-instances *instance-name* protocols]** hierarchy level:

```
routing-instances {
    customer {
        instance-type vpls;
        ...
        protocols {
            pim-snooping{
                traceoptions {
                    file pim.log size 10m;
                    flag all;
                    flag timer disable;
                }
            }
        }
    }
}
```

Example: Configuring PIM Snooping for VPLS explains the PIM snooping method. The use of the PIM proxying method is not discussed here and is outside the scope of this document. For more information about PIM proxying, see PIM Snooping over VPLS.

Related Documentation

- *Example: Configuring PIM Snooping for VPLS*

VPLS Label Blocks Operation

A virtual private LAN service (VPLS) is a Layer 2 (L2) service that emulates a local area network (LAN) across a wide area network (WAN). VPLS labels are defined and exchanged in the Border Gateway Protocol (BGP) control plane. In the Junos OS implementation, label blocks are allocated and used in the VPLS control plane for two primary functions: autodiscovery and signaling.

- Autodiscovery—A method for automatically recognizing each provider edge (PE) router in a particular VPLS domain, using BGP update messages.
- Signaling—Each pair of PE routers in a VPLS domain sends and withdraws VPN labels to each other. The labels are used to establish and dismantle pseudowires between the routers. Signaling is also used to transmit certain characteristics of a pseudowire.

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

The PE router uses BGP extended communities to identify the members of its VPLS. Once the PE router discovers its members, it is able to establish and tear down pseudowires between members by exchanging and withdrawing labels and transmitting certain characteristics of the pseudowires.

The PE router sends common update messages to all remote PE routers, using a distinct BGP update message, thereby reducing the control plane load. This is achieved by using VPLS label blocks.

Elements of Network Layer Reachability Information

VPLS BGP network layer reachability information (NLRI) is used to exchange VPLS membership and parameters. The elements of a VPLS BGP NLRI are defined in Table 3 on page 40.

Table 3: NLRI Elements

Element	Acronym	Description	Default Size (Octets)
Length		Total length of the NLRI size represented in bytes.	2
Route Distinguisher	RD	Unique identifier for each routing instance configured on a PE.	8
VPLS Edge ID	VE ID	Unique number to identify the edge site.	2
VE Block Offset	VBO	Value used to identify a label block from which a label value is selected to set up pseudowires for a remote site.	2

Table 3: NLRI Elements *(continued)*

Element	Acronym	Description	Default Size (Octets)
VE Block Size	VBS	Indicates the number of pseudowires that peers can have in a single block.	2
Label Base	LB	Starting value of the label in the advertised label block.	3

Requirements for NLRI Elements

Junos OS requires a unique route distinguisher (RD) for each routing instance configured on a PE router. A PE router might use the same RD across a VPLS (or VPN) domain or it might use different RDs. Using different RDs helps identify the originator of the VPLS NLRI.

The VPLS edge (VE) ID can be a unique VE ID, site ID, or customer edge (CE) ID. The VE ID is used by a VPLS PE router to index into label blocks used to derive the transmit and receive VPN labels needed for transport of VPLS traffic. The VE ID identifies a particular site, so it needs to be unique within the VPLS domain, except for some scenarios such as multihoming.

All PE routers have full mesh connectivity with each other to exchange labels and set up pseudowires. The VE block size (VBS) is a configurable value that represents the number of label blocks required to cover all the pseudowires for the remote peer.

A single label block contains 8 labels (1 octet) by default. The default VBS in Junos OS is 2 blocks (2 octets) for a total of 16 labels.

How Labels are Used in Label Blocks

Each PE router creates a mapping of the labels in the label block to the sites in a VPLS domain. A PE router advertising a label block with a block offset indicates which sites can use the labels to reach it. When a PE router is ready to advertise its membership to a VPLS domain, it allocates a label block and advertises the VPLS NLRI. In this way, other PE routers in the same VPLS domain can learn of the existence of the VPLS and set up pseudowires to it if needed. The VPLS NLRI advertised for this purpose is referred to as the *default VPLS NLRI*. The label block in the default VPLS NLRI is referred to as the *default label block*.

Label Block Composition

A label block (set of labels) is used to reach a given site ID. A single label block contains 8 labels (1 octet) by default. The VBS is 2 octets by default in Junos OS.

The label block advertised is defined as a label base (LB) and a VE block size (VBS). It is a contiguous set of labels (LB, LB+1,...LB+VBS-1). For example, when Router PE-A sends a VPLS update, it sends the same label block information to all other PE routers. Each PE router that receives the LB advertisement infers the label intended for Router PE-A by adding its own site ID to the label base.

In this manner, each receiving PE gets a unique label for PE-A for that VPLS. This simple method is enhanced by using a VE block offset (VBO).

A label block is defined as: <Label Base (LB), VE block offset (VBO), VE block size (VBS)> is the set {LB+VBO, LB+VBO+1,...,LB+VBO+VBS-1}.

Label Blocks in Junos OS

Instead of a single large label block to cover all VE IDs in a VPLS, the Junos OS implementation contains several label blocks, each with a different label base. This makes label block management easier, and also allows Router PE-A to seemlessly integrate a PE router joining a VPLS with a site ID not covered by the set of label blocks that Router PE-A has already advertised.

VPLS Label Block Structure

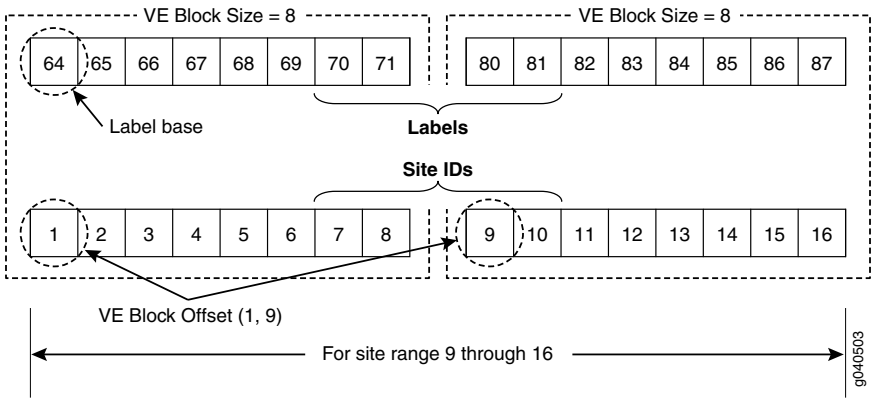
This section illustrates how a label block is uniquely identified.

A VPLS BGP NLRI with site ID V, VE block offset VBO, VE block size VBS, and label base LB communicates the following to its peers:

- Label block for V: Labels from LB to (LB + VBS -1).
- Remote VE set for V: from VBO to (VBO + VBS -1).

The label block advertised is a set of labels used to reach a given site ID. If there are several label blocks, the remote VE set helps to identify which label block to use. The example in Figure 3 on page 42 illustrates label blocks. There are two blocks and each block has eight labels. In this example, the label values are 64 to 71 and 80 to 87.

Figure 3: VPLS Label Block Structure



To create a one-to-one mapping of these 16 labels to 16 sites, assume the site IDs are the numbers 1 to 16, as shown in the illustration. The site block indicates which site ID can use which label in the label block. So, in the first block, site ID 1 uses 64, site ID 2 uses 65, and so forth. Finally, site ID 8 uses 71. The 9th site ID will use the second block instead of the first block.

The labels are calculated by comparing the values of VBO <= Local site ID < (VBO +VBS). Consequently, site ID 9 uses 80, site ID 10 uses 81, and so on.

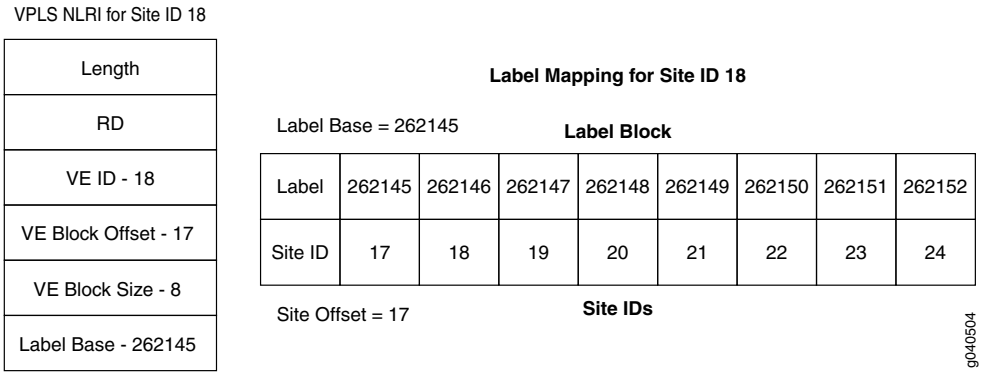
To further illustrate the one-to-one mapping of labels to sites, assume a label block with site offset of 1 and a label base of 10. The combination of label base and block offset contained in the VPLS NLRI provides the mapping of labels to site IDs. The block offset is the starting site ID that can use the label block as advertised in the VPLS NLRI.

To advertise the default VPLS NLRI, a PE router picks a starting block offset that fits its own site ID and is such that the end block offset is a multiple of a single label block. In Junos OS a single label block is eight labels by default.

The end block offset is the last site ID that maps to the last label in the label block. The end offset for the first block is 8 which maps to label 17 and the second block is 16. For example, a site with ID 3 picks a block offset of 1 and advertises a label block of size 8 to cover sites with IDs 1 to 8. A site with ID 10 picks a block offset of 9 to cover sites with IDs 9 to 16.

The VPLS NLRI shown in Figure 4 on page 43 is for site ID 18. The label base contains value 262145. The block offset contains value 17. The illustration shows which site IDs correspond to which labels.

Figure 4: Label Mapping Example



If a PE router configured with site ID 17 is in the same VPLS domain as a PE router configured with site ID 18, it receives the VPLS NLRI as shown in Figure 2. So it uses label 262145 to send traffic to site 18. Similarly, a PE router configured with site ID 19 uses label 262147 to send traffic to a PE router configured with site ID 18. However, only PE routers configured with site IDs 17 to 24 can use the label block shown to set up pseudowires.

Related Documentation

- *Example: Building a VPLS From Router 1 to Router 3 to Validate Label Blocks*

Configuring the Label Block Size for VPLS

VPLS MPLS packets have a two-label stack. The outer label is used for normal MPLS forwarding in the service provider's network. If BGP is used to establish VPLS, the inner label is allocated by a PE router as part of a label block. One inner label is needed for each remote VPLS site. Four sizes are supported. We recommend using the default size of 8, unless the network design requires a different size for optimal label usage, to allow the router to support a larger number of VPLS instances.

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

If you allocate a large number of small label blocks to increase efficiency, you also increase the number of routes in the VPLS domain. This has an impact on the control plane overhead.

Changing the configured label block size causes all existing pseudowires to be deleted. For example, if you configure the label block size to be 4 and then change the size to 8, all existing label blocks of size 4 are deleted, which means that all existing pseudowires are deleted. The new label block of size 8 is created, and new pseudowires are established.

Four label block sizes are supported: 2, 4, 8, and 16. Consider the following scenarios:

- 2—Allocate the label blocks in increments of 2. For a VPLS domain that has only two sites with no future expansion plans.
- 4—Allocate the label blocks in increments of 4.
- 8 (default)—Allocate the label blocks in increments of 8.
- 16—Allocate the label blocks in increments of 16. A label block size of 16 enables you to minimize the number of routes in the VPLS domain. Use this setting only if the number of routes is the most important concern.

Configure the label block size:

```
[edit routing-instances instance-name protocols vpls]
user@router# set label-block-size 2
```

Related Documentation

- Configuring VPLS Routing Instances on page 21

PE Router Mesh Groups for VPLS Routing Instances

A PE router mesh group consists of a set of routers participating in a VPLS routing instance that share the same signaling protocol, either BGP or LDP. Each VPLS routing instance can have just one BGP mesh group. However, you can configure multiple LDP mesh groups for each routing instance.

The Junos OS can support up to 16 mesh groups on MX Series routers and up to 128 on M Series and T Series routers. However, two mesh groups are created by default, one for

the CE routers and one for the PE routers. Therefore, the maximum number of user-defined mesh groups is 14 for MX Series routers and 126 for M Series and T Series routers.

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

The Junos OS supports both forwarding equivalency class (FEC) 128 and FEC 129. FEC 129 uses VPLS autodiscovery to convey endpoint information. FEC 128 requires manually configured pseudowires.

The following describes the behavior of mesh groups in regards to BGP-signaled PE routers and LDP-signaled PE routers:

- BGP-signaled PE routers—Automatically discovered PE routers that use BGP for signaling are associated with the default VE mesh group. You cannot configure the Junos OS to associate these routers with a user-defined VE mesh group.
- LDP-signaled PE routers (FEC 128)—PE routers statically configured using FEC-128 LDP signaling are placed in a default mesh group. However, you can configure a VE mesh group and associate each LDP FEC-128 neighbor with it. Each configured VE mesh group contains a set of VEs that are in the same interior gateway protocol (IGP) routing instance and are fully meshed with each other in the control and data planes.
- LDP-signaled PE routers (FEC 129)—Configuration for a mesh group for FEC 129 is very similiar to the configuration for FEC 128.

  Note the following differences for FEC 129:

  - Each user-defined mesh group must have a unique route distinguisher. Do not use the route distinguisher that is defined for the default mesh group at the **[edit routing-intances]** hierarchy level.
  - Each user-defined mesh group must have its own import and export route target.
  - Each user-defined mesh group can have a unique Layer 2 VPN ID. By default, all the mesh groups that are configured for the a VPLS routing-instance use the same Layer 2 VPN ID, the one that you configure at the **[edit routing-instances]** hierarchy level.

Related Documentation

- *Example: Configuring BGP Autodiscovery for LDP VPLS*

CHAPTER 5

Associating Interfaces with VPLS

- Configuring Interfaces for VPLS Routing on page 47
- VPLS and Aggregated Ethernet Interfaces on page 54
- Configuring VPLS Without a Tunnel Services PIC on page 55

Configuring Interfaces for VPLS Routing

On each PE router and for each VPLS routing instance, specify which interfaces are intended for the VPLS traffic traveling between PE and CE routers. To specify the interface for VPLS traffic, include the **interface** statement in the routing instance configuration:

```
interface interface-name;
```

You can include this statement at the following hierarchy levels:

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

You must also define each interface by including the following statements:

```
vlan-taggingvlan-tagging;
encapsulation encapsulation-type;
unit logical-unit-number {
   family vpls;
   vlan-id vlan-id-number;
}
```

You can include these statements at the following hierarchy levels:

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

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

The following sections provide enough information to enable you to configure interfaces for VPLS routing.

- Configuring the VPLS Interface Name on page 48
- Configuring VPLS Interface Encapsulation on page 48
- Enabling VLAN Tagging on page 51
- Configuring VLAN IDs for Logical Interfaces on page 51
- Enabling VLANs for Hub and Spoke VPLS Networks on page 52
- Configuring Aggregated Ethernet Interfaces for VPLS on page 52

Configuring the VPLS Interface Name

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

physical.logical

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

A logical interface can be associated with only one routing instance.

If you enable a routing protocol on all instances by specifying **interfaces all** when configuring the master instance of the protocol at the **[edit protocols]** hierarchy level, and you configure a specific interface for VPLS routing at the **[edit routing-instances *routing-instance-name*]** hierarchy level, the latter interface statement takes precedence and the interface is used exclusively for VPLS.

If you explicitly configure the same interface name at both the **[edit protocols]** and **[edit routing-instances *routing-instance-name*]** hierarchy levels and then attempt to commit the configuration, the commit operation fails.

Configuring VPLS Interface Encapsulation

You need to specify an encapsulation type for each PE-router-to-CE-router interface configured for VPLS. This section describes the **encapsulation** statement configuration options available for VPLS.

To configure the encapsulation type on the physical interface, include the **encapsulation** statement:

encapsulation (ethernet-vpls | ether-vpls-over-atm-llc |extended-vlan-vpls | vlan-vpls);

NOTE: ACX Series routers do not support the *ether-vpls-over-atm-llc* and *extended-vlan-vpls* options for encapsulation.

You can include the **encapsulation** statement for physical interfaces at the following hierarchy levels:

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

NOTE: **ACX Series routers do not support the [edit logical-systems] hierarchy.**

You can configure the following physical interface encapsulations for VPLS routing instances:

- **ethernet-vpls**—Use Ethernet VPLS encapsulation on Ethernet interfaces that have VLAN 802.1Q tagging and VPLS enabled. The PE router expects to receive Ethernet frames with VLAN tags that are not service-delimiting. The Ethernet frames are not meaningful to the PE router and cannot be used by the service provider to separate customer traffic.

  On M Series routers (except the M320 router), the 4-port Fast Ethernet TX PIC and the 1-port, 2-port, and 4-port, 4-slot Gigabit Ethernet PICs can use the Ethernet VPLS encapsulation type.

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

- **extended-vlan-vpls**—Use extended virtual LAN (VLAN) VPLS encapsulation on Ethernet interfaces that have VLAN 802.1Q tagging and VPLS enabled and that must accept packets carrying TPIDs 0x8100, 0x9100, and 0x9901. On M Series routers (except the M320 router), the 4-port Fast Ethernet TX PIC and the 1-port, 2-port, and 4-port, 4-slot Gigabit Ethernet PICs can use the Ethernet VPLS encapsulation type.

  NOTE: **The built-in Gigabit Ethernet PIC on an M7i router does not support extended VLAN VPLS encapsulation.**

- **vlan-vpls**—Use VLAN VPLS encapsulation on Ethernet interfaces with VLAN 802.1Q tagging and VPLS enabled. The PE router expects to receive Ethernet frames with VLAN tags that are service-delimiting. These VLAN tags can be used by the service provider to separate customer traffic. For example, LAN traffic from different customers can flow through the same service provider switch, which can then apply VLAN tags to distinguish one customer's traffic from the others. The traffic can then be forwarded to the PE router.

  Interfaces with VLAN VPLS encapsulation accept packets carrying standard TPID values only. On M Series routers (except the M320 router), the 4-port Fast Ethernet

TX PIC and the 1-port, 2-port, and 4-port, 4-slot Gigabit Ethernet PICs can use the Ethernet VPLS encapsulation type.

To configure the encapsulation type for logical interfaces, include the **encapsulation** statement:

```
encapsulation (ether-vpls-over-atm-llc | vlan-vpls);
```

You can include the **encapsulation** statement for logical interfaces at the following hierarchy levels:

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

You can configure the following logical interface encapsulations for VPLS routing instances:

- **ether-vpls-over-atm-llc**—Use Ethernet VPLS over Asynchronous Transfer Mode (ATM) logical link control (LLC) encapsulation to bridge Ethernet interfaces and ATM interfaces over a VPLS routing instance (as described in RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*). Packets from the ATM interfaces are converted to standard ENET2/802.3-encapsulated Ethernet frames with the frame check sequence (FCS) field removed. This encapsulation type is supported on ATM intelligent queuing (IQ) interfaces only.
- **vlan-vpls**—Use VLAN VPLS encapsulation on Ethernet interfaces with VLAN 802.1Q tagging and VPLS enabled. The PE router expects to receive Ethernet frames with VLAN tags that are service-delimiting. These VLAN tags can be used by the service provider to separate customer traffic. For example, LAN traffic from different customers can flow through the same service provider switch, which can then apply VLAN tags to distinguish one customer's traffic from the others. The traffic can then be forwarded to the PE router.

  Interfaces with VLAN VPLS encapsulation accept packets carrying standard TPID values only. On M Series routers (except the M320 router), the 4-port Fast Ethernet TX PIC and the 1-port, 2-port, and 4-port, 4-slot Gigabit Ethernet PICs can use the Ethernet VPLS encapsulation type.

NOTE: **Label-switched interfaces (LSIs) do not support VLAN VPLS encapsulation. Therefore, you can only use VLAN VPLS encapsulation on a PE-router-to-CE-router interface and not a core-facing interface.**

When you configure the physical interface encapsulation as **vlan-vpls**, you also need to configure the same interface encapsulation for the logical interface. You need to configure the **vlan-vpls** encapsulation on the logical interface because the **vlan-vpls** encapsulation allows you to configure a mixed mode, where some of the logical interfaces use regular Ethernet encapsulation (the default for logical interfaces) and some use **vlan-vpls**.

NOTE: Starting with Junos OS release 13.3, a commit error occurs when you configure **vlan-vpls** encapsulation on a physical interface and configure **family inet** on one of the logical units. Previously, it was possible to commit this invalid configuration.

Enabling VLAN Tagging

Junos OS supports receiving and forwarding routed Ethernet frames with 802.1Q virtual local area network (VLAN) tags and running the Virtual Router Redundancy Protocol (VRRP) over 802.1Q-tagged interfaces. For VPLS to function properly, configure the router to receive and forward frames with 802.1Q VLAN tags by including the **vlan-tagging** statement at the **[edit interfaces *interface-name*]** hierarchy level:

```
[edit interfaces interface-name]
vlan-tagging;
```

Gigabit Ethernet interfaces can be partitioned. You can assign up to 4095 different logical interfaces, one for each VLAN, but you are limited to a maximum of 1024 VLANs on any single Gigabit Ethernet or 10-Gigabit Ethernet port. Fast Ethernet interfaces can also be partitioned, with a maximum of:

- 1024 logical interfaces for the 4-port FE PIC
- 1024 logical interfaces for the 2-port Fixed Interface Card (FIC) on an M7i router
- 16 logical interfaces for the M40e router

Table 4 on page 51 lists VLAN ID ranges by interface type.

Table 4: VLAN ID Range by Interface Type

Interface Type	VLAN ID Range
Fast Ethernet	512 through 1023
Gigabit Ethernet	512 through 4094

Configuring VLAN IDs for Logical Interfaces

You can bind a VLAN identifier to a logical interface by including the **vlan-id** statement:

```
vlan-id number;
```

You can include this statement at the following hierarchy levels:

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

You can also configure a logical interface to forward packets and learn MAC addresses within each VPLS routing instance configured with a VLAN ID that matches a VLAN ID specified in a list using the **vlan-id-list** statement. VLAN IDs can be entered individually using a space to separate each ID, entered as an inclusive list separating the starting VLAN ID and ending VLAN ID with a hyphen, or a combination of both.

For example, to configure the VLAN IDs 20 and 45 and the range of VLAN IDs between 30 and 40, issue the following command from the CLI:

set interfaces ge-1/0/1 unit 1 vlan-id-list [20 30-40 45];

To configure a list of VLAN IDs for a logical interface, include the **vlan-id-list** statement:

vlan-id-list *list-of-vlan-ids*;

You can include this statement at the following hierarchy levels:

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

NOTE: ACX Series routers do not support the **[edit logical-systems]** hierarchy.

For more information about how to configure VLANs, see the *Junos OS Network Interfaces Library for Routing Devices*. For detailed information about how VLAN identifiers in a VPLS routing instance are processed and translated, see the *MX Series Layer 2 Configuration Guide*.

Enabling VLANs for Hub and Spoke VPLS Networks

For hub and spoke VPLS networks, you need to configure the **swap** option for the **output-vlan-map** statement on the hub facing interface of each spoke PE router. The **output-vlan-map** statement ensures that the vlan ID of the spoke PE router matches the VLAN ID of the hub PE router in the VPLS network. The following configuration example illustrates an interface configuration with the output-vlan-map statement included:

```
[edit interfaces xe-4/0/0]
vlan-tagging;
encapsulation flexible-ethernet-services;
unit 610 {
   encapsulation vlan-ccc;
   vlan-id 610;
   output-vlan-map swap;
}
```

Configuring Aggregated Ethernet Interfaces for VPLS

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

For more information about how aggregated Ethernet interfaces function in the context of VPLS, see "VPLS and Aggregated Ethernet Interfaces" on page 54.

To configure aggregated Ethernet interfaces for VPLS, configure the interface for the VPLS routing instance as follows:

```
interfaces aex {
    vlan-tagging;
    encapsulation encapsulation-type;
    unit logical-unit-number {
        vlan-id number;
    }
}
```

You can configure the following physical link-layer encapsulation types for the VPLS aggregated Ethernet interface:

- **ethernet-vpls**
- **extended-vlan-vpls**
- **flexible-ethernet-services**
- **vlan-vpls**

NOTE: ACX Series routers do not support the **extended-vlan-vpls** and **vlan-vpls** encapsulation types.

For the **interface** configuration statement, in **ae*x***, the ***x*** represents the interface instance number to complete the link association; ***x*** can be from 0 through 127, for a total of 128 aggregated interfaces.

For more information about how to configure aggregated Ethernet interfaces, see the *Ethernet Interfaces Feature Guide for Routing Devices*.

The aggregated Ethernet interface must also be configured for the VPLS routing instance as shown in the following example:

```
[edit]
routing-instances {
    green {
        instance-type vpls;
        interface ae0.0;
        route-distinguisher 10.255.234.34:1;
        vrf-target target:11111:1;
        protocols {
            vpls {
                site-range 10;
                site green3 {
                    site-identifier 3;
                }
            }
        }
    }
```

```
}
```

Interface **ae0.0** represents the aggregated Ethernet interface in the routing instance configuration. The VPLS routing instance configuration is otherwise standard.

VPLS and Aggregated Ethernet Interfaces

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

Forwarding is based on a lookup of the DA MAC address. For the remote site, if a packet needs to be forwarded over an LSP, the packet is encapsulated and forwarded through the LSP. If the packet destination is a local site, it is forwarded over appropriate local site interface. For an aggregated Ethernet interface on the local site, packets are sent out of the load-balanced child interface. The Packet Forwarding Engine acquires the child link to transmit the data.

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

When a received packet does not have a match to a MAC address in the forwarding database, the packet is forwarded over a set of interfaces determined from a lookup in the flooding database based on the incoming interface. This is denoted by a flood next hop. The flood next hop can include the aggregated Ethernet interface as the set of interfaces to flood the packet.

Each VPLS routing instance configured on a PE router has its own forwarding database entries that associate all of the MAC addresses the VPLS routing instance acquires with each corresponding port. A route is added to the kernel with a MAC address as the prefix and the next hop used to reach the destination. The route is an interface if the destination is local. For a remote destination, the route is a next hop for the remote site.

For local aggregated Ethernet interfaces on M Series and T Series routers, learning is based on the parent aggregated Ethernet logical interface. To age out MAC addresses for aggregated Ethernet interfaces, each Packet Forwarding Engine is queried to determine where the individual child interfaces are located. MAC addresses are aged out based on the age of the original interface.

For MX Series routers and EX Series switches, when a Dense Port Concentrator (DPC) learns a MAC address it causes the Routing Engine to age out the entry. This behavior applies to all logical interfaces. For an aggregated Ethernet logical interface, once all the member DPCs have aged out the entry, the entry is deleted from the Routing Engine.

Related Documentation

- Configuring Interfaces for VPLS Routing on page 47
- Configuring Aggregated Ethernet Interfaces for VPLS on page 52

Configuring VPLS Without a Tunnel Services PIC

VPLS normally uses a dynamic virtual tunnel logical interface on a Tunnel Services PIC to model traffic from a remote site (a site on a remote PE router that is in a VPLS domain). All traffic coming from a remote site is treated as coming in over the virtual port representing this remote site, for the purposes of Ethernet flooding, forwarding, and learning. An MPLS lookup based on the inner VPN label is done on a PE router. The label is stripped and the Layer 2 Ethernet frame contained within is forwarded to a Tunnel Services PIC. The PIC loops back the packet and then a lookup based on Ethernet MAC addresses is completed. This approach requires that the router have a Tunnel Services PIC and that the PE router complete two protocol lookups.

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

You can configure VPLS without a Tunnel Services PIC by configuring the **no-tunnel-services** statement. This statement creates a label-switched interface (LSI) to provide VPLS functionality. An LSI MPLS label is used as the inner label for VPLS. This label maps to a VPLS routing instance. On the PE router, the LSI label is stripped and then mapped to a logical LSI interface. The Layer 2 Ethernet frame is then forwarded using the LSI interface to the correct VPLS routing instance.

By default, VPLS requires a Tunnel Services PIC. To configure VPLS on a router without a Tunnel Services PIC and create an LSI, include the **no-tunnel-services** statement:

```
no-tunnel-services;
```

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

To configure a VPLS routing instance on a router without a tunnel services PIC, include the **no-tunnel-services** statement at the **[edit routing-instances *routing-instance-name* protocols vpls]** hierarchy level. To configure static VPLS on a router without a tunnel services PIC, include the **no-tunnel-services** statement at the **[edit protocols vpls static-vpls]** hierarchy level.

When you configure VPLS without a Tunnel Services PIC by including the **no-tunnel-services** statement, the following limitations apply:

- An Enhanced FPC is required.
- ATM1 interfaces are not supported.
- Aggregated SONET/SDH interfaces are not supported as core-facing interfaces.
- Channelized interfaces are not supported as core-facing interfaces.
- GRE-encapsulated interfaces are not supported as core-facing interfaces.

Related Documentation

- Configuring Static Pseudowires for VPLS on page 59

CHAPTER 6

Configuring Pseudowires

- VPLS Path Selection Process for PE Routers on page 57
- Configuring Static Pseudowires for VPLS on page 59

VPLS Path Selection Process for PE Routers

The VPLS path selection process is used to select the best path between a remote PE router and a local PE router in a VPLS network. This path selection process is applied to routes received from both single-homed and multi-homed PE routers.

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

When the VPLS path selection process is complete, a PE router is made the designated VPLS edge (VE) device. The designated VE device effectively acts as the endpoint for the VPLS pseudowire that is signaled from the remote PE router. Once a PE router is made the designated VE device, a pseudowire can be signaled between the remote PE router and the local PE router and then VPLS packets can begin to flow between the PE routers.

Routes from multihomed PE routers connected to the same customer site share the same site ID, but can have different route distinguishers and block offsets. You can alter the configurations of the route distinguishers and block offsets to make a router more likely or less likely to be selected as the designated VE device.

On each PE router in the VPLS network, the best path to the CE device is determined by completing the following VPLS path selection process on each route advertisement received:

1. If the advertisement has the down bit set to 0, the advertisement is discarded.
2. Select the path with a higher preference. The preference attribute is obtained from the site-preference configured using the **site-preference** statement at the **[edit routing-instances *routing-instance-name* protocols vpls site *site-name*]** hierarchy level. If the site preference is 0, the preference attribute is obtained from the local preference.
3. If the preference values are the same, select the path with the lower router ID.

4. If the router IDs are the same, the routes are from the same PE router and the advertisement is considered to be an update. The router ID corresponds to the value of the originator ID for the BGP attribute (if present). Otherwise, the IP address for the remote BGP peer is used.
5. If the block offset values are the same, the advertisement is considered to be an update.

Once the VPLS path selection process has been completed and the designated VE device has been selected, a pseudowire is signaled between the remote PE router and the local PE router.

NOTE: **The VPLS path selection process works the same whether or not the route has been received from another PE router, a route reflector, or an autonomous system border router (ASBR).**

When the remote PE router establishes or refreshes a pseudowire to the local PE router, it verifies that the prefix is in the range required for the site ID based on the block offset and label range advertised by the designated VE device. If the prefix is out of range, the pseudowire status is set to out of range.

The following cases outline the potential decisions that could be made when a PE router completes the VPLS path selection process for a Layer 2 advertisement in the VPLS network:

- The PE router originated one of the advertisements and selected its own advertisement as the best path.

  This PE router has been selected as the designated VE device. Selection as the designated VE device triggers the creation of pseudowires to and from the other PE routers in the VPLS network. If the remote customer site is multihomed, the designated VE device triggers the creation of pseudowires to and from only the designated VE device for the remote site.

- The PE router originated one of the advertisements but did not select its own advertisement as the best path.

  This PE router is a redundant PE router for a multihomed site, but it was not selected as the designated VE device. However, if this PE router has just transitioned from being the designated VE device (meaning it was receiving traffic from the remote PE routers addressed to the mulithomed customer site), the PE router tears down all the pseudowires that it had to and from the other PE routers in the VPLS network.

- The PE router received the route advertisements and selected a best path. It did not originate any of these advertisements because it was not connected to the customer site.

  If the best path to the customer site (the designated VE device) has not changed, nothing happens. If the best path has changed, this PE router brings up pseudowires to and from the newly designated VE device and tears down the pseudowires to and from the previously designated VE device.

If this PE router does not select a best path after running the VPLS path selection process, then the local PE router does not consider the remote site to exist.

When a VE device receives an advertisement for a Layer 2 NLRI that matches its own site ID but the site is not multihomed, the pseudowire between the VE device and the transmitting PE router transitions to a site collision state and is not considered to be up.

Related Documentation

- BGP Route Reflectors for VPLS on page 14

Configuring Static Pseudowires for VPLS

You can configure a VPLS domain using static pseudowires. A VPLS domain consists of a set of PE routers that act as a single virtual Ethernet bridge for the customer sites connected to these routers. By configuring static pseudowires for the VPLS domain, you do not need to configure the LDP or BGP protocols that would normally be used for signaling. However, if you configure static pseudowires, any changes to the VPLS network topology have to be managed manually.

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

Static pseudowires require that you configure a set of in and out labels for each pseudowire configured for the VPLS domain. You still need to configure a VPLS identifier and neighbor identifiers for a static VPLS domain. You can configure both static and dynamic neighbors within the same VPLS routing instance.

To configure a static pseudowire for a VPLS neighbor, include the **static** statement:

```
static (Protocols VPLS) {
    incoming-label label;
    outgoing-label label;
}
```

You must configure an incoming and outgoing label for the static pseudowire using the **incoming-label** and **outgoing-label** statements. These statements identify the static pseudowire's incoming traffic and destination.

To configure a static pseudowire for a VPLS neighbor, include the **static** statement at the **[edit routing-instances *routing-instance-name* protocols vpls neighbor *address*]** hierarchy level.

You can also configure the **static** statement for a backup neighbor (if you configure the neighbor as static the backup must also be static) by including it at the **[edit routing-instances *routing-instance-name* protocols vpls neighbor *address* backup-neighbor *address*]** hierarchy level and for a mesh group by including it at the **[edit routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name* neighbor *address*]** hierarchy level.

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

To enable static VPLS on a router, you need to either configure a virtual tunnel interface (requires the router to have a tunnel services PIC) or you can configure a label switching interface (LSI). To configure an LSI, include the **no-tunnel-services** statement at the **[edit protocols vpls static-vpls]** hierarchy level. For more information, see "Configuring VPLS Without a Tunnel Services PIC" on page 55.

NOTE: Static pseudowires for VPLS using an LSI is supported on MX series routers and EX Series switches only. For M series and T series routers, a tunnel services PIC is required.

If you issue a **show vpls connections** command, static neighbors are displayed with **"SN"** next to their addresses in the command output.

Related Documentation

- Configuring VPLS Without a Tunnel Services PIC on page 55

CHAPTER 7

Configuring Multihoming

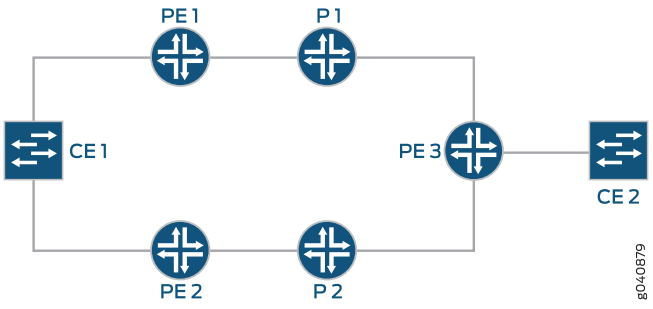
- VPLS Multihoming Overview on page 61
- VPLS Multihoming Reactions to Network Failures on page 63
- BGP and VPLS Path Selection for Multihomed PE Routers on page 64
- Configuring VPLS Multihoming (FEC 128) on page 66

VPLS Multihoming Overview

Virtual private LAN service (VPLS) multihoming enables you to connect a customer site to two or more PE routers to provide redundant connectivity. A redundant PE router can provide network service to the customer site as soon as a failure is detected. VPLS multihoming helps to maintain VPLS service and traffic forwarding to and from the multihomed site in the event of the following types of network failures:

- PE router to CE device link failure
- PE router failure
- MPLS-reachability failure between the local PE router and a remote PE router

Figure 5: CE Device Multihomed to Two PE Routers



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

Figure 5 on page 61 illustrates how a CE device could be multihomed to two PE routers. Device CE1 is multihomed to Routers PE1 and PE2. Device CE2 has two potential paths to reach Device CE1, but only one path is active at any one time. If Router PE1 were the designated VPLS edge (VE) device (also called a designated forwarder), BGP would signal a pseudowire from Router PE3 to Router PE1. If a failure occurred over this path, Router PE2 would be made the designated VE device, and BGP would re-signal the pseudowire from Router PE3 to Router PE2.

Multihomed PE routers advertise network layer reachability information (NLRI) for the multihomed site to the other PE routers in the VPLS network. The NLRI includes the site ID for the multihomed PE routers. For all of the PE routers multihomed to the same CE device, you need to configure the same site ID. The remote VPLS PE routers use the site ID to determine where to forward traffic addressed to the customer site. To avoid route collisions, the site ID shared by the multihomed PE routers must be different than the site IDs configured on the remote PE routers in the VPLS network.

Although you configure the same site ID for each of the PE routers multihomed to the same CE device, you can configure unique values for other parameters, such as the route distinguisher. These values help to determine which multihomed PE router is selected as the designated VE device to be used to reach the customer site.

BEST PRACTICE: **We recommend that you configure unique route distinguishers for each multihomed PE router. Configuring unique route distinguishers helps with faster convergence when the connection to a primary multihomed PE router goes down. If you configure unique route distinguishers, the other PE routers in the VPLS network must maintain additional state for the multihomed PE routers.**

Remote PE routers in the VPLS network need to determine which of the multihomed PE routers should forward traffic to reach the CE device. To make this determination, remote PE routers use the VPLS path-selection process to select one of the multihomed PE routers based on its NLRI advertisement. Because remote PE routers pick only one of the NLRI advertisements, it establishes a pseudowire to only one of the multihomed PE routers, the PE router that originated the winning advertisement. This prevents multiple paths from being created between sites in the network, preventing the formation of Layer 2 loops. If the selected PE router fails, all PE routers in the network automatically switch to the backup PE router and establish new pseudowires to it.

BEST PRACTICE: **To prevent the formation of Layer 2 loops between the CE devices and the multihomed PE routers, we recommend that you employ the Spanning Tree Protocol (STP) on your CE devices. Layer 2 loops can form due to incorrect configuration. Temporary Layer 2 loops can also form during convergence after a change in the network topology.**

The PE routers run the BGP path selection procedure on locally originated and received Layer 2 route advertisements to establish that the routes are suitable for advertisement to other peers, such as BGP route reflectors. If a PE router in a VPLS network is also a

route reflector, the path selection process for the multihomed site has no effect on the path selection process performed by this PE router for the purpose of reflecting Layer 2 routes. Layer 2 prefixes that have different route distinguishers are considered to have different NLRIs for route reflection. The VPLS path selection process enables the route reflector to reflect all routes that have different route distinguishers to the route reflector clients, even though only one of these routes is used to create the VPLS pseudowire to the multihomed site.

Junos OS supports VPLS multihoming for both FEC 128 and FEC 129. Support for FEC 129 is added in Junos OS Release 12.3.

Related Documentation

- Configuring VPLS Multihoming (FEC 128) on page 66
- *Advantages of Using Autodiscovery for VPLS Multihoming*
- *Example: Configuring VPLS Multihoming (FEC 129)*
- *Example: VPLS Multihoming, Improved Convergence Time*

VPLS Multihoming Reactions to Network Failures

VPLS multihoming is designed to protect customer sites from a loss of network connectivity in the event of the following types of network failures:

- Link failure between the CE device and the PE router—BGP on the PE router is notified when the link goes down. BGP sets the circuit status vector bit in the MP_REACH_NLRI to indicate that the circuit is down.

  If all of the VPLS local attachment circuits are down, then BGP modifies the down bit in the VPLS advertisement Layer2-Extended-Community to indicate that the customer site is down. When the bit is modified, BGP advertises the route to all of the remote PE routers to notify them that the circuit (and site) is down. Each of the remote PE routers run the BGP and VPLS path selection procedures again and reroute the VPLS pseudowires as needed.

- MPLS connectivity failure to the remote PE router—On the multihomed PE router, BGP discovers that MPLS cannot connect to the BGP next hop in the service provider's network. BGP modifies the circuit status vector bit in the MP_REACH_NLRI to indicate that the LSP is down. Once the bit is modified, BGP readvertises the route to all of the remote PE routers to notify them that connectivity from the local site to the remote site is down.

  The remote PE routers each run the BGP and VPLS path selection procedures again. With the LSP to the original multihomed PE router down, the remote PE routers designate the backup multihomed PE router as the VE device for the multihomed customer site. The pseudowires to and from the remote PE routers are then rerouted to the backup multihomed PE router.

- PE router failure—When either the multihomed PE router or the BGP process running on it fails, the remote PE routers detect the expiration of the holdtimer, bring down their peering sessions, and delete the Layer 2 advertisements from that multihomed PE router. The remote PE routers each run the BGP and VPLS path selection procedures again and reroute their pseudowires to the backup multihomed PE router.

Alternatively, the remote PE routers could discover that the BGP next hop, represented by the failed multihomed PE router, is unreachable. For this case, the remote PE routers mark the Layer 2 routes advertised by the multihomed PE router as unreachable. The remote PE routers each run the BGP and VPLS path selection procedures again and reroute their pseudowires to the backup multihomed PE router.

The remote PE routers behave in the same manner if you reconfigure the local preference attribute of the primary multihomed PE router (effectively performing an administrative failover to the backup multihomed PE router). On the primary multihomed PE router, BGP advertises a Layer 2 update with the new local preference attribute to all of the remote PE routers. The remote PE routers each run the BGP and VPLS path selection procedures again and reroute their pseudowires to the backup multihomed PE router.

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

BGP and VPLS Path Selection for Multihomed PE Routers

The BGP and VPLS path selection procedures are used to select the best path between the remote PE router and one of the multihomed PE routers. As part of these path selection procedures, one of the multihomed PE routers is made the designated VE device. The designated VE device effectively acts as the endpoint for the VPLS pseudowire from the remote PE router. Once a multihomed PE router is made the designated VE device, a pseudowire can be created between the remote PE router and the multihomed PE router.

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

Routes from multihomed PE routers connected to the same customer site share the same site ID, but can have different route distinguishers and block offsets. On each PE router in the VPLS network, the best path to the multihomed PE router is determined by completing the following VE device-selection procedures on each route advertisement received from a multihomed PE router:

1. BGP designated VE device-selection procedure—Runs before the VPLS designated VE device-selection procedure. However, the BGP designated VE device-selection procedure is used only when the route distinguishers for the multihomed PE routers are identical. If the route distinguishers are unique, only the VPLS designated VE device-selection procedure is run.
2. VPLS designated VE device-selection procedure—Runs after the BGP designated VE device-selection procedure. However, if the route distinguishers for each multihomed PE router are unique, the advertisements are not considered relevant to the BGP designated VE device-selection procedure. As a consequence, only the VPLS designated VE device-selection procedure is used.

The BGP designated VE device-selection procedure is as follows:

1. If the advertisement has the down bit set to 0, the advertisement is discarded.
2. Select the path with a higher preference. The preference attribute is obtained from the site-preference configured using the **site-preference** statement at the [edit routing-instances *routing-instance-name* protocols vpls site *site-name*] hierarchy level. If the site-preference is 0, the preference attribute is obtained from the local-preference.
3. If the preference values are the same, select the path with the lower router-id.
4. If the router-ids are the same, the routes are from the same PE router and the advertisement is considered to be an update.

Once the BGP designated VE device-selection procedure is complete, the VPLS designated VE device-selection procedure begins. This procedure is carried out regardless of the outcome of the BGP designated VE device-selection procedure:

1. If the advertisement has the down bit set to 0, the advertisement is discarded.
2. Select the path with a higher preference. The preference attribute is obtained from the site-preference configured using the **site-preference** statement at the [edit routing-instances *routing-instance-name* protocols vpls site *site-name*] hierarchy level. If the site-preference is 0, the preference attribute is obtained from the local-preference.
3. If the preference values are the same, select the path with the lower router-id.
4. If the router-ids are the same, select the path with a lower route distinguisher.
5. If the route distinguishers are the same, select the path with the lower block offset value.
6. If the block offset values are the same, the advertisement is considered to be an update.

Once the BGP and VPLS path selection procedures have been completed and the designated VE devices have been selected, a pseudowire can be created between the remote PE router and the multihomed PE router.

When the remote PE router establishes or refreshes a pseudowire to the local PE router, it verifies that the prefix is in the range required for the site ID based on the block offset and label range advertised by the designated VE device. If the prefix is out of range, the pseudowire status is set to out of range.

The following cases outline the potential decisions that could be made when a PE router completes the BGP and VPLS path selection procedures for a Layer 2 advertisement in the VPLS network:

- The PE router originated one of the multihomed advertisements and selected its own advertisement as the best path.

  This PE router has been selected as the designated VE device. Selection as the designated VE device triggers the creation of pseudowires to and from the other PE

routers in the VPLS network. When the remote customer site is also multihomed, the designated VE device triggers the creation of pseudowires to and from only the designated VE device for the remote site.

- The PE router originated one of the multihomed advertisements but did not select its own advertisement as the best path.

  This PE router is one of the redundant PE routers for the multihomed site; it was not selected as the designated VE device. However, if this PE router has just transitioned from being the designated VE device (meaning it was receiving traffic from the remote PE routers addressed to the mulithomed customer site), the PE router tears down all the pseudowires that it had to and from the other PE routers in the VPLS network.

- The PE router receives the multihomed advertisements and selects a best path; it does not originate any of these advertisements because it is not connected to the multihomed customer site.

  If the preferred path to the customer site (the designated VE device) has not changed, nothing happens. If the preferred path has changed, this PE router brings up pseudowires to and from the newly designated VE device and tears down the pseudowires to and from the previously designated VE device.

  If this PE router does not select a best path after running the BGP and VPLS path selection process, the local PE router does not consider the remote site to exist.

When a VE device receives an advertisement for a Layer 2 NLRI which matches its own site ID but the site is not multihomed, the pseudowire between it and the transmitting PE router transitions to a site collision state and is not considered to be up.

Configuring VPLS Multihoming (FEC 128)

VPLS multihoming allows you to connect a customer site to multiple PE routers to provide redundant connectivity while preventing the formation of Layer 2 loops in the service provider's network. A VPLS site multihomed to two or more PE routers provides redundant connectivity in the event of a PE router-to-CE device link failure or the failure of a PE router. For more information about VPLS multihoming, see "VPLS Multihoming Overview" on page 61.

NOTE: If you want to enable multihoming for a VPLS routing instance, you cannot also enable LDP signaling. You can only enable BGP signaling.

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

The following sections describe how to configure VPLS multihoming. Some information is also provided on single-homed site configuration versus multihomed site configuration.

- VPLS Multihomed Site Configuration on page 67
- VPLS Single-Homed Site Configuration on page 68

VPLS Multihomed Site Configuration

The following describes the requirements for a VPLS multihomed site configuration:

- Assign the same site ID on all PE routers connected to the same CE devices.
- Assign unique route distinguishers for each multihomed PE router.
- Reference all interfaces assigned to the multihomed VPLS site on each PE router. Only one of these interfaces is used to send and receive traffic for this site at a time.
- Either designate a primary interface or allow the router to select the interface to be used as the primary interface.

  If the router selects the interface, the interface used to connect the PE router to the site depends on the order in which interfaces are listed in the PE router's configuration. The first operational interface in the set of configured interfaces is chosen to be the designated interface. If this interface fails, the next interface in the list is selected to send and receive traffic for the site.
- Configure multihoming for the site.

The following configuration shows the statements you need to configure to enable VPLS multihoming:

```
[edit routing-instances routing-instance-name]
instance-type vpls;
interface interface-name;
interface interface-name;
protocols vpls {
  site site-name {
    active-interface {
      any;
      primary interface-name;
    }
    interface interface-name;
    interface interface-name;
    multi-homing;
    site-identifier number;
  }
}
route-distinguisher (as-number:id | ip-address:id);
```

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

Most of these statements are explained in more detail in the rest of this chapter. The following sections explain how to configure the statements that are specific to VPLS multihoming:

- Specifying an Interface as the Active Interface on page 68
- Configuring Multihoming on the PE Router on page 68

Specifying an Interface as the Active Interface

You need to specify one of the interfaces for the multihomed site as the primary interface. If there are multiple interfaces, the remaining interfaces are activated only when the primary interface goes down. If no active interfaces are configured at the site level, all traffic for a VPLS site travels through a single, non-multihomed PE router.

You must configure one of the following options for the **active-interface** statement:

- **any**—One configured interface is randomly designated as the active interface for the VPLS site.
- **primary**—Specify the name of the multihomed interface to be used as the primary interface by the VPLS site.

To specify a multihomed interface as the primary interface for the VPLS site, include the **active-interface** statement:

```
active-interface {
    any;
    primary interface-name;
}
```

You can include this statement at the following hierarchy levels:

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

Configuring Multihoming on the PE Router

When a CE device is connected to the same VPLS site on more than one PE router, include the **multi-homing** statement on all associated PE routers. Configuration of this statement tracks BGP peers. If no BGP peer is available, VPLS deactivates all active interfaces for a site. To specify that the PE router is part of a multihomed VPLS site, include the **multi-homing** statement:

```
multi-homing;
```

You can include this statement at the following hierarchy levels:

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

Include the **multi-homing** statement on all PE routers associated with a particular VPLS site.

VPLS Single-Homed Site Configuration

All VPLS single-homed sites are connected to the same default VE device. All interfaces in a VPLS routing instance that are not configured as part of a multihomed site are assumed to be single-homed to the default VE device.

Related Documentation

- VPLS Multihoming Overview on page 61
- *Example: VPLS Multihoming, Improved Convergence Time*
- active-interface on page 109
- multi-homing on page 141

CHAPTER 8

Configuring Point-to-Multipoint LSPs

- Flooding Unknown Traffic Using Point-to-Multipoint LSPs in VPLS on page 71
- Mapping VPLS Traffic to Specific LSPs on page 75

Flooding Unknown Traffic Using Point-to-Multipoint LSPs in VPLS

For a VPLS routing instance, you can flood unknown unicast, broadcast, and multicast traffic using point-to-multipoint (also called *P2MP*) LSPs. By default, VPLS relies upon ingress replication to flood unknown traffic to the members of a VPLS routing instance. This can cause replication of data at routing nodes shared by multiple VPLS members, as shown in Figure 6 on page 71. The flood data is tripled between PE router PE1 and provider router P1 and doubled between provider routers P1 and P2. By configuring point-to-multipoint LSPs to handle flood traffic, the VPLS routing instance can avoid this type of traffic replication in the network, as shown in Figure 7 on page 71.

Figure 6: Flooding Unknown VPLS Traffic Using Ingress Replication

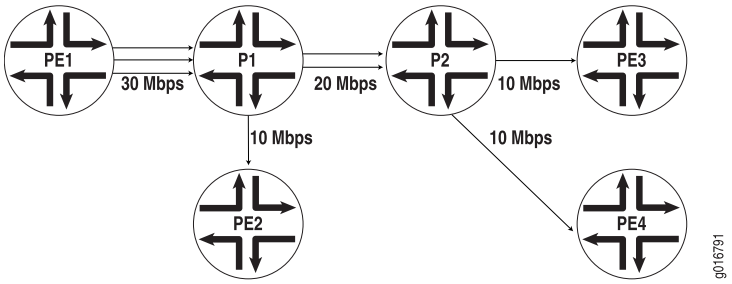
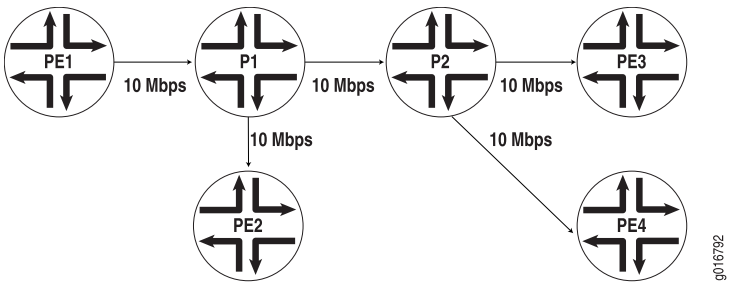


Figure 7: Flooding Unknown VPLS Traffic Using a Point-to-Multipoint LSP



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

The point-to-multipoint LSP used for VPLS flooding can be either static or dynamic. In either case, for each VPLS routing instance, the PE router creates a dedicated point-to-multipoint LSP. All of the neighbors of the VPLS routing instance are added to the point-to-multipoint LSP when the feature is enabled. If there are *n* PE routers in the VPLS routing instance, *n* point-to-multipoint LSPs are created in the network where each PE router is the root of the point-to-multipoint tree and includes the rest of the *n – 1* PE routers as leaf nodes. If you configured static point-to-multipoint LSPs for flooding, any additional VPLS neighbors added to the routing instance later are not automatically added to the point-to-multipoint LSP. You will need to manually add the new VPLS neighbors to the static point-to-multipoint flooding LSP. If you configure dynamic point-to-multipoint LSPs, whenever VPLS discovers a new neighbor through BGP, a sub-LSP for this neighbor is added to the point-to-multipoint LSP for the routing instance.

This feature can be enabled incrementally on any PE router that is part of a specific VPLS routing instance. The PE routers can then use point-to-multipoint LSPs to flood traffic, whereas other PE routers in the same VPLS routing instance can still use ingress replication to flood traffic. However, when this feature is enabled on any PE router, you must ensure that all PE routers in the VPLS routing instance that participate in the flooding of traffic over point-to-multipoint LSPs are upgraded to Junos OS Release 8.3 or later to support this feature.

To flood unknown unicast, broadcast, and multicast traffic using point-to-multipoint LSPs, configure the **rsvp-te** statement as follows:

```
rsvp-te {
  label-switched-path-template (Multicast) {
    (default-template | lsp-template-name);
  }
  static-lsp lsp-name;
}
```

You can include this statement at the following hierarchy levels:

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

You can configure either a static point-to-multipoint LSP for VPLS flooding or a dynamic point-to-multipoint LSP.

NOTE: You cannot specify both the **static** and **label-switched-path-template** statements at the same time.

The following sections describe how to configure static and dynamic point-to-multipoint LSPs for flooding unknown traffic in a VPLS routing instance:

- Configuring Static Point-to-Multipoint Flooding LSPs on page 73
- Configuring Dynamic Point-to-Multipoint Flooding LSPs on page 73

Configuring Static Point-to-Multipoint Flooding LSPs

The **static-lsp** option creates a static flooding point-to-multipoint LSP that includes all of the neighbors in the VPLS routing instance. Flood traffic is sent to all of the VPLS neighbors using the generated point-to-multipoint LSP. VPLS neighbors added to the routing instance later are not automatically added to the point-to-multipoint LSP. You will need to manually add the new VPLS neighbors to the static point-to-multipoint flooding LSP. By configuring static point-to-multipoint LSPs for flooding, you have more control over which path each sub-LSP follows.

To configure a static flooding point-to-multipoint LSP, specify the name of the static flooding point-to-multipoint LSP by including the **static-lsp** statement:

```
static-lsp lsp-name;
```

You can include this statement at the following hierarchy levels:

- **[edit routing-instances *routing-instance-name* provider-tunnel rsvp-te]**
- **[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel rsvp-te]**

Configuring Dynamic Point-to-Multipoint Flooding LSPs

To configure a dynamic point-to-multipoint flooding LSP, include the **label-switched-path-template** statement option at the **[edit routing-instances *routing-instance-name* provider-tunnel rsvp-te]** hierarchy level:

```
[edit routing-instances routing-instance-name provider-tunnel rsvp-te]
label-switched-path-template (Multicast) {
   (default-template | lsp-template-name);
}
```

You can automatically generate the point-to-multipoint LSP to be used for flooding unknown traffic or you can manually configure the point-to-multipoint LSP:

- Configuring Dynamic Point-to-Multipoint Flooding LSPs with the Default Template on page 73
- Configuring Dynamic Point-to-Multipoint Flooding LSPs with a Preconfigured Template on page 74

Configuring Dynamic Point-to-Multipoint Flooding LSPs with the Default Template

The **default-template** option, specified at the **[edit routing-instances *routing-instance-name* provider-tunnel rsvp-te label-switched-path-template]** hierarchy level, causes the point-to-multipoint LSPs to be created with default parameters. The default parameters are for a minimally configured point-to-multipoint LSP. The name of this

point-to-multipoint LSP is also generated automatically and is based on the following model:

id:vpls:*router-id*:*routing-instance-name*

The following **show** command output for **show mpls lsp p2mp** illustrates how a point-to-multipoint flood LSP name could appear if you configure the **label-switched-path-template** statement with the **default-template** option:

```
user@host> show mpls lsp p2mp ingress
Ingress LSP: 2 sessions P2MP name: static, P2MP branch count: 3
To              From            State Rt ActivePath       P     LSPname
10.255.14.181   10.255.14.172   Up     0                  *     vpn02-vpn11
10.255.14.177   10.255.14.172   Up     0 path2            *     vpn02-vpn07
10.255.14.174   10.255.14.172   Up     0 path3            *     vpn02-vpn04
P2MP name: 9:vpls:10.255.14.172:green, P2MP branch count: 2
To              From            State Rt ActivePath       P     LSPname
10.255.14.177   10.255.14.172   Up     0                  *
11:vpls:10.255.14.172:green
10.255.14.174   10.255.14.172   Up     0                  *
10:vpls:10.255.14.172:green
Total 5 displayed, Up 5, Down 0
```

The dynamically generated point-to-multipoint LSP name is **9:vpls:10.255.14.172:green**.

Configuring Dynamic Point-to-Multipoint Flooding LSPs with a Preconfigured Template

You can configure a point-to-multipoint flooding LSP template for the VPLS routing instance. The template allows you to specify the properties of the dynamic point-to-multipoint LSPs that are used to flood traffic for the VPLS routing instance. You can specify all of the standard options available for a point-to-multipoint LSP within this template. These properties are inherited by the dynamic point-to-multipoint flood LSPs.

To configure a point-to-multipoint LSP template for flooding VPLS traffic, specify all of the properties you want to include in a point-to-multipoint LSP configuration. To specify this LSP as a point-to-multipoint flooding template, include the **p2mp** and **template** statements:

```
p2mp;
template;
```

You can include these statements at the following hierarchy levels:

- **[edit protocols mpls label-switched-path *p2mp-lsp-template-name*]**
- **[edit logical-systems *logical-system-name* protocols mpls label-switched-path *p2mp-lsp-template-name*]**

For more information about how to configure the **p2mp** statement and point-to-multipoint LSPs, see the *Junos OS MPLS Applications Configuration Guide*.

Once you have configured the point-to-multipoint LSP template, specify the name of the point-to-multipoint LSP template with the **label-switched-path-template** statement:

```
label-switched-path-template (Multicast) p2mp-lsp-template-name;
```

You can include this statement at the following hierarchy levels:

- **[edit routing-instances *routing-instance-name* provider-tunnel rsvp-te]**
- **[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel rsvp-te]**

Mapping VPLS Traffic to Specific LSPs

You can map VPLS traffic to specific LSPs by configuring forwarding table policies. This procedure is optional but can be useful. The following example illustrates how you can map lower priority VPLS routing instances to slower LSPs while mapping other higher priority VPLS routing instances to faster LSPs. In this example configuration, **a-to-b1** and **a-to-c1** are high-priority LSPs between the PE routers, while **a-to-b2** and **a-to-c2** are low-priority LSPs between the PE routers.

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

To map VPLS traffic, include the **policy-statement vpls-priority** statement:

```
policy-statement vpls-priority {
  term a {
    from {
      rib mpls.0;
      community company-1;
    }
    then {
      install-nexthop lsp [ a-to-b1 a-to-c1 ];
      accept;
    }
  }
  term b {
    from {
      rib mpls.0;
      community company-2;
    }
    then {
      install-nexthop lsp-regex [ "^a-to-b2$" "^a-to-c2$" ];
      accept;
    }
  }
}
community company-1 members target:11111:1;
community company-2 members target:11111:2;
```

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

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

Include the **export** statement to apply the **vpls-priority** policy to the forwarding table:

```
export vpls-priority;
```

You can include this statement at the following hierarchy levels:

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

For more information about how to configure routing policies, see the *Routing Policies, Firewall Filters, and Traffic Policers Feature Guide*.

CHAPTER 9

Configuring BGP Path Selection for Layer 2 VPNs

- Enabling BGP Path Selection for Layer 2 VPNs and VPLS on page 78

Enabling BGP Path Selection for Layer 2 VPNs and VPLS

Layer 2 VPNs and VPLS share the same path selection process for determining the optimal path to reach all of the destinations shared within a single routing instance. For Layer 2 VPN and VPLS topologies, the path selection process is straightforward if there is just a single path from each PE router to each CE device. However, the path selection process becomes more complex if the PE routers receive two or more valid paths to reach a specific CE device.

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

The following network scenarios provide examples of what might cause a PE router to receive more than one valid path to reach a specific CE device:

- Multihoming—One or more CE devices within a routing instance are multihomed to two or more PE routers. Each multihomed CE device has at least two valid paths.
- Route reflectors—There are multiple route reflectors deployed within the same network and they are supporting PE routers within the same routing instance. Due to time delays in large complex networks, the route reflectors can separately receive a different valid path to reach a CE device at different times. When they readvertise these valid paths, a PE router could receive two or more separate but apparently valid paths to the same CE device.

By default, Juniper Networks routers use just the designated forwarder path selection algorithm to select the best path to reach each Layer 2 VPN or VPLS routing instance destination (for more information, see "VPLS Path Selection Process for PE Routers" on page 57). However, you can also configure the routers in your network to use both the BGP path selection algorithm and the designated forwarder path selection algorithm as follows:

- On the Provider routers within the service providers network, the standard BGP path selection algorithm is used (for more information, see *Understanding BGP Path Selection*). Using the standard BGP path selection for Layer 2 VPN and VPLS routes allows a service provider to leverage the existing Layer 3 VPN network infrastructure to also support Layer 2 VPNs and VPLS. The BGP path selection algorithm also helps to ensure that the service provider's network behaves predictably with regard to Layer 2 VPN and VPLS path selection. This is particularly important in networks employing route reflectors and multihoming.

  When a Provider router receives multiple paths for the same destination prefix (for example, a multihomed CE device), one path is selected based on the BGP path selection algorithm and placed in the bgp.l2vpn.0 routing table and the appropriate *instance*.l2vpn.0 routing table.

- When a PE router receives all of the available paths to each CE device, it runs the designated forwarder path selection algorithm to select the preferred path to reach each CE device, independently of the results of the earlier BGP path selection algorithm run on the Provider router. The VPLS designated forwarder algorithm uses the D-bit, preference, and PE router identifier to determine which of the valid paths to each CE device to use. The PE router might select a path to reach a CE device which is different from the path selected by the BGP-based Provider routers. In this scenario, the following is the expected behavior for traffic sent to the multihomed CE device:

  - If the path selected by the remote PE router is available, traffic will traverse the network to the multihomed CE device using the remote PE router's preferred path (again, ignoring the path selected by the BGP-based Provider routers).
  - If the path selected by the remote PE router fails:

    1. The Provider routers switch the traffic destined for the multihomed CE device to the alternate path as soon as failure is detected.
    2. The Provider routers notify the remote PE routers of the path failure.
    3. The remote PE routers update their routing tables accordingly.

  For more information about the VPLS designated forwarder path selection algorithm, see "VPLS Path Selection Process for PE Routers" on page 57. This algorithm is also described in the Internet draft draft-kompella-l2vpn-vpls-multihoming-03.txt, *Multi-homing in BGP-based Virtual Private LAN Service*.

To enable the BGP path selection algorithm for Layer 2 VPN and VPLS routing instances, complete the following steps:

1. Run Junos OS Release 12.3 or later on all of the PE and Provider routers participating in Layer 2 VPN or VPLS routing instances.

   Attempting to enable this functionality on a network with a mix of routers that both do and do not support this feature can result in anomalous behavior.

2. Specify a unique route distinguisher on each PE router participating in a Layer 2 VPN or VPLS routing instance.

3. Configure the **l2vpn-use-bgp-rules** statement on all of the PE and Provider routers participating in Layer 2 VPN or VPLS routing instances.

   You can configure this statement at the **[edit protocols bgp path-selection]** hierarchy level to apply this behavior to all of the routing instances on the router or at the **[edit routing-instances *routing-instance-name* protocols bgp path-selection]** hierarchy level to apply this behavior to a specific routing instance.

Related Documentation

- *Understanding BGP Path Selection*
- VPLS Path Selection Process for PE Routers on page 57
- *VPLS Feature Guide for EX9200 Switches*
- *l2vpn-use-bgp-rules*

- *route-distinguisher*

CHAPTER 10

Configuring Load Balancing and Performance

- Configuring VPLS Load Balancing on page 81

Configuring VPLS Load Balancing

By default, when there are multiple equal-cost paths to the same destination for the active route, the Junos OS uses a hash algorithm to select one of the next-hop addresses to install in the forwarding table. Whenever the set of next hops for a destination changes, the next-hop address is reselected using the hash algorithm.

You can configure the Junos OS so that, for the active route, all next-hop addresses for a destination are installed in the forwarding table. This feature is called per-packet load balancing. You can use load balancing to spread traffic across multiple paths between routers. You can also configure per-packet load balancing to optimize VPLS traffic flows across multiple paths.

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

You can load-balance VPLS traffic based on Layer 2 media access control (MAC) information, IP information and MPLS labels, or MPLS labels only.

NOTE: For platform support information, see family multiservice.

To optimize VPLS traffic flows across multiple paths, include the **family multiservice** statement at the **[edit forwarding-options hash-key]** hierarchy level:

```
family multiservice {
    destination-mac;
    label-1;
    label-2;
    payload {
        ip {
            layer-3 {
                (destination-ip-only | source-ip-only);
```

```
        }
        layer-3-only;
        layer-4;
      }
    }
    source-mac;
    symetric-hash {
      complement;
    }
  }
```

You can configure one or more of the following options to load-balance using the specified packet information:

- **destination-mac**—Include the destination-address MAC information in the hash key for Layer 2 load balancing.
- **source-mac**—Include the source-address MAC information in the hash key.
- **label-1**—Include the first MPLS label in the hash key. Used for including a one-label packet for per-flow load balancing of IPv4 VPLS traffic based on IP information and MPLS labels.
- **label-2** —Include the second MPLS label in the hash key. If both **label-1** and **label-2** are specified, the entire first label and the first 16 bits of the second label are hashed.
- **payload**—Include the packet's IP payload in the hash key.
- **ip**—Include the IP address of the IPv4 or IPv6 payload in the hash key.
- **layer-3-only**—Include only the Layer 3 information from the packet's IP payload in the hash key.
- **layer-3**—Include Layer 3 information from the packet's IP payload in the hash key.
- **destination-address-only**—Include only the destination IP address in the payload in the hash key.

  NOTE: **You can include either the source-address-only or the destination-address-only statement, not both. They are mutually exclusive.**

- **source-address-only**—Include only the source IP address in the payload in the hash key.

  NOTE: **You can include either the source-address-only or the destination-address-only statement, not both. They are mutually exclusive.**

- **layer-4**—Include Layer 4 information from the packet's IP payload in the hash key.
- **symmetric-hash**—Configure the symmetric hash or symmetric hash complement for configuring symmetrical load balancing on an 802.3ad Link Aggregation Group.
- **complement**—Include the complement of the symmetric hash in the hash key.

For more information about how to configure per-packet load balancing, see the*Routing Policies, Firewall Filters, and Traffic Policers Feature Guide*.

Related Documentation

- *Configuring VPLS Load Balancing Based on IP and MPLS Information*
- *Configuring VPLS Load Balancing on MX Series 3D Universal Edge Routers*

CHAPTER 11

Configuring Class of Service and Firewall Filters in VPLS

- Configuring EXP-Based Traffic Classification for VPLS on page 85
- Configuring Firewall Filters and Policers for VPLS on page 85
- Firewall Filter Match Conditions for VPLS Traffic on page 90

Configuring EXP-Based Traffic Classification for VPLS

You can enable EXP classification on traffic entering core facing VPLS LSI interfaces on a VPLS routing instance by configuring either a logical tunnel interface (**lt-**) or the **no-tunnel-services** statement. By configuring either of these, a default EXP classifier is enabled on every core facing interface that includes **family mpls** in its configuration. This feature works on MX Series routers and EX Series switches only. You can configure an EXP classifier explicitly at the **[edit class-of-service]** hierarchy level. For more information about EXP classifiers, see the *Class of Service Feature Guide for Routing Devices*.

To enable EXP classification on traffic entering core facing VPLS LSI interfaces on a VPLS routing instance, include the **no-tunnel-services** statement:

```
no-tunnel-services;
```

You can include this statement at the following hierarchy levels:

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

Related Documentation

- Configuring Firewall Filters and Policers for VPLS on page 85
- Firewall Filter Match Conditions for VPLS Traffic on page 90

Configuring Firewall Filters and Policers for VPLS

You can configure both firewall filters and policers for VPLS. Firewall filters allow you to filter packets based on their components and to perform an action on packets that match

the filter. Policers allow you to limit the amount of traffic that passes into or out of an interface.

VPLS filters and policers act on a Layer 2 frame that includes the media access control (MAC) header (after any VLAN rewrite or other rules are applied), but does not include the cyclical redundancy check (CRC) field.

You can apply VPLS filters and policers on the PE router to customer-facing interfaces only.

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

NOTE: The behavior of firewall filters processing with MAC addresses differs between DPCs and MPCs. On MPCs, interface filters are always applied before MAC learning occurs. The input forwarding table filter is applied after MAC learning is completed. However, on DPCs, MAC learning occurs independently of the application of filters. If the CE-facing interface of the PE where the firewall filter is applied is an MPC, then the MAC entry times out and is never learned again. However, if the CE-facing interface of the PE where the firewall filter is applied is an DP, then the MAC entry is not timed out and if the MAC address entry is manually cleared, it is relearned.

The following sections explain how to configure filters and policers for VPLS:

- Configuring a VPLS Filter on page 86
- Configuring a VPLS Policer on page 89

Configuring a VPLS Filter

To configure a filter for VPLS, include the **filter** statement at the **[edit firewall family vpls]** hierarchy level:

```
[edit firewall family vpls]
filter filter-name {
    interface-specific;
    term term-name {
        from {
            match-conditions;
        }
        then {
            actions;
        }
    }
}
```

For more information about how to configure firewall filters, see the *Routing Policies, Firewall Filters, and Traffic Policers Feature Guide*. For information on how to configure a

VPLS filter match condition, see "Firewall Filter Match Conditions for VPLS Traffic" on page 90.

To configure a filter for VPLS traffic, complete the following tasks:

- Configuring an Interface-Specific Counter for VPLS on page 87
- Configuring an Action for the VPLS Filter on page 87
- Configuring VPLS FTFs on page 87
- Changing Precedence for Spanning-Tree BPDU Packets on page 88
- Applying a VPLS Filter to an Interface on page 88
- Applying a VPLS Filter to a VPLS Routing Instance on page 88
- Configuring a Filter for Flooded Traffic on page 89

Configuring an Interface-Specific Counter for VPLS

When you configure a firewall filter for VPLS and apply it to multiple interfaces, you can specify individual counters specific to each interface. This allows you to collect separate statistics on the traffic transiting each interface.

To generate an interface-specific counter for VPLS, you configure the **interface-specific** statement. A separate instantiation of the filter is generated. This filter instance has a different name (based on the interface name) and collects statistics on the interface specified only.

To configure interface-specific counters, include the **interface-specific** statement at the **[edit firewall family vpls filter *filter-name*]** hierarchy level:

```
[edit firewall family vpls filter filter-name]
interface-specific;
```

NOTE: The counter name is restricted to 24 bytes. If the renamed counter exceeds this maximum length, it might be rejected.

Configuring an Action for the VPLS Filter

You can configure the following actions for a VPLS filter at the **[edit firewall family vpls filter *filter-name* term *term-name* then]** hierarchy level: **accept**, **count**, **discard**, **forwarding-class**, **loss-priority**, **next**, **policer**.

Configuring VPLS FTFs

Forwarding table filters (FTFs) are filters configured for forwarding tables. For VPLS, they are attached to the destination MAC (DMAC) forwarding table of the VPLS routing instance. You define VPLS FTFs in the same manner as any other type of FTF. You can only apply a VPLS FTF as an input filter.

To specify a VPLS FTF, include the **filter input** statement at the **[edit routing-instance *routing-instance-name* forwarding-options family vpls]** hierarchy level:

```
[edit routing-instance routing-instance-name forwarding-options family vpls]
```

filter input *filter-name*;

Changing Precedence for Spanning-Tree BPDU Packets

Spanning tree BPDU packets are automatically set to a high precedence. The queue number on these packets is set to 3. On M Series routers (except the M320 router) by default, a queue value of 3 indicates high precedence. To enable this higher precedence on BPDU packets, an instance-specific BPDU precedence filter named **default_bpdu_filter** is automatically attached to the VPLS DMAC table. This filter places a high precedence on all packets sent to **01:80:c2:00:00:00/24**.

You can overwrite this filter by configuring a VPLS FTF filter and applying it to the VPLS routing instance. For more information, see "Configuring VPLS FTFs" on page 87 and "Applying a VPLS Filter to a VPLS Routing Instance" on page 88.

Applying a VPLS Filter to an Interface

To apply a VPLS filter to an interface, include the **filter** statement:

```
filter {
  group index;
  input input-filter-name;
  output output-filter-name;
}
```

You can include this statement at the following hierarchy levels:

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

NOTE: ACX Series routers do not support the **[edit logical-systems]** hierarchy.

In the **input** statement, list the name of the VPLS filter to be evaluated when packets are received on the interface. In the **output** statement, list the name of the VPLS filter to be evaluated when packets are transmitted on the interface.

NOTE: For output interface filters, MAC addresses are learned after the filter action is completed. When an output interface filter's action is **discard**, the packet is dropped before the MAC address is learned. However, an input interface filter learns the MAC address before discarding the packet.

Applying a VPLS Filter to a VPLS Routing Instance

You can apply a VPLS filter to a VPLS routing instance. The filter checks traffic passing through the specified routing instance.

Input routing instance filters learn the MAC address before the filter action is completed, so if the filter action is **discard**, the MAC address is learned before the packet is dropped.

To apply a VPLS filter to packets arriving at a VPLS routing instance and specify the filter, include the **filter input** statement at the **[edit routing-instances *routing-instance-name* forwarding-options family vpls]** hierarchy level:

```
[edit routing-instances routing-instance-name forwarding-options family vpls]
filter input input-filter-name;
```

Configuring a Filter for Flooded Traffic

You can configure a VPLS filter to filter flooded packets. CE routers typically flood the following types of packets to PE routers in VPLS routing instances:

- Layer 2 broadcast packets
- Layer 2 multicast packets
- Layer 2 unicast packets with an unknown destination MAC address
- Layer 2 packets with a MAC entry in the DMAC routing table

You can configure filters to manage how these flooded packets are distributed to the other PE routers in the VPLS routing instance.

To apply a flooding filter to packets arriving at the PE router in the VPLS routing instance, and specify the filter, include the **flood input** statement:

```
flood input filter-name;
```

You can include this statement at the following hierarchy levels:

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

NOTE: ACX Series routers do not support the **[edit logical-systems]** hierarchy.

Configuring a VPLS Policer

You can configure a policer for VPLS traffic. The VPLS policer configuration is similar to the configuration of any other type of policer.

VPLS policers have the following characteristics:

- You cannot police the default VPLS routes stored in the flood table from PE router–sourced flood traffic.
- When specifying policing bandwidth, the VPLS policer considers all Layer 2 bytes in a packet to determine the packet length.

To configure a VPLS policer, include the **policer** statement at the **[edit firewall]** hierarchy level:

```
[edit firewall]
```

```
policer policer-name {
  bandwidth-limit limit;
  burst-size-limit limit;
  then action;
}
```

To apply a VPLS policer to an interface, include the **policer** statement:

```
policer {
  input input-policer-name;
  output output-policer-name;
}
```

You can include this statement at the following hierarchy levels:

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

NOTE: ACX Series routers do not support the **[edit logical-systems]** hierarchy.

In the **input** statement, list the name of the VPLS policer to be evaluated when packets are received on the interface. In the **output** statement, list the name of the VPLS policer to be evaluated when packets are transmitted on the interface. This type of VPLS policer can only apply to unicast packets. For information about how to filter flood packets, see "Configuring a Filter for Flooded Traffic" on page 89.

Related Documentation

- *Routing Policies, Firewall Filters, and Traffic Policers Feature Guide*
- Firewall Filter Match Conditions for VPLS Traffic on page 90

Firewall Filter Match Conditions for VPLS Traffic

In the **from** statement in the VPLS filter term, you specify conditions that the packet must match for the action in the **then** statement to be taken. All conditions in the **from** statement must match for the action to be taken. The order in which you specify match conditions is not important, because a packet must match all the conditions in a term for a match to occur.

If you specify no match conditions in a term, that term matches all packets.

An individual condition in a **from** statement can contain a list of values. For example, you can specify numeric ranges. You can also specify multiple source addresses or destination addresses. When a condition defines a list of values, a match occurs if one of the values in the list matches the packet.

Individual conditions in a **from** statement can be negated. When you negate a condition, you are defining an explicit mismatch. For example, the negated match condition for **forwarding-class** is **forwarding-class-except**. If a packet matches a negated condition, it

is immediately considered not to match the **from** statement, and the next term in the filter is evaluated, if there is one. If there are no more terms, the packet is discarded.

You can configure a firewall filter with match conditions for Virtual Private LAN Service (VPLS) traffic (**family vpls**). Table 5 on page 91 describes the ***match-conditions*** you can configure at the **[edit firewall family vpls filter *filter-name* term *term-name* from]** hierarchy level.

NOTE: Not all match conditions for VPLS traffic are supported on all routing platforms or switching platforms. A number of match conditions for VPLS traffic are supported only on MX Series 3D Universal Edge Routers.

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

Table 5: Firewall Filter Match Conditions for VPLS Traffic

Match Condition	Description
destination-mac-address *address*	Match the destination media access control (MAC) address of a VPLS packet.
destination-port *number*	(MX Series routers and EX Series switches only) Match the UDP or TCP destination port field. You cannot specify both the **port** and **destination-port** match conditions in the same term. In place of the numeric value, you can specify one of the following text synonyms (the port numbers are also listed): **afs** (1483), **bgp** (179), **biff** (512), **bootpc** (68), **bootps** (67), **cmd** (514), **cvspserver** (2401), **dhcp** (67), **domain** (53), **eklogin** (2105), **ekshell** (2106), **exec** (512), **finger** (79), **ftp** (21), **ftp-data** (20), **http** (80), **https** (443), **ident** (113), **imap** (143), **kerberos-sec** (88), **klogin** (543), **kpasswd** (761), **krb-prop** (754), **krbupdate** (760), **kshell** (544), **ldap** (389), **ldp** (646), **login** (513), **mobileip-agent** (434), **mobilip-mn** (435), **msdp** (639), **netbios-dgm** (138), **netbios-ns** (137), **netbios-ssn** (139), **nfsd** (2049), **nntp** (119), **ntalk** (518), **ntp** (123), **pop3** (110), **pptp** (1723), **printer** (515), **radacct** (1813), **radius** (1812), **rip** (520), **rkinit** (2108), **smtp** (25), **snmp** (161), **snmptrap** (162), **snpp** (444), **socks** (1080), **ssh** (22), **sunrpc** (111), **syslog** (514), **tacacs** (49), **tacacs-ds** (65), **talk** (517), **telnet** (23), **tftp** (69), **timed** (525), **who** (513), or **xdmcp** (177).
destination-port-except *number*	(MX Series routers and EX Series switches only) Do not match on the TCP or UDP destination port field. You cannot specify both the **port** and **destination-port** match conditions in the same term.
destination-prefix-list *name*	(MX Series routers and EX Series switches only) Match destination prefixes in the specified list. Specify the name of a prefix list defined at the **[edit policy-options prefix-list** *prefix-list-name*] hierarchy level. NOTE: VPLS prefix lists support only IPv4 addresses. IPv6 addresses included in a VPLS prefix list will be discarded.
destination-prefix-list *name* **except**	(MX Series routers and EX Series switches only) Do not match destination prefixes in the specified list. For more information, see the **destination-prefix-list** match condition.

Table 5: Firewall Filter Match Conditions for VPLS Traffic *(continued)*

Match Condition	Description
dscp *number*	(MX Series routers and EX Series switches only) Match the Differentiated Services code point (DSCP). The DiffServ protocol uses the type-of-service (ToS) byte in the IP header. The most significant 6 bits of this byte form the DSCP. For more information, see the *Understanding How Behavior Aggregate Classifiers Prioritize Trusted Traffic*. You can specify a numeric value from **0** through **63**. To specify the value in hexadecimal form, include **0x** as a prefix. To specify the value in binary form, include **b** as a prefix. In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed): • RFC 3246, *An Expedited Forwarding PHB (Per-Hop Behavior)*, defines one code point: **ef** (46). • RFC 2597, *Assured Forwarding PHB Group*, defines 4 classes, with 3 drop precedences in each class, for a total of 12 code points: **af11** (10), **af12** (12), **af13** (14), **af21** (18), **af22** (20), **af23** (22), **af31** (26), **af32** (28), **af33** (30), **af41** (34), **af42** (36), **af43** (38)
dscp-except *number*	(MX Series routers and EX Series switches only) Do not match on the DSCP. For details, see the **dscp** match condition.
ether-type *values*	Match the 2-octet IEEE 802.3 Length/EtherType field to the specified value or list of values. You can specify decimal or hexadecimal values from 0 through 65535 (0xFFFF). A value from 0 through 1500 (0x05DC) specifies the length of an Ethernet Version 1 frame. A value from 1536 (0x0600) through 65535 specifies the EtherType (nature of the MAC client protocol) of an Ethernet Version 2 frame. In place of the numeric value, you can specify one of the following text synonyms (the hexadecimal values are also listed): **aarp** (0x80F3), **appletalk** (0x809B), **arp** (0x0806), **ipv4** (0x0800), **ipv6** (0x86DD), **mpls-multicast** (0x8848), **mpls-unicast** (0x8847), **oam** (0x8902), **ppp** (0x880B), **pppoe-discovery** (0x8863), **pppoe-session** (0x8864), or **sna** (0x80D5).
ether-type-except *values*	Do not match the 2-octet Length/EtherType field to the specified value or list of values. For details about specifying the ***values***, see the **ether-type** match condition.

Table 5: Firewall Filter Match Conditions for VPLS Traffic *(continued)*

Match Condition	Description	
flexible-match-mask *value*	**bit-length**	Length of the data to be matched in bits, not needed for string input (0..128)
	bit-offset	Bit offset after the (match-start + byte) offset (0..7)
	byte-offset	Byte offset after the match start point
	flexible-mask-name	Select a flexible match from predefined template field
	mask-in-hex	Mask out bits in the packet data to be matched
	match-start	Start point to match in packet
	prefix	Value data/string to be matched
flexible-match-range *value*	**bit-length**	Length of the data to be matched in bits (0..32)
	bit-offset	Bit offset after the (match-start + byte) offset (0..7)
	byte-offset	Byte offset after the match start point
	flexible-range-name	Select a flexible match from predefined template field
	match-start	Start point to match in packet
	range	Range of values to be matched
	range-except	Do not match this range of values
forwarding-class *class*	Match the forwarding class. Specify **assured-forwarding**, **best-effort**, **expedited-forwarding**, or **network-control**.	
forwarding-class-except *class*	Do not match the forwarding class. For details, see the **forwarding-class** match condition.	

Table 5: Firewall Filter Match Conditions for VPLS Traffic *(continued)*

Match Condition	Description
icmp-code *message-code*	Match the ICMP message code field. If you configure this match condition, we recommend that you also configure the **next-header icmp** or **next-header icmp6** match condition in the same term. If you configure this match condition, you must also configure the **icmp-type** *message-type* match condition in the same term. An ICMP message code provides more specific information than an ICMP message type, but the meaning of an ICMP message code is dependent on the associated ICMP message type. In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed). The keywords are grouped by the ICMP type with which they are associated: • parameter-problem: **ip6-header-bad** (0), **unrecognized-next-header** (1), **unrecognized-option** (2) • time-exceeded: **ttl-eq-zero-during-reassembly** (1), **ttl-eq-zero-during-transit** (0) • destination-unreachable: **address-unreachable** (3), **administratively-prohibited** (1), **no-route-to-destination** (0), **port-unreachable** (4)
icmp-code-except *message-code*	Do not match the ICMP message code field. For details, see the **icmp-code** match condition.
icmp-code *number*	(MX Series routers and EX Series switches only) Match the ICMP message code field. If you configure this match condition, we recommend that you also configure the **ip-protocol icmp** or **ip-protocol icmp6** match condition in the same term. If you configure this match condition, you must also configure the **icmp-type** *message-type* match condition in the same term. An ICMP message code provides more specific information than an ICMP message type, but the meaning of an ICMP message code is dependent on the associated ICMP message type. In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed). The keywords are grouped by the ICMP type with which they are associated: • parameter-problem: **ip6-header-bad** (0), **unrecognized-next-header** (1), **unrecognized-option** (2) • time-exceeded: **ttl-eq-zero-during-reassembly** (1), **ttl-eq-zero-during-transit** (0) • destination-unreachable: **address-unreachable** (3), **administratively-prohibited** (1), **no-route-to-destination** (0), **port-unreachable** (4)
icmp-code-except *number*	(MX Series routers and EX Series switches only) Do not match on the ICMP code field. For details, see the **icmp-code** match condition.
icmp-type *number*	(MX Series routers and EX Series switches only) Match the ICMP message type field. If you configure this match condition, we recommend that you also configure the **ip-protocol icmp**, **ip-protocol icmp6**, or **ip-protocol icmpv6** match condition in the same term. In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed): **destination-unreachable** (1), **echo-reply** (129), **echo-request** (128), **membership-query** (130), **membership-report** (131), **membership-termination** (132), **neighbor-advertisement** (136), **neighbor-solicit** (135), **node-information-reply** (140), **node-information-request** (139), **packet-too-big** (2), **parameter-problem** (4), **redirect** (137), **router-advertisement** (134), **router-renumbering** (138), **router-solicit** (133), or **time-exceeded** (3).

Table 5: Firewall Filter Match Conditions for VPLS Traffic *(continued)*

Match Condition	Description
icmp-type-except *number*	(MX Series routers and EX Series switches only) Do not match the ICMP message type field. For details, see the **icmp-type** match condition.
interface *interface-name*	Interface on which the packet was received. You can configure a match condition that matches packets based on the interface on which they were received. NOTE: If you configure this match condition with an interface that does not exist, the term does not match any packet.
interface-group *group-number*	Match the logical interface on which the packet was received to the specified interface group or set of interface groups. For *group-number*, specify a single value or a range of values from **0** through **255**. To assign a logical interface to an interface group *group-number*, specify the *group-number* at the **[interfaces** *interface-name* **unit** *number* **family** *family* **filter group]** hierarchy level. For more information, see *Filtering Packets Received on a Set of Interface Groups Overview*. NOTE: This match condition is not supported on T4000 Type 5 FPCs.
interface-group-except *group-name*	Do not match the logical interface on which the packet was received to the specified interface group or set of interface groups. For details, see the **interface-group** match condition. NOTE: This match condition is not supported on T4000 Type 5 FPCs.
interface-set *interface-set-name*	Match the interface on which the packet was received to the specified interface set. To define an interface set, include the **interface-set** statement at the **[edit firewall]** hierarchy level. For more information, see *Filtering Packets Received on an Interface Set Overview*.
ip-address *address*	(MX Series routers and EX Series switches only) 32-bit address that supports the standard syntax for IPv4 addresses.
ip-destination-address *address*	(MX Series routers and EX Series switches only) 32-bit address that is the final destination node address for the packet.
ip-precedence *ip-precedence-field*	(MX Series routers and EX Series switches only) IP precedence field. In place of the numeric field value, you can specify one of the following text synonyms (the field values are also listed): **critical-ecp** (0xa0), **flash** (0x60), **flash-override** (0x80), **immediate** (0x40), **internet-control** (0xc0), **net-control** (0xe0), **priority** (0x20), or **routine** (0x00).
ip-precedence-except *ip-precedence-field*	(MX Series routers and EX Series switches only) Do not match on the IP precedence field.
ip-protocol *number*	(MX Series routers and EX Series switches only) IP protocol field.
ipv6-address *address*	(MX Series only) 128-bit address that supports the standard syntax for IPv6 addresses.
ip-protocol-except *number*	(MX Series routers and EX Series switches only) Do not match on the IP protocol field.
ipv6-destination-address *address*	(MX Series only) 128-bit address that is the final destination node address for this packet.

Table 5: Firewall Filter Match Conditions for VPLS Traffic *(continued)*

Match Condition	Description
ipv6-destination-prefix-list *named-list*	(MX Series only) Match the IPv6 destination addresses in a *named-list*.
ipv6-next-header *protocol*	(MX Series only) Match IPv6 next header protocol type. The following list shows the supported values for *protocol*: • **ah**—IP Security authentication header • **dstopts**—IPv6 destination options • **egp**—Exterior gateway protocol • **esp**—IPSec Encapsulating Security Payload • **fragment**—IPv6 fragment header • **gre**—Generic routing encapsulation • **hop-by-hop**—IPv6 hop by hop options • **icmp**—Internet Control Message Protocol • **icmp6**—Internet Control Message Protocol Version 6 • **igmp**—Internet Group Management Protocol • **ipip**—IP in IP • **ipv6**—IPv6 in IP • **no-next-header**—IPv6 no next header • **ospf**—Open Shortest Path First • **pim**—Protocol Independent Multicast • **routing**—IPv6 routing header • **rsvp**—Resource Reservation Protocol • **sctp**—Stream Control Transmission Protocol • **tcp**—Transmission Control Protocol • **udp**—User Datagram Protocol • **vrrp**—Virtual Router Redundancy Protocol
ipv6-next-header-except *protocol*	(MX Series only) Do not match the IPv6 next header protocol type.

Table 5: Firewall Filter Match Conditions for VPLS Traffic *(continued)*

Match Condition	Description
ipv6-payload-protocol *protocol*	(MX Series only) Match IPv6 payload protocol type. The following list shows the supported values for *protocol*: • **ah**—IP Security authentication header • **dstopts**—IPv6 destination options • **egp**—Exterior gateway protocol • **esp**—IPSec Encapsulating Security Payload • **fragment**—IPv6 fragment header • **gre**—Generic routing encapsulation • **hop-by-hop**—IPv6 hop by hop options • **icmp**—Internet Control Message Protocol • **icmp6**—Internet Control Message Protocol Version 6 • **igmp**—Internet Group Management Protocol • **ipip**—IP in IP • **ipv6**—IPv6 in IP • **no-next-header**—IPv6 no next header • **ospf**—Open Shortest Path First • **pim**—Protocol Independent Multicast • **routing**—IPv6 routing header • **rsvp**—Resource Reservation Protocol • **sctp**—Stream Control Transmission Protocol • **tcp**—Transmission Control Protocol • **udp**—User Datagram Protocol • **vrrp**—Virtual Router Redundancy Protocol
ipv6-payload-protocol-except *protocol*	(MX Series only) Do not match the IPv6 payload protocol.
ipv6-prefix-list *named-list*	(MX Series only) Match the IPv6 address in a *named-list*.
ipv6-source-address *address*	(MX Series only) 128-bit address that is the originating source node address for this packet.
ipv6-source-prefix-list *named-list*	(MX Series only) Match the IPv6 source address in a *named-list*.

Table 5: Firewall Filter Match Conditions for VPLS Traffic *(continued)*

Match Condition	Description
ipv6-traffic-class *number*	(MX Series only) Differentiated Services code point (DSCP). The DiffServ protocol uses the type-of-service (ToS) byte in the IP header. The most significant 6 bits of this byte form the DSCP. For more information, see *Understanding How Behavior Aggregate Classifiers Prioritize Trusted Traffic*. You can specify a numeric value from **0** through **63**. To specify the value in hexadecimal form, include **0x** as a prefix. To specify the value in binary form, include **b** as a prefix. In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed): • RFC 3246, *An Expedited Forwarding PHB (Per-Hop Behavior)*, defines one code point: **ef** (46). • RFC 2597, *Assured Forwarding PHB Group*, defines 4 classes, with 3 drop precedences in each class, for a total of 12 code points: **af11** (10), **af12** (12), **af13** (14), **af21** (18), **af22** (20), **af23** (22), **af31** (26), **af32** (28), **af33** (30), **af41** (34), **af42** (36), **af43** (38)
ipv6-traffic-class-except *number*	Do not match the DSCP **number**.
ip-source-address *address*	(MX Series routers and EX Series switches only) IP address of the source node sending the packet.
learn-vlan-1p-priority *number*	(MX Series routers, M320 router, and EX Series switches only) Match on the IEEE 802.1p learned VLAN priority bits in the provider VLAN tag (the only tag in a single-tag frame with 802.1Q VLAN tags or the outer tag in a dual-tag frame with 802.1Q VLAN tags). Specify a single value or multiple values from **0** through **7**. Compare with the **user-vlan-1p-priority** match condition. NOTE: This match condition supports the presence of a control word for MX Series routers and the M320 router.
learn-vlan-1p-priority-except *number*	(MX Series routers, M320 router, and EX Series switches only) Do not match on the IEEE 802.1p learned VLAN priority bits. For details, see the **learn-vlan-1p-priority** match condition. NOTE: This match condition supports the presence of a control word for MX Series routers and the M320 router.
learn-vlan-dei	(MX Series routers and EX Series switches only) Match the user VLAN ID drop eligability indicator (DEI) bit.
learn-vlan-dei-except	(MX Series routers and EX Series switches only) Do not match the user VLAN ID DEI bit.
learn-vlan-id *number*	(MX Series routers and EX Series switches only) VLAN identifier used for MAC learning.
learn-vlan-id-except *number*	(MX Series routers and EX Series switches only) Do not match on the VLAN identifier used for MAC learning.

Table 5: Firewall Filter Match Conditions for VPLS Traffic *(continued)*

Match Condition	Description
loss-priority *level*	Packet loss priority (PLP) level. Specify a single level or multiple levels: **low**, **medium-low**, **medium-high**, or **high**. Supported on M120 and M320 routers; M7i and M10i routers with the Enhanced CFEB (CFEB-E); and MX Series routers. For IP traffic on M320, MX Series, and T Series routers with Enhanced II Flexible PIC Concentrators (FPCs) and EX Series switches, you must include the **tri-color** statement at the **[edit class-of-service]** hierarchy level to commit a PLP configuration with any of the four levels specified. If the **tri-color** statement is not enabled, you can only configure the **high** and **low** levels. This applies to all protocol families. For information about the **tri-color** statement and about using behavior aggregate (BA) classifiers to set the PLP level of incoming packets, see *Understanding How Forwarding Classes Assign Classes to Output Queues*.
loss-priority-except *level*	Do not match on the packet loss priority level. Specify a single level or multiple levels: **low**, **medium-low**, **medium-high**, or **high**. For information about using behavior aggregate (BA) classifiers to set the PLP level of incoming packets, see *Understanding How Behavior Aggregate Classifiers Prioritize Trusted Traffic*.
port *number*	(MX Series routers and EX Series switches only) TCP or UDP source or destination port. You cannot specify both the **port** match condition and either the **destination-port** or **source-port** match condition in the same term.
port-except *number*	(MX Series routers and EX Series switches only) Do not match on the TCP or UDP source or destination port. You cannot specify both the **port** match condition and either the **destination-port** or **source-port** match condition in the same term.
prefix-list *name*	(MX Series routers and EX Series switches only) Match the destination or source prefixes in the specified list. Specify the name of a prefix list defined at the **[edit policy-options prefix-list** *prefix-list-name*] hierarchy level. NOTE: VPLS prefix lists support only IPV4 addresses. IPV6 addresses included in a VPLS prefix list will be discarded.
prefix-list *name* **except**	(MX Series routers and EX Series switches only) Do not match the destination or source prefixes in the specified list. For more information, see the **destination-prefix-list** match condition.
source-mac-address *address*	Source MAC address of a VPLS packet.
source-port *number*	(MX Series routers and EX Series switches only) TCP or UDP source port field. You cannot specify the **port** and **source-port** match conditions in the same term.
source-port-except *number*	(MX Series routers and EX Series switches only) Do not match on the TCP or UDP source port field. You cannot specify the **port** and **source-port** match conditions in the same term.

Table 5: Firewall Filter Match Conditions for VPLS Traffic *(continued)*

Match Condition	Description
source-prefix-list *name*	(MX Series routers and EX Series switches only) Match the source prefixes in the specified prefix list. Specify a prefix list name defined at the **[edit policy-options prefix-list** *prefix-list-name***]** hierarchy level. NOTE: VPLS prefix lists support only IPV4 addresses. IPV6 addresses included in a VPLS prefix list will be discarded.
source-prefix-list *name* **except**	(MX Series routers and EX Series switches only) Do not match the source prefixes in the specified prefix list. For more information, see the **source-prefix-list** match condition.
tcp-flags *flags*	Match one or more of the low-order 6 bits in the 8-bit TCP flags field in the TCP header. To specify individual bit fields, you can specify the following text synonyms or hexadecimal values: • **fin** (0x01) • **syn** (0x02) • **rst** (0x04) • **push** (0x08) • **ack** (0x10) • **urgent** (0x20) In a TCP session, the SYN flag is set only in the initial packet sent, while the ACK flag is set in all packets sent after the initial packet. You can string together multiple flags using the bit-field logical operators. If you configure this match condition for IPv6 traffic, we recommend that you also configure the **next-header tcp** match condition in the same term to specify that the TCP protocol is being used on the port.
traffic-type *type-name*	(MX Series routers and EX Series switches only) Traffic type. Specify **broadcast**, **multicast**, **unknown-unicast**, or **known-unicast**.
traffic-type-except *type-name*	(MX Series routers and EX Series switches only) Do not match on the traffic type. Specify **broadcast**, **multicast**, **unknown-unicast**, or **known-unicast**.
user-vlan-1p-priority *number*	(MX Series routers, M320 router, and EX Series switches only) Match on the IEEE 802.1p user priority bits in the customer VLAN tag (the inner tag in a dual-tag frame with 802.1Q VLAN tags). Specify a single value or multiple values from **0** through **7**. Compare with the **learn-vlan-1p-priority** match condition. NOTE: This match condition supports the presence of a control word for MX Series routers and the M320 router.
user-vlan-1p-priority-except *number*	(MX Series routers, M320 rouer, and EX Series switches only) Do not match on the IEEE 802.1p user priority bits. For details, see the **user-vlan-1p-priority** match condition. NOTE: This match condition supports the presence of a control word for MX Series routers and the M320 router.
user-vlan-id *number*	(MX Series routers and EX Series switches only) Match the first VLAN identifier that is part of the payload.

Table 5: Firewall Filter Match Conditions for VPLS Traffic *(continued)*

Match Condition	Description
user-vlan-id-except *number*	(MX Series routers and EX Series switches only) Do not match on the first VLAN identifier that is part of the payload.
vlan-ether-type *value*	VLAN Ethernet type field of a VPLS packet.
vlan-ether-type-except *value*	Do not match on the VLAN Ethernet type field of a VPLS packet.

Related Documentation

- *Guidelines for Configuring Firewall Filters*
- *Firewall Filter Terminating Actions*
- *Firewall Filter Nonterminating Actions*

CHAPTER 12

Monitoring and Tracing VPLS

- Tracing VPLS Traffic and Operations on page 103

Tracing VPLS Traffic and Operations

To trace VPLS traffic, include the **traceoptions** statement:

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

You can include this statement at the following hierarchy levels:

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

The following trace flags display the operations associated with VPLS:

- **all**—All VPLS tracing options
- **connections**—VPLS connections (events and state changes)
- **error**—Error conditions
- **nlri**—VPLS advertisements received or sent using BGP
- **route**—Trace-routing information
- **topology**—VPLS topology changes caused by reconsideration or advertisements received from other PE routers using BGP

PART 3

Configuration Statements and Operational Commands

- Configuration Statements on page 107
- Operational Commands on page 169

CHAPTER 13

Configuration Statements

- active-interface (VPLS Multihoming) on page 109
- any (VPLS Multihoming) on page 110
- automatic-site-id on page 111
- best-site on page 112
- bfd-liveness-detection (Layer 2 VPN and VPLS) on page 113
- connectivity-type on page 114
- encapsulation (Physical Interface) on page 115
- encapsulation-type (Layer 2 VPNs) on page 121
- family multiservice on page 123
- fast-reroute-priority on page 126
- identifier (VPLS Multihoming for FEC 129) on page 127
- interface (Routing Instances) on page 128
- interface (VPLS Multihoming for FEC 129) on page 129
- interface (VPLS Routing Instances) on page 130
- interface-mac-limit (VPLS) on page 131
- l2vpn-id on page 132
- label-block-size on page 133
- label-switched-path-template (Multicast) on page 134
- local-switching (VPLS) on page 135
- mac-flush on page 136
- mac-table-aging-time on page 138
- mac-table-size on page 139
- mesh-group (Protocols VPLS) on page 140
- multi-homing (VPLS Multihoming for FEC 128) on page 141
- multi-homing (VPLS Multihoming for FEC 129) on page 142
- neighbor (Protocols VPLS) on page 143
- no-tunnel-services on page 145
- peer-active (VPLS Multihoming for FEC 129) on page 146

- peer-as (VPLS) on page 147
- ping-interval on page 148
- preference (Interface-Level Preference for VPLS Multihoming for FEC 129) on page 149
- preference (Site-Level Preference for VPLS Multihoming for FEC 129) on page 150
- primary (VPLS Multihoming) on page 151
- rsvp-te (Routing Instances Provider Tunnel) on page 152
- site (VPLS Multihoming for FEC 128) on page 153
- site (VPLS Multihoming for FEC 129) on page 154
- site-identifier (VPLS) on page 155
- site-preference on page 156
- site-range on page 157
- static (Protocols VPLS) on page 158
- template on page 159
- traceoptions (Protocols VPLS) on page 160
- tunnel-services (Routing Instances VPLS) on page 162
- vlan-id on page 163
- vlan-id-list (Interface in VPLS) on page 163
- vlan-tagging on page 164
- vpls (Interfaces) on page 165
- vpls (Routing Instance) on page 166
- vpls-id on page 168

active-interface (VPLS Multihoming)

Syntax

```
active-interface {
  any;
  primary interface-name;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols vpls multi-homing],
[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols vpls multi-homing site *site-name*],
[edit routing-instances *instance-name* protocols vpls multi-homing],
[edit routing-instances *instance-name* protocols vpls multi-homing site *site-name*]

Release Information

Statement introduced in Junos OS Release 7.5.

Description

Specify a multihomed interface as the primary interface for the VPLS site. If there are multiple interfaces, the remaining interfaces are activated only when the primary interface goes down. If no active interfaces are configured at the site level, it is assumed that all traffic for a VPLS site travels through a single, nonmultihomed PE router.

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

The remaining statements are explained separately.

For FEC 128, use the **[edit routing-instances *instance-name* protocols vpls multi-homing]** hierarchy level.

Required Privilege Level

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

Related Documentation

- Specifying an Interface as the Active Interface on page 68

any (VPLS Multihoming)

Syntax any;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *instance-name* protocols vpls multi-homing active-interface],
[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols vpls multi-homing site *site-name* active-interface],
[edit routing-instances *instance-name* protocols vpls multi-homing active-interface],
[edit routing-instances *instance-name* protocols vpls multi-homing site *site-name* active-interface]

Release Information Statement introduced in Junos OS Release 7.5.

Description Specify that any multihomed interface can be used as the primary interface by the VPLS site. Depending on the order in which interfaces are listed in the PE router's configuration, the first operational interface in the set of configured interfaces is chosen to be the primary interface.

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

For FEC 128, use the **[edit routing-instances *instance-name* protocols vpls multi-homing active-interface]** hierarchy level.

Default This is the default behavior.

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

Related Documentation

- Specifying an Interface as the Active Interface on page 68
- primary on page 151

automatic-site-id

Syntax

```
automatic-site-id {
  collision-detect-time seconds;
  new-site-wait-time seconds;
  reclaim-wait-time minimum seconds maximum seconds;
  startup-wait-time seconds;
}
```

Hierarchy Level

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

Release Information

Statement introduced in Junos OS Release 9.1.

Description

Enable automatic site identifiers for VPLS routing instances.

When you configure **automatic-site-id** for the first time, you must deactivate and then activate **protocol vpls**. However, if you already have **automatic-site-id** configured, you do not need to deactivate and then activate **protocol vpls**.

Options

collision-detect-time—The time in seconds to wait after a claim advertisement is sent to the other routers in a VPLS instance before a PE router can begin using a site identifier. If the PE router receives a competing claim advertisement for the same site identifier during this time period, it initiates the collision resolution procedure for site identifiers.

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

new-site-wait-time—The time in seconds to wait to receive VPLS information for a newly configured routing instance or a new site. This time interval is also applied whenever the automatic site identifier feature is activated on a VPLS routing instance other than at startup. Effectively, this timer indicates how long to wait before an attempt is made to allocate a site identifier. This timer is also triggered whenever a VPLS routing instance is enabled.

reclaim-wait-time—The time in seconds to wait to receive VPLS information for a newly configured routing instance or a new site. This time interval is also applied whenever the automatic site identifier feature is activated on a VPLS routing instance other than at startup. Effectively, this timer indicates how long to wait before an attempt is made to allocate a site identifier. This timer is also triggered whenever a VPLS routing instance is enabled. You can configure two values for this option: the **minimum** wait time and the **maximum** wait time.

startup-wait-time—The time in seconds to wait at startup to receive all the VPLS information for the route targets configured on the other PE routers included in the VPLS routing instance.

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

Related Documentation
- Configuring Automatic Site Identifiers for VPLS on page 24

best-site

Syntax
best-site;

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

Release Information
Statement introduced in Junos OS Release 12.2.

Description
Enables the VPLS multihoming best site functionality, allowing the site on which it has been enabled to be the preferred site for this PE router. This statement must be configured on all PE routers within the optimized VPLS routing instance.

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

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

Related Documentation
- *Example: VPLS Multihoming, Improved Convergence Time*

bfd-liveness-detection (Layer 2 VPN and VPLS)

Syntax

```
bfd-liveness-detection {
  detection-time {
    threshold milliseconds;
  }
  minimum-interval milliseconds;
  minimum-receive-interval milliseconds;
  multiplier number;
  no-adaptation;
  transmit-interval {
    minimum-interval milliseconds;
    threshold milliseconds;
  }
  version (1 | automatic);
}
```

Hierarchy Level

[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn oam],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls neighbor *neighbor-id* oam],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name* neighbor *neighbor-id* oam],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls oam],
[edit routing-instances *routing-instance-name* protocols l2vpn oam],
[edit routing-instances *routing-instance-name* protocols vpls neighbor *neighbor-id* oam],
[edit routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name* neighbor *neighbor-id* oam],
[edit routing-instances *routing-instance-name* protocols vpls oam]

Release Information

Statement introduced in Junos OS Release 10.0.
Statement introduced in Junos OS Release 13.2 for Layer 2 VPNs and VPLS.

Description

Configure bidirectional failure detection timers.

The BFD failure detection timers are adaptive and can be adjusted to be more or less aggressive. For example, the timers can adapt to a higher value if the adjacency fails, or a neighbor can negotiate a higher value for a timer than the configured value. The timers adapt to a higher value when a BFD session flap occurs more than three times in a span of 15 seconds. A back-off algorithm increases the receive (Rx) interval by two if the local BFD instance is the reason for the session flap. The transmission (Tx) interval is increased by two if the remote BFD instance is the reason for the session flap. You can use the **clear bfd adaptation** command to return BFD interval timers to their configured values. The **clear bfd adaptation** command is hitless, meaning that the command does not affect traffic flow on the routing device.

The remaining statements are explained separately.

Required Privilege Level

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

Related Documentation

- *Configuring BFD for Layer 2 VPN and VPLS*
- *Example: Configuring BFD for Static Routes for Faster Network Failure Detection*

connectivity-type

Syntax connectivity-type (ce | irb | permanent);

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

Release Information Statement introduced in Junos OS Release 9.1.
irb option introduced in Junos OS Release 9.3.
permanent option introduced in Junos OS Release 10.4.

Description Specify when a VPLS connection is taken down depending on whether or not the interface for the VPLS routing instance is customer-facing or integrated routing and bridging (IRB).

NOTE: The **connectivity-type** statement is not supported for FEC 129 VPLS (also known as LDP VPLS with BGP-based autodiscovery).

Default ce

Options

ce—Require that for the VPLS connection to be up, the customer-facing interface for the VPLS routing instance must also be up. If the customer-facing interface fails, the VPLS connection is taken down.

irb—Allow a VPLS connection to remain up so long as an IRB interface is configured for the VPLS routing instance.

permanent—Allow a VPLS connection to remain up until specifically taken down. This option is reserved for use in configuring Layer 2 Wholesale subscriber networks. See the *Broadband Subscriber Management Solutions Guide* for details about configuring a Layer 2 Wholesale network.

NOTE: To specifically take down a VPLS routing instance that is using the **permanent** option, all associated static logical interfaces must also be down.

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

Related Documentation

- Configuring VPLS Routing Instance and VPLS Interface Connectivity on page 31
- *Configuring Separate NNI Routing Instances for Layer 2 Wholesale Service Retailers*

encapsulation (Physical Interface)

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

Hierarchy Level [edit interfaces *interface-name*],
[edit interfaces rlsq *number:number*]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 11.1 for EX Series switches.
Statement introduced in Junos OS Release 12.1X48 for PTX Series Packet Transport Routers (**flexible-ethernet-services**, **ethernet-ccc**, and **ethernet-tcc** options only).

Description Specify the physical link-layer encapsulation type. Not all encapsulation types are supported on the switches. See the switch CLI.

Default **ppp**—Use serial PPP encapsulation.

Options

NOTE: Frame Relay, ATM, PPP, SONET, and SATSOP options are not supported on the EX Series switches.

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

atm-pvc—Defined in RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*. When you configure physical ATM interfaces with ATM PVC encapsulation, an RFC 2684-compliant ATM Adaptation Layer 5 (AAL5) tunnel is set up to route the ATM cells over a Multiprotocol Label Switching (MPLS) path that is typically established between two MPLS-capable routers using the Label Distribution Protocol (LDP).

cisco-hdlc—Use Cisco-compatible High-Level Data Link Control (HDLC) framing. E1, E3, SONET/SDH, T1, and T3 interfaces can use Cisco HDLC encapsulation. Two related versions are supported:

- CCC version (**cisco-hdlc-ccc**)—The logical interface does not require an encapsulation statement. When you use this encapsulation type, you can configure the **ccc** family only.
- TCC version (**cisco-hdlc-tcc**)—Similar to CCC and has the same configuration restrictions, but used for circuits with different media on either side of the connection.

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

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

ethernet-bridge—Use Ethernet bridge encapsulation on Ethernet interfaces that have bridging enabled and that must accept all packets.

ethernet-ccc—Use Ethernet CCC encapsulation on Ethernet interfaces that must accept packets carrying standard Tag Protocol ID (TPID) values. For 8-port, 12-port, and 48-port Fast Ethernet PICs, CCC is not supported.

ethernet-over-atm—For interfaces that carry IPv4 traffic, use Ethernet over ATM encapsulation. When you use this encapsulation type, you cannot configure multipoint interfaces. As defined in RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*, this encapsulation type allows ATM interfaces to connect to devices that support only bridge protocol data units (BPDUs). Junos OS does not completely support bridging, but accepts BPDU packets as a default gateway. If you use the router as an edge device, then the router acts as a default gateway. It accepts Ethernet LLC/SNAP frames with IP or ARP in the payload, and drops the rest. For packets destined to the Ethernet LAN, a route lookup is done using the destination IP address. If the route lookup yields a full address match, the packet is encapsulated with an LLC/SNAP and MAC header, and the packet is forwarded to the ATM interface.

ethernet-tcc—For interfaces that carry IPv4 traffic, use Ethernet TCC encapsulation on interfaces that must accept packets carrying standard TPID values. For 8-port, 12-port, and 48-port Fast Ethernet PICs, TCC is not supported.

ethernet-vpls—Use Ethernet VPLS encapsulation on Ethernet interfaces that have VPLS enabled and that must accept packets carrying standard TPID values. On M Series routers, except the M320 router, the 4-port Fast Ethernet TX PIC and the 1-port, 2-port, and 4-port, 4-slot Gigabit Ethernet PICs can use the Ethernet VPLS encapsulation type.

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 the MAC address 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.

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

extended-frame-relay-ccc—Use Frame Relay encapsulation on CCC circuits. This encapsulation type allows you to dedicate DLCIs 1 through 1022 to CCC. When you use this encapsulation type, you can configure the **ccc** family only.

extended-frame-relay-ether-type-tcc—Use extended Frame Relay ether type TCC for Cisco-compatible Frame Relay for DLCIs 1 through 1022. This encapsulation type is used for circuits with different media on either side of the connection.

extended-frame-relay-tcc—Use Frame Relay encapsulation on TCC circuits to connect different media. This encapsulation type allows you to dedicate DLCIs 1 through 1022 to TCC.

extended-vlan-bridge—Use extended VLAN bridge encapsulation on Ethernet interfaces that have IEEE 802.1Q VLAN tagging and bridging enabled and that must accept packets carrying TPID 0x8100 or a user-defined TPID.

extended-vlan-ccc—Use extended VLAN encapsulation on CCC circuits with Gigabit Ethernet and 4-port Fast Ethernet interfaces that must accept packets carrying 802.1Q values. Extended VLAN CCC encapsulation supports TPIDs 0x8100, 0x9100, and 0x9901. When you use this encapsulation type, you can configure the **ccc** family only. For 8-port, 12-port, and 48-port Fast Ethernet PICs, extended VLAN CCC is not supported. For 4-port Gigabit Ethernet PICs, extended VLAN CCC is not supported.

extended-vlan-tcc—For interfaces that carry IPv4 traffic, use extended VLAN encapsulation on TCC circuits with Gigabit Ethernet interfaces on which you want to use 802.1Q tagging. For 4-port Gigabit Ethernet PICs, extended VLAN TCC is not supported.

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. On M Series routers, except the M320 router, the 4-port Fast Ethernet TX PIC and the 1-port, 2-port, and 4-port, 4-slot Gigabit Ethernet PICs can use the Ethernet VPLS encapsulation type.

NOTE: **The built-in Gigabit Ethernet PIC on an M7i router does not support extended VLAN VPLS encapsulation.**

flexible-ethernet-services—For Gigabit Ethernet IQ interfaces and Gigabit Ethernet PICs with small form-factor pluggable transceivers (SFPs) (except the 10-port Gigabit Ethernet PIC and the built-in Gigabit Ethernet port on the M7i router), use flexible Ethernet services encapsulation when you want to configure multiple per-unit Ethernet encapsulations. Aggregated Ethernet bundles can use this encapsulation type. This encapsulation type allows you to configure any combination of route, TCC, CCC, Layer 2 virtual private networks (VPNs), and VPLS encapsulations on a single physical port. If you configure flexible Ethernet services encapsulation on the physical interface, VLAN IDs from 1 through 511 are no longer reserved for normal VLANs.

flexible-frame-relay—For IQ interfaces only, use flexible Frame Relay encapsulation when you want to configure multiple per-unit Frame Relay encapsulations. This encapsulation type allows you to configure any combination of TCC, CCC, and standard Frame Relay encapsulations on a single physical port. Also, each logical interface can have any DLCI value from 1 through 1022.

frame-relay—Use Frame Relay encapsulation is defined in RFC 1490, *Multiprotocol Interconnect over Frame Relay*. E1, E3, link services, SONET/SDH, T1, T3, and voice services interfaces can use Frame Relay encapsulation.

frame-relay-ccc—Use Frame Relay encapsulation on CCC circuits. This encapsulation is same as standard Frame Relay for DLCIs 0 through 511. DLCIs 512 through 1022 are dedicated to CCC. The logical interface must also have **frame-relay-ccc** encapsulation. When you use this encapsulation type, you can configure the **ccc** family only.

frame-relay-ether-type—Use Frame Relay ether type encapsulation for compatibility with the Cisco Frame Relay. IETF frame relay encapsulation identifies the payload format using NLPID and SNAP formats. Cisco-compatible Frame Relay encapsulation uses the Ethernet type to identify the type of payload.

NOTE: When the encapsulation type is set to Cisco-compatible Frame Relay encapsulation, ensure that the LMI type is set to ANSI or Q933-A.

frame-relay-ether-type-tcc—Use Frame Relay ether type TCC for Cisco-compatible Frame Relay on TCC circuits to connect different media. This encapsulation is Cisco-compatible Frame Relay for DLCIs 0 through 511. DLCIs 512 through 1022 are dedicated to TCC.

frame-relay-port-ccc—Use Frame Relay port CCC encapsulation to transparently carry all the DLCIs between two customer edge (CE) routers without explicitly configuring each DLCI on the two provider edge (PE) routers with Frame Relay transport. The connection between the two CE routers can be either user-to-network interface (UNI) or network-to-network interface (NNI); this is completely transparent to the PE routers. When you use this encapsulation type, you can configure the **ccc** family only.

frame-relay-tcc—This encapsulation is similar to Frame Relay CCC and has the same configuration restrictions, but used for circuits with different media on either side of the connection.

generic-services—Use generic services encapsulation for services with a hierarchical scheduler.

multilink-frame-relay-uni-nni—Use MLFR UNI NNI encapsulation. This encapsulation is used on link services, voice services interfaces functioning as FRF.16 bundles, and their constituent T1 or E1 interfaces, and is supported on LSQ and redundant LSQ interfaces.

ppp—Use serial PPP encapsulation. This encapsulation is defined in RFC 1661, *The Point-to-Point Protocol (PPP) for the Transmission of Multiprotocol Datagrams over Point-to-Point Links*. PPP is the default encapsulation type for physical interfaces. E1, E3, SONET/SDH, T1, and T3 interfaces can use PPP encapsulation.

ppp-ccc—Use serial PPP encapsulation on CCC circuits. When you use this encapsulation type, you can configure the **ccc** family only.

ppp-tcc—Use serial PPP encapsulation on TCC circuits for connecting different media. When you use this encapsulation type, you can configure the **tcc** family only.

vlan-ccc—Use Ethernet VLAN encapsulation on CCC circuits. VLAN CCC encapsulation supports TPID 0x8100 only. When you use this encapsulation type, you can configure the **ccc** family only.

vlan-vci-ccc—Use ATM-to-Ethernet interworking encapsulation on CCC circuits. When you use this encapsulation type, you can configure the **ccc** family only. All logical interfaces configured on the Ethernet interface must also have the encapsulation type set to **vlan-vci-ccc**.

vlan-vpls—Use VLAN VPLS encapsulation on Ethernet interfaces with VLAN tagging and VPLS enabled. Interfaces with VLAN VPLS encapsulation accept packets carrying standard TPID values only. On M Series routers, except the M320 router, the 4-port Fast Ethernet TX PIC and the 1-port, 2-port, and 4-port, 4-slot Gigabit Ethernet PICs can use the Ethernet VPLS encapsulation type.

NOTE:

- Label-switched interfaces (LSIs) do not support VLAN VPLS encapsulation. Therefore, you can only use VLAN VPLS encapsulation on a PE-router-to-CE-router interface and not a core-facing interface.
- Starting with Junos OS release 13.3, a commit error occurs when you configure **vlan-vpls** encapsulation on a physical interface and configure **family inet** on one of the logical units. Previously, it was possible to commit this invalid configuration.

Required Privilege Level

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

Related Documentation

- *Configuring Interface Encapsulation on Physical Interfaces*
- *Configuring CCC Encapsulation for Layer 2 VPNs*
- *Configuring Layer 2 Switching Cross-Connects Using CCC*
- *Configuring TCC Encapsulation for Layer 2 VPNs and Layer 2 Circuits*
- *Configuring ATM Interface Encapsulation*
- *Configuring ATM-to-Ethernet Interworking*
- *Configuring VLAN and Extended VLAN Encapsulation*
- *Configuring VLAN and Extended VLAN Encapsulation*
- *Configuring Encapsulation for Layer 2 Wholesale VLAN Interfaces*
- *Configuring Interfaces for Layer 2 Circuits*
- *Configuring Interface Encapsulation on PTX Series Packet Transport Routers*
- *Configuring MPLS LSP Tunnel Cross-Connects Using CCC*
- *Configuring TCC*
- Configuring VPLS Interface Encapsulation on page 48
- Configuring Interfaces for VPLS Routing on page 47
- *Defining the Encapsulation for Switching Cross-Connects*

encapsulation-type (Layer 2 VPNs)

Syntax encapsulation-type (atm-aal5 | atm-cell | atm-cell-port-mode | atm-cell-vc-mode | atm-cell-vp-mode | cesop | cisco-hdlc | ethernet | ethernet-vlan | frame-relay | frame-relay-port-mode | interworking | ppp | satop-e1 | satop-e3 | satop-t1 | satop-t3);

Hierarchy Level
[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls],
[edit protocols l2circuit neighbor *address* interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols l2vpn],
[edit routing-instances *routing-instance-name* protocols l2vpn neighbor *address*],
[edit routing-instances *routing-instance-name* protocols vpls],
[edit routing-instances *routing-instance-name* protocols vpls neighbor *address*]

Release Information
Statement introduced in Junos OS Release 9.2.
Statement introduced in Junos OS Release 11.1 for EX Series switches.

Description Specify the type of Layer 2 traffic originating from the CE device. Only the **ethernet** and **ethernet-vlan** encapsulation types are supported for VPLS. Not all encapsulation types are supported on the switches. See the switch CLI.

Options
atm-aal5—ATM Adaptation Layer (AAL/5)

atm-cell—ATM cell relay

atm-cell-port-mode—ATM cell relay port promiscuous mode

atm-cell-vc-mode—ATM VC cell relay nonpromiscuous mode

atm-cell-vp-mode—ATM virtual path (VP) cell relay promiscuous mode

cesop—CESOP-based Layer 2 VPN

cisco-hdlc—Cisco Systems–compatible HDLC

ethernet—Ethernet

ethernet-vlan—Ethernet VLAN

frame-relay—Frame Relay

frame-relay-port-mode—Frame Relay port mode

interworking—Layer 2.5 interworking VPN

ppp—PPP

satsop-e1—SATSOP-E1–based Layer 2 VPN

satsop-e3—SATSOP-E3–based Layer 2 VPN

satsop-t1—SATSOP-T1–based Layer 2 VPN

satsop-t3—SATSOP-T3–based Layer 2 VPN

Default: For VPLS networks, the default encapsulation type is **ethernet**.

Required Privilege Level

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

Related Documentation

- *Configuring the Local Site on PE Routers in Layer 2 VPNs*
- Configuring VPLS Routing Instances on page 21
- *Configuring Interfaces for Layer 2 Circuits*
- *Configuring an MPLS-Based Layer 2 VPN (CLI Procedure)*

family multiservice

Syntax

```
family multiservice {
  destination-mac;
  label-1;
  label-2;
  payload {
    ip {
      layer-3 {
        (source-ip-only | destination-ip-only);
      }
      layer-3-only;
      layer-4;
    }
  }
  source-mac;
  symmetric-hash {
    complement;
  }
}
```

Hierarchy Level [edit forwarding-options hash-key]

Release Information Statement introduced in Junos OS Release 8.0.
ip, **label-1**, **label-2**, **layer-3-only**, and **payload** options introduced in Junos OS Release 9.4.
layer-3, **layer-**. **source-ip-only**, and **destination-ip-only** options introduced in Junos OS Release 9.5.
symmetric-hash and **complement** options introduced in Junos OS Release 9.6.

Description Configure load balancing based on Layer 2 media access control information. On MX Series routers, configure VPLS load balancing. On M120 and M320 routers only, configure VPLS load balancing based on MPLS labels and IP information. For IPv4 traffic, only the IP source and destination addresses are included in the hash key. For MPLS and IPv4 traffic, one or two MPLS labels and IPv4 source and destination addresses are included. For MPLS Ethernet pseudowires, only one or two MPLS labels are included in the hash key.

Options You can configure one or more options to load-balance using the packet information that you specify.

destination-mac—Include the destination-address MAC information in the hash key for Layer 2 load balancing.

label-1 (M120 and M320 routers only)—Include the first MPLS label in the hash key. Used for including a one-label packet for per-flow load balancing of IPv4 VPLS traffic based on IP information and MPLS labels.

label-2 (M120 and M320 routers only)—Include the second MPLS label in the hash key. If both **label-1** and **label-2** are specified, the entire first label and the first 16 bits of the second label are hashed.

payload (MX Series, M120, and M320 routers only)—Include the packet's IP payload in the hash key.

- **ip** (MX Series, M120, and M320 routers only)—Include the IP address of the IPv4 or IPv6 payload in the hash key.
  - **layer-3** (MX Series routers only)—Use this to include Layer 3 information from the packet's IP payload in the hash key.
    - **destination-ip-only** (MX Series routers only)—Use this to include only the destination IP address in the payload in the hash key.
    - **source-ip-only** (MX Series routers only)—Use this to include only the source IP address in the payload in the hash key.

NOTE: You can include either the source-ip-only or the destination-ip-only statement, not both. They are mutually exclusive.

  - **layer-3-only** (M120, and M320 routers only)—Include only the Layer 3 information from the packet's IP payload in the hash key.
  - **layer-4** (MX Series routers only)—Include Layer 4 information from the packet's IP payload in the hash key.

NOTE: On MX Series routers only, you can configure either Layer 3 or Layer 4 load balancing, or both at the same time.

NOTE: On I chip platforms, an unknown Layer 4 header is excluded from load-balance hashing to avoid undesired packet reordering.

source-mac—Include the source-address MAC information in the hash key.

symmetric-hash (MX Series routers only)—Configure the symmetric hash or symmetric hash complement for configuring symmetrical load balancing on an 802.3ad Link Aggregation Group.

- **complement** —Include the complement of the symmetric hash in the hash key.

Required Privilege Level

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

Related Documentation

- *Configuring Load Balancing Based on MAC Addresses*
- *Configuring VPLS Load Balancing Based on IP and MPLS Information*
- *Configuring VPLS Load Balancing on MX Series 3D Universal Edge Routers*
- Configuring VPLS Load Balancing on page 81

fast-reroute-priority

Syntax fast-reroute-priority (high | low | medium);

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

Release Information Statement introduced in Junos OS Release 9.5.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.

Description Specify the fast reroute priority for a VPLS routing instance. You can configure **high**, **medium**, or **low** fast reroute priority to prioritize specific VPLS routing instances for faster convergence and traffic restoration. Because the router repairs next hops for high-priority VPLS routing instances first, the traffic traversing a VPLS routing instance configured with **high** fast reroute priority is restored faster than the traffic for VPLS routing instances configured with **medium** or **low** fast reroute priority.

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

Default low

Options

high—Set the fast reroute priority for a VPLS routing instance to high. During a fast reroute event, the router repairs next hops for high-priority VPLS routing instances first.

low—Set the fast reroute priority for a VPLS routing instance to low, which is the default. During a fast reroute event, the router repairs next hops for low-priority VPLS routing instances last.

medium—Set the fast reroute priority for a VPLS routing instance to medium. During a fast reroute event, the router repairs next hops for medium-priority VPLS instances after high-priority VPLS routing instances but before low-priority VPLS routing instances.

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

Related Documentation

- *Configuring VPLS Fast Reroute Priority*

identifier (VPLS Multihoming for FEC 129)

Syntax identifier *identifier*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *instance-name* protocols vpls multi-homing site *site-name*],
[edit routing-instances *instance-name* protocols vpls multi-homing site *site-name*]

Release Information Statement introduced in Junos OS Release 12.3.

Description Configure a Layer 2 VPN or VPLS multihoming identifier (MHID). An identifier needs to be configured for each multihomed site. Multihoming site identifiers are specific to a VPLS domain. They need not be unique on a provider edge (PE) router when multiple VPLS instances are present. The network layer reachability information (NLRI) advertisements sent to a CE device are identified as candidates for designated forwarder selection because the advertisements have the same multihoming identifier. Thus, you should assign the same identifier on all VPLS PE routers that are multihomed to the same customer site.

NOTE: The route distinguisher must be unique among PE routers participating in a multihomed site, so that the RD:MHID combination is unique across multiple VPLS domains. For example, one PE router might have a route distinguisher of 192.0.2.4:1, and another PE router in the same site might have a route distinguisher of 192.0.2.:1. The first number can be, for example, the loopback interface address that identifies the PE router. The second number is the multihoming identifier.

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

Options ***identifier***—Number that identifies the multihomed site.

Range: 1 through 65535

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

Related Documentation

- *Example: Configuring VPLS Multihoming (FEC 129)*

interface (Routing Instances)

Syntax

```
interface interface-name {
    description text;
}
```

Hierarchy Level

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

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.2 for EX Series switches.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.
Statement introduced in Junos OS Release 13.2 for MX 3D Series routers.

Description

Specify the interface over which the VPN traffic travels between the PE device and CE device. You configure the interface on the PE device. If the value **vrf** is specified for the **instance-type** statement included in the routing instance configuration, this statement is required.

Options

interface-name—Name of the interface.

The remaining statement is explained separately.

Required Privilege Level

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

Related Documentation

- *Configuring Routing Instances on PE Routers in VPNs*
- *Configuring EVPN Routing Instances*
- *Configuring EVPN Routing Instances on EX9200 Switches*
- interface (VPLS Routing Instances) on page 130

interface (VPLS Multihoming for FEC 129)

Syntax

```
interface interface-name {
    preference preference-value;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols vpls multi-homing site *site-name*],
[edit routing-instances *instance-name* protocols vpls multi-homing site *site-name*]

Release Information

Statement introduced in Junos OS Release 12.3.

Description

Configure the interface that connects this site to the VPN.

The remaining statement is explained separately.

Options

interface-name—Name of the interface (for example, **ge-0/1/0.1**).

Required Privilege Level

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

Related Documentation

- *Example: Configuring VPLS Multihoming (FEC 129)*

interface (VPLS Routing Instances)

Syntax

```
interface interface-name {
    mac-pinning;
    interface-mac-limit limit;
}
```

Hierarchy Level

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

Release Information

Statement introduced before Junos OS Release 7.4.

Description

Configure the interface for a pseudowire to the VPLS customer site. To complete the configuration of interfaces for a VPLS routing instance, you must also configure the interfaces specified for a VPLS site at the **[edit routing-instances *routing-instance-name*]** hierarchy level as described in *Configuring Routing Instances on PE Routers in VPNs*.

Options

interface-name—Specify the name of the interface used by the VPLS site.

The remaining statement is explained separately.

Required Privilege Level

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

Related Documentation

- Configuring the VPLS Site Interfaces on page 27
- *Configuring Routing Instances on PE Routers in VPNs*
- interface (Routing Instances) on page 128

interface-mac-limit (VPLS)

Syntax

```
interface-mac-limit limit {
    packet-action drop;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls site *site-name* interfaces *interface-name*],
[edit routing-instances *routing-instance-name* protocols evpn],
[edit routing-instances *routing-instance-name* protocols evpn interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols vpls site *site-name* interfaces *interface-name*]

Release Information

Statement introduced before Junos OS Release 7.4.
Support for EVPNs introduced in Junos OS Release 13.2 on MX 3D Series routers.
Support for EVPNs introduced in Junos OS Release 14.2 on EX Series switches.

Description

Specify the maximum number of media access control (MAC) addresses that can be learned by the EVPN or VPLS routing instance. You can configure the same limit for all interfaces configured for a routing instance. You can also configure a limit for a specific interface.

Starting with Junos OS Release 12.3R4, if you do not configure the parameter to limit the number of MAC addresses to be learned by a VPLS instance, the default value is not effective. Instead, if you do not include the **interface-mac-limit** option at the **[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls site *site-name* interfaces *interface-name*],** hierarchy level, this setting is not present in the configuration with the default value of 1024 addresses. If you upgrade a router running a Junos OS release earlier than Release 12.3R4 to Release 12.3R4 or later, you must configure the **interface-mac-limit** option with a valid value for it to be saved in the configuration.

Options

limit—Number of MAC addresses that can be learned from each interface.

Range: 16 through 65,536 MAC addresses

Default: 1024 addresses

The remaining statement is explained separately.

Required Privilege Level

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

Related Documentation

- *Configuring EVPN Routing Instances*
- *Configuring EVPN Routing Instances on EX9200 Switches*
- Limiting the Number of MAC Addresses Learned from an Interface on page 34
- *interface*
- mac-table-size on page 139

l2vpn-id

Syntax l2vpn-id (*as-number:id* | *ip-address:id*);

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

Release Information Statement introduced in Junos OS Release 10.4R2.

Description Specify a globally unique Layer 2 VPN community identifier for the instance.

Options ***as-number:id***—Autonomous system number (**l2vpn-id:*as-number:2-byte-number***. For example: **l2vpn-id l2vpn-id:100:200**. The AS number can be in the range from 1 through 65,535.

ip-address:id—IP address (**l2vpn-id:*ip-address:2-byte-number***. For example: **l2vpn-id l2vpn-id:10.1.1.1:2**. The IP address can be any globally unique unicast address.

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

Related Documentation

- *Example: Configuring BGP Autodiscovery for LDP VPLS*
- *Example: Configuring BGP Autodiscovery for LDP VPLS with User-Defined Mesh Groups*
- *Example: Configuring FEC 129 BGP Autodiscovery for VPWS*

label-block-size

Syntax label-block-size *size*;

Hierarchy Level [edit logical-systems *profile-name* routing-instances *instance-name* protocols vpls],
[edit routing-instances *instance-name* protocols vpls]

Release Information Statement introduced in Junos OS Release 10.0.

Description Configure the label block size for VPLS labels.

Default 8

Options

- 2—Allocate the label blocks in increments of 2. For a VPLS domain that has only two sites with no future expansion plans.
- 4—Allocate the label blocks in increments of 4.
- 8 (default)—Allocate the label blocks in increments of 8.
- 16—Allocate the label blocks in increments of 16. A label block size of 16 enables you to minimize the number of routes in the VPLS domain. Use this setting only if the number of routes is the most important concern.

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

Related Documentation

- Configuring the Label Block Size for VPLS on page 44

label-switched-path-template (Multicast)

Syntax

```
label-switched-path-template {
    (default-template | lsp-template-name);
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel rsvp-te],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel ingress-replication label-switched-path],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective group *address* source *source-address* rsvp-te],
[edit logical-systems *logical-system-name* routing-options dynamic-tunnels *tunnel-name* rsvp-te *entry-name*],
[edit protocols mvpn inter-region-segmented template *template-name* region *region-name* ingress-replication label-switched-path],
[edit protocols mvpn inter-region-segmented template *template-name* region *region-name* rsvpe-te],
[edit protocols mvpn inter-region-template template *template-name* all-regions ingress-replication label-switched-path],
[edit protocols mvpn inter-region-template template *template-name* all-regions rsvp-te],
[edit routing-instances *routing-instance-name* provider-tunnel ingress-replication label-switched-path],
[edit routing-instances *routing-instance-name* provider-tunnel rsvp-te],
[edit routing-instances *routing-instance-name* provider-tunnel selective group *address* source *source-address* rsvp-te],
[edit routing-options dynamic-tunnels *tunnel-name* rsvp-te *entry-name*]

Release Information

Statement introduced in Junos OS Release 8.5.

Description

Specify the LSP template. An LSP template is used as the basis for other dynamically generated LSPs. This feature can be used for a number of applications, including point-to-multipoint LSPs, flooding VPLS traffic, configuring ingress replication for IP multicast using MBGP MVPNs, and to enable RSVP automatic mesh. There is no default setting for the **label-switched-path-template** statement, so you must configure either the default-template using the **default-template** option, or you must specify the name of your preconfigured LSP template.

Options

default-template—Specify that the default LSP template be used for the dynamically generated LSPs.

lsp-template-name—Specify the name of an LSP to be used as a template for the dynamically generated LSPs.

Required Privilege Level

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

Related Documentation

- *Example: Configuring Ingress Replication for IP Multicast Using MBGP MVPNs*
- *Configuring Point-to-Multipoint LSPs for an MBGP MVPN*
- Configuring Dynamic Point-to-Multipoint Flooding LSPs on page 73

- *Configuring RSVP Automatic Mesh*

local-switching (VPLS)

Syntax local-switching;

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

Release Information Statement introduced in Junos OS Release 9.3.

Description Allows you to terminate multiple Layer 2 circuit pseudowires at a single VPLS mesh group. Note that when using **vpls-id-list**, you can enable local-switching to allow local switching of traffic (including BUM traffic) between multiple pseudo wires. Enabling local-switching also eliminates the need to have a dedicated mesh-group for each LDP spoke pseudo wire (which has a limit of 14).

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

Related Documentation
- Configuring Switching Between Pseudowires Using VPLS Mesh Groups on page 17

mac-flush

Syntax mac-flush [*explicit-mac-flush-message-options*];

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

Release Information Statement introduced in Junos OS Release 10.0.

Description Enable media access control (MAC) flush processing for the virtual private LAN service (VPLS) routing instance or for the mesh group under a VPLS routing instance. MAC flush processing removes MAC addresses from the MAC address database that have been learned dynamically. With the dynamically learned MAC addresses removed, MAC address convergence requires less time to complete.

For certain cases where MAC flush processing is not initiated by default, you can also specify ***explicit-mac-flush-message-options*** that additionally configure the router to send explicit MAC flush messages. To configure the router to send explicit MAC flush messages under specific conditions, include ***explicit-mac-flush-message-options*** with the statement.

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

In certain cases, BGP updates sent by the provider edge (PE) device are delayed for 1 to 5 seconds.

This happens when all of the following condtions are true:

- BGP-based VPLS multihoming sites are configured.
- The **mac-flush** statement is included in the configuration.
- a non-minimum designated-forwarder site (site-x, for example) transitions to non-designated-forwarder status

The BGP update being delayed corresponds to the explicit-MAC flush notification message sent by site-x's PE device (PE2, for example). This BGP update message is not deferred if the designated-forwarder status is lost due to a locally-triggered event (for example, a local attachment-circuit interface going down). In other words, BGP update messages are deferred (in Device PE2) only when the designated-forwarder state is lost due to external events taking place in remote PE devices that also hold site-x (for example, in PE1). Suppose, for example, that Device PE1 is the default designated-forwarder with site-x's local interface in the DOWN state. Device PE2 defers BGP update message after Device PE1's local interface comes back to the UP state.

Options

explicit-mac-flush-message-options—(Optional) You can specify one or more of the following explicit MAC flush message options:

- **any-interface**—(Optional) Send a MAC flush message when any customer-facing attachment circuit interface goes down.
- **any-spoke**—(Optional) Send a MAC FLUSH-FROM-ME flush message to all provider edge (PE) routers in the core when one of the spoke pseudowires between the multitenant unit switch and the other network-facing provider edge (NPE) router goes down, causing the multitenant unit switch to switch to this NPE router.

NOTE: This option has a similar effect in a VPLS multihoming environment with multiple multitenant unit switches connected to NPE routers, where both multitenant unit switches have pseudowires that terminate in a mesh group with local-switching configured. If the **any-spoke** option is enabled, then both PE routers send MAC FLUSH-FROM-ME flush messages to all PEs in the core.

- **propagate**—(Optional) Propagate MAC flush to the core.

Required Privilege Level

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

Related Documentation

- Configuring VPLS Routing Instances on page 21
- Configuring Interoperability Between BGP Signaling and LDP Signaling in VPLS on page 15

mac-table-aging-time

Syntax mac-table-aging-time *time*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls],
[edit routing-instances *routing-instance-name* protocols vpls]
[edit bridge-options],
[edit routing-instances *routing-instance-name* protocols evpn]

NOTE: For MX Series routers, the configuration statement is supported at the **[bridge-options]**, **[protocols vpls]**, and **[protocols evpn]** hierarchy levels only.

Release Information Statement introduced in Junos OS Release 7.4.
Statement introduced in Junos OS Release 15.1 for MX Series routers.

Description Modify the timeout interval for the MAC table.

For MX Series routers, you can use the **global-mac-table-aging-time** statement at the **[edit protocols l2-learning]** hierarchy level to configure the timeout interval at the global level or use the **mac-table-aging-time** to configure the timeout interval for a bridge domain or for a specific VPLS or EVPN instance. If multiple timeout interval values are configured on a router, the router determines the timeout interval value in the following order of priority:

- Timeout interval configured at the VPLS or EVPN instance
- Timeout interval configured for the bridge domain
- Global timeout interval configured on the router

NOTE: For MX Series routers, the timeout interval configuration feature is supported on routers with MPCs only.

Options *time*—Specify the number of seconds to wait between MAC table clearings.

Range: 10 through 1,000,000 seconds

Default: 300 seconds

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

Related Documentation

- Configuring the VPLS MAC Table Timeout Interval on page 32
- *Configuring the MAC Table Timeout Interval*

mac-table-size

Syntax

```
mac-table-size size {
  packet-action drop;
}
```

Hierarchy Level

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

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 13.2 for EVPNs on MX 3D Series routers.
Statement introduced in Junos OS Release 14.2 for EX Series switches.

Description

Specify the size of the MAC address table.

Options

size—Size of the MAC address table.

Range:

- (M Series and T Series routers only) 16 through 65,536 MAC addresses
- (MX Series routers only) 16 through 1,048,575 MAC addresses
- (T4000 routers with Type 5 FPCs only) 16 through 262,143 MAC addresses

NOTE: **Before modifying the size of the MAC address table (to 262,143 addresses), you must enable network services mode by including the enhanced-mode statement at the [edit chassis network-services] hierarchy level and then reboot the router.**

Default: 512 MAC addresses

The remaining statement is explained separately.

Required Privilege Level

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

Related Documentation

- *Configuring EVPN Routing Instances*
- *Configuring EVPN Routing Instances on EX9200 Switches*
- Configuring the Size of the VPLS MAC Address Table on page 33
- *Configuring Improved VPLS MAC Address Learning on T4000 Routers with Type 5 FPCs*
- *enhanced-mode*
- *evpn*

mesh-group (Protocols VPLS)

Syntax

```
mesh-group mesh-group-name {
  interface interface-name;
  l2vpn-id (as-number:id | ip-address:id);
  local-switching;
  mac-flush [ explicit-mac-flush-message-options ];
  neighbor address {...};
  peer-as all;
  pseudowire-status-tlv hot-standby-vc-on;
  route-distinguisher (as-number:id | ip-address:id);
  vpls-id number;
  vrf-export [ policy-names ];
  vrf-import [ policy-names ];
  vrf-target {
    community;
    import community-name;
    export community-name;
  }
}
```

Hierarchy Level

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

Release Information

Statement introduced in Junos OS Release 9.0.
local-switching, **mac-tlv-receive**, **mac-tlv-send**, and **peer-as** options introduced in Junos OS Release 9.3.
pseudowire-status-tlv and **mac-flush** options introduced in Junos OS Release 10.0.
route-distinguisher, **vrf-export**, **vrf-import**, and **vrf-target** options introduced in Junos OS Release 11.2.
hot-standby-vc-on and **interface *interface-name*** options introduced in Junos OS Release 15.1R2.

Description

Specify the virtual private LAN service (VPLS) mesh group. The statement options allow you to specify each provider edge (PE) router that is a member of the mesh group. This statement is also used in the configuration of inter-autonomous system (AS) VPLS with media access control (MAC) operations.

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

Options

mesh-group-name—Name of the VPLS mesh group.

The remaining statements are explained separately.

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

Related Documentation
- Configuring VPLS Routing Instances on page 21
- Configuring Interoperability Between BGP Signaling and LDP Signaling in VPLS on page 15

multi-homing (VPLS Multihoming for FEC 128)

Syntax
multi-homing;

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

Release Information
Statement introduced in Junos OS Release 7.5.

Description
Specify the PE router as being a part of a multihomed site. Include this statement on all PE routers associated with a particular site. Configuration of this statement tracks BGP peers. If no BGP peer is available, all active interfaces for a site are deactivated.

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

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

Related Documentation
- Configuring Multihoming on the PE Router on page 68

multi-homing (VPLS Multihoming for FEC 129)

Syntax

```
multi-homing {
  peer-active;
  site site-name {
    active-interface interface-name {
      any;
      primary interface-name;
    }
    identifier identifier;
    interface interface-name {
      preference preference-value;
    }
    peer-active;
    preference (preference-value | backup | primary);
  }
}
```

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

Release Information Statement introduced in Junos OS Release 12.3.

Description For VPLS autodiscovery (FEC 129), specify the parameters for multihoming to two or more provider edge (PE) routers.

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

The remaining statements are explained separately.

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

Related Documentation

- *Example: Configuring VPLS Multihoming (FEC 129)*

neighbor (Protocols VPLS)

Syntax

```
neighbor neighbor-id {
  mac-pinning;
  associate-profile{
    dynamic-profile-name;
    profile-variable-set profile-variable-set-name;
  }
  backup-neighbor {...}
  community community-name;
  connection-protection;
  encapsulation-type type;
  ignore-encapsulation-mismatch;
  oam {
    bfd-liveness-detection {
      detection-time {
        threshold milliseconds;
      }
      minimum-interval milliseconds;
      minimum-receive-interval milliseconds;
      multiplier number;
      no-adaptation;
      transmit-interval {
        minimum-interval milliseconds;
        threshold milliseconds;
      }
      version (1 | automatic);
    }
    ping-interval;
  }
  pseudowire-status-tlv;
  psn-tunnel-endpoint address;
  revert-time seconds;
  static {
    incoming-label label;
    outgoing-label label;
  }
  switchover-delay milliseconds;
  vpls-id-list vc-id-numbers;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls],
[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols vpls mesh-group *mesh-group-name*],
[edit routing-instances *routing-instance-name* protocols vpls],
[edit routing-instances *instance-name* protocols vpls mesh-group *mesh-group-name*]

Release Information

Statement introduced in Junos OS Release 8.4.
The **pseudowire-status-tlv** option was added in Junos OS Release 10.0.
The **vpls-id-list** option was added in Junos OS Release 14.2 for MX Series routers to provide support for multiple pseudowires between the same pair of PEs in LDP-VPLS.

Description Specify each of the PE routers participating in the VPLS domain. Configuring this statement enables LDP for signaling VPLS.

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

Options ***neighbor-id***—Specify the neighbor identifier for each PE router participating in the VPLS domain.

The remaining statements are explained separately.

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

Related Documentation

- Configuring LDP Signaling for VPLS on page 28

no-tunnel-services

Syntax no-tunnel-services;

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

Release Information
Statement introduced in Junos OS Release 7.6.
Support for static VPLS added in Junos OS Release 10.2.

Description
Configure VPLS on a router without a Tunnel Services PIC. Configuring the **no-tunnel-services** statement creates a label-switched interface (LSI) to provide VPLS functionality. An LSI MPLS label is used as the inner label for VPLS. This label maps to a VPLS routing instance. On the PE router, the LSI label is stripped and then mapped to a logical LSI interface. The Layer 2 Ethernet frame is then forwarded using the LSI interface to the correct VPLS routing instance.

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

Label-switched interfaces configured with the **no-tunnel-services** statement are not supported with GRE tunnels.

NOTE: Although visible in the CLI, the **no-tunnel-services** statement is not supported on DPC cards at the **[edit logical-systems *logical-system-name* protocols vpls static-vpls]** and the **[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls]** hierarchy levels.

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

Related Documentation
- Configuring VPLS Without a Tunnel Services PIC on page 55
- Configuring Static Pseudowires for VPLS on page 59
- Configuring EXP-Based Traffic Classification for VPLS on page 85

peer-active (VPLS Multihoming for FEC 129)

Syntax peer-active;

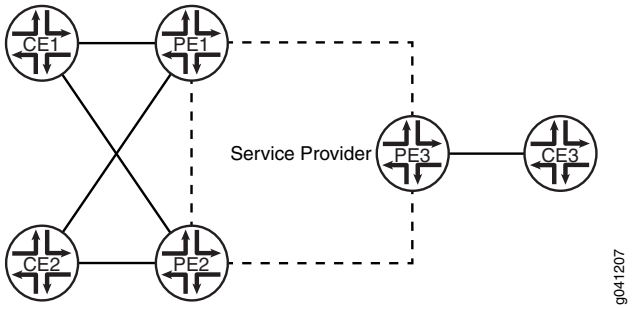
Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *instance-name* protocols vpls multi-homing],
[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols vpls multi-homing site *site-name*],
[edit routing-instances *instance-name* protocols vpls multi-homing],
[edit routing-instances *instance-name* protocols vpls multi-homing site *site-name*]

Release Information Statement introduced in Junos OS Release 12.3.

Description Keep customer edge (CE) interfaces in the up state when all BGP peers go down.

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

Consider a scenario in which two provider edge (PE) routers are sharing two multihomed sites under one routing instance, with two CE devices, CE1 and CE2.



If the BGP peering session drops between Router PE1 and Router PE2, each one would consider itself to be the designated forwarder (DF) for Device CE1 and Device CE2. This creates a loop through the two CE devices, in which traffic loops from one CE device to the other then back to the first.

Junos OS overcomes this scenario by dropping all multihomed CE interface traffic on all multihoming PE routers when the BGP session drops between the PE routers. This functionality is enabled by default for all sites in a routing instance.

The **peer-active** statement disables the default functionality, so that PE routers keep their multihomed CE interfaces in the up state, even though the BGP peering session is down.

If you configure this statement in the **multi-homing** hierarchy, the default functionality is disabled for all sites. If you configure this statement for a site, the default functionality is disabled only for that particular site.

Default If you omit this statement, Junos OS drops all multihomed CE interface traffic on all multihoming PE routers when the BGP session drops between the PE routers.

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

Related Documentation
- *Example: Configuring VPLS Multihoming (FEC 129)*

peer-as (VPLS)

Syntax
```
peer-as {
  all;
}
```

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

Release Information Statement introduced in Junos OS Release 9.3.

Description Enable the autonomous system border router (ASBR) to establish a single pseudowire to each of the other ASBRs interconnected using inter-AS VPLS with MAC processing at the ASBR.

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

Options **all**—This option is required. All peer routers, the ASBRs, are placed within the same VPLS mesh group.

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

Related Documentation
- Configuring Inter-AS VPLS with MAC Processing at the ASBR on page 18

ping-interval

Syntax ping-interval *seconds*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name* oam],
[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols l2vpn oam],
[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols vpls neighbor *address* oam],
[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols vpls mesh-group *mesh-group-name* neighbor *address* oam],
[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols vpls oam],
[edit protocols l2circuit neighbor *address* interface *interface-name* oam],
[edit routing-instances *instance-name* protocols l2vpn oam],
[edit routing-instances *instance-name* protocols vpls neighbor *address* oam],
[edit routing-instances *instance-name* protocols vpls mesh-group *mesh-group-name* neighbor *address* oam],
[edit routing-instances *instance-name* protocols vpls oam]

Release Information
Statement introduced in Junos OS Release 10.0.
Support for FEC 129 VPLS added in Junos OS Release 12.2.

Description Configure the time interval between ping messages for bidirectional forwarding detection (BFD) sessions enabled over pseudowires inside a VPN.

Options *seconds*—Time interval between ping messages.

Range: 30 through 3600

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

Related Documentation
- *Configuring BFD for VCCV for Layer 2 VPNs, Layer 2 Circuits, and VPLS* in the *Junos OS VPNs Library for Routing Devices*

preference (Interface-Level Preference for VPLS Multihoming for FEC 129)

Syntax preference *preference-value*;

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

Release Information Statement introduced in Junos OS Release 12.3.

Description Specify a preference for the interface to become the designated forwarder (DF) for a multihomed VPLS site. This **preference** statement can be useful when you want the interface preference for a site to change dynamically so that the DF election can be influenced depending on the interface state. Among the list of interface preferences, Junos OS advertises the best preference as the VPLS site's preference value. For example, if the site has three interfaces configured with preference values 12, 10, and 9, respectively, 12 is advertised as the site preference. If that interface goes down, 10 is advertised as the site preference.

If you configure interface-level preference, you cannot configure site-level preference.

Options ***preference-value***—Preference value for the interface.

Range: 1 through 65535

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

Related Documentation

- *Example: Configuring VPLS Multihoming (FEC 129)*

preference (Site-Level Preference for VPLS Multihoming for FEC 129)

Syntax preference (*preference-value* | backup | primary);

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *instance-name* protocols vpls multi-homing site *site-name*],
[edit routing-instances *instance-name* protocols vpls multi-homing site *site-name*]

Release Information Statement introduced in Junos OS Release 12.3.

Description Influence the designated forwarder (DF) selection for a multihomed VPLS site. Configure the preference in terms of keywords **primary** and **backup**, or configure the preference value explicitly.

Default If this statement is omitted, the default preference value for the site is 100.

Options *preference-value*—Preference value for the DF.

Range: 1 through 65535

backup—Less likely to become the DF.

primary—Most likely to become the DF.

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

Related Documentation

- *Example: Configuring VPLS Multihoming (FEC 129)*

primary (VPLS Multihoming)

Syntax primary *interface-name*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *instance-name* protocols vpls multi-homing active-interface],
[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols vpls multi-homing site *site-name* active-interface],
[edit routing-instances *instance-name* protocols vpls multi-homing active-interface],
[edit routing-instances *instance-name* protocols vpls multi-homing site *site-name* active-interface]

Release Information Statement introduced in Junos OS Release 7.5.

Description Specify the name of the multihomed interface to be used as the primary interface by the VPLS site.

For FEC 128, use the **[edit routing-instances *instance-name* protocols vpls multi-homing active-interface]** hierarchy level.

Default If you omit this statement, depending on the order in which interfaces are listed in the PE router's configuration, the first operational interface in the set of configured interfaces is chosen to be the primary interface.

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

Options ***interface-name***—Name of the interface (for example, **ge-0/1/0.1**).

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

Related Documentation

- Specifying an Interface as the Active Interface on page 68
- any on page 110

rsvp-te (Routing Instances Provider Tunnel)

Syntax

```
rsvp-te {
  label-switched-path-template {
    (default-template | lsp-template-name);
  }
  static-lsp lsp-name;
}
```

Hierarchy Level

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

Release Information

Statement introduced in Junos OS Release 8.3.

Description

Configure VPLS unknown unicast, broadcast, and multicast traffic flooding using point-to-multipoint LSPs.

Options

static-lsp *lsp-name*—Create a static point-to-multipoint LSP and automatically include all of the neighbors in the VPLS routing instance.

The remaining option is explained separately.

Required Privilege Level

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

Related Documentation

- Flooding Unknown Traffic Using Point-to-Multipoint LSPs in VPLS on page 71

site (VPLS Multihoming for FEC 128)

Syntax

```
site site-name {
    mac-pinning;
    active-interface (any | primary interface-name);
    best-site;
    interface interface-name {
        interface-mac-limit limit;
    }
    mesh-group mesh-group-name;
    multi-homing;
    site-identifier identifier;
    site-preference preference-value;
}
```

Hierarchy Level

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

Release Information

Statement introduced before Junos OS Release 7.4.

Description

Specify the site name and site identifier for a site. Allows you to configure a remote site ID for remote sites.

Options

site-name—Name of the site.

The remaining statements are explained separately.

Required Privilege Level

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

site (VPLS Multihoming for FEC 129)

Syntax

```
site site-name {
  active-interface interface-name {
    any;
    primary interface-name;
  }
  identifier identifier;
  interface interface-name {
    preference preference-value;
  }
  peer-active;
  preference (preference-value | backup | primary);
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols vpls multi-homing],
[edit routing-instances *instance-name* protocols vpls multi-homing]

Release Information

Statement introduced in Junos OS Release 12.3.

Description

For VPLS autodiscovery (FEC 129), specify the parameters for a VPLS site that is multihomed to two or more provider edge (PE) routers.

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

Options

site-name—Name of the site.

The remaining statements are explained separately.

Required Privilege Level

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

Related Documentation

- *Example: Configuring VPLS Multihoming (FEC 129)*

site-identifier (VPLS)

Syntax site-identifier *identifier*;

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

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify the numerical identifier for the local VPLS site.

Options ***identifier***—Specify the numerical identifier for the local VPLS site. The identifier must be an unsigned 16-bit number greater than zero.

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

Related Documentation

- Configuring the VPLS Site Name and Site Identifier on page 23

site-preference

Syntax

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

Hierarchy Level

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

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced for Layer 2 VPNs in Junos OS Release 9.1.

Description

Specify the preference value advertised for a particular Layer 2 VPN or VPLS site. The site preference value is encoded in the BGP local preference attribute. When a PE router receives multiple advertisements with the same VE identifier, the advertisement with the highest local preference value is preferred. You can use this statement to enable multihoming for Layer 2 VPNs and VPLS.

Options

preference-value—Specify the preference value advertised for a Layer 2 VPN or VPLS site.

Range: 1 through 65,535

backup—Set the preference value to 1.

primary—Set the preference value to 65,535.

Required Privilege Level

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

Related Documentation

- *Configuring the Local Site on PE Routers in Layer 2 VPNs*
- Configuring the VPLS Site Preference on page 27

site-range

Syntax site-range *number*;

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

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify an upper limit on the maximum site identifier that can be accepted to allow a pseudowire to be brought up. Pseudowires cannot be established to sites with site identifiers greater than the configured site range. If you issue the **show vpls connections** command, such sites are displayed as OR (out of range).

Options *number*—Maximum number of site identifiers. We recommend using the default value.

Range: 1 through 65,534

Default: 65,534

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

Related Documentation

- Configuring the Site Range on page 25

static (Protocols VPLS)

Syntax

```
static {
    incoming-label label;
    outgoing-label label;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls neighbor *address* backup-neighbor *address*],
[edit routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols vpls neighbor *address*],
[edit routing-instances *routing-instance-name* protocols vpls neighbor *address* backup-neighbor *address*]

Release Information

Statement introduced in Junos OS Release 10.2.

Description

Specifies a static pseudowire for a VPLS domain. By configuring static pseudowires for the VPLS domain, you do not need to configure the LDP or BGP protocols that would normally be used for signaling. Static pseudowires require that you configure a set of in and out labels for each pseudowire configured for the VPLS domain. You can configure both static and dynamic neighbors within the same VPLS routing instance. You can also configure a static pseudowire for a backup neighbor (if you configure the neighbor as static the backup must also be static) and for a mesh group.

Options

incoming-label *label*—You must configure an incoming label for the static pseudowire.

Range: 29,696 through 41,983 and 1,000,000 through 1,048,575

outgoing-label *label*—You must configure an outgoing label for the static pseudowire.

Range: 16 through 1,048,575

Required Privilege Level

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

Related Documentation

- See Configuring Static Pseudowires for VPLS on page 59.

template

Syntax template;

Hierarchy Level [edit logical-systems *logical-system-name* protocols mpls label-switched-path *p2mp-lsp-template-name*],
[edit protocols mpls label-switched-path *p2mp-lsp-template-name*]

Release Information Statement introduced in Junos OS Release 8.3.
Statement introduced in Junos OS Release 12.3X50 for the QFX Series.

Description Specify a template for the dynamically generated point-to-multipoint LSPs used for VPLS flooding.

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

Related Documentation
- Configuring Dynamic Point-to-Multipoint Flooding LSPs with a Preconfigured Template on page 74

traceoptions (Protocols VPLS)

Syntax

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

Hierarchy Level

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

Release Information

Statement introduced before Junos OS Release 7.4.

Description

Trace traffic flowing through a VPLS routing instance.

Options

disable—(Optional) Disable the tracing operation. You can use this option to disable a single operation when you have defined a broad group of tracing operations, such as **all**.

file *filename*—Name of the file to receive the output of the tracing operation. Enclose the name in quotation marks (" ").

files *number*—(Optional) Maximum number of trace files. When a trace file named ***trace-file*** reaches this size, it is renamed ***trace-file*.0**. When ***trace-file*** again reaches its maximum size, ***trace-file*.0** is renamed ***trace-file*.1** and ***trace-file*** is renamed ***trace-file*.0**. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum number of files, you also must specify a maximum file size with the **size** option.

Range: 2 through 1000 files

Default: 2 files

flag *flag*—Tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements. You can specify the following tracing flags:

- **all**—All VPLS tracing options
- **connections**—VPLS connections (events and state changes)
- **error**—Error conditions
- **nlri**—VPLS advertisements received or sent by means of the BGP
- **route**—Routing information
- **topology**—VPLS topology changes caused by reconfiguration or advertisements received from other provider edge (PE) routers using BGP

flag-modifier—(Optional) Modifier for the tracing flag. You can specify the following modifiers:

- **detail**—Provide detailed trace information.

- **disable**—Disable the tracing flag.
- **receive**—Trace received packets.
- **send**—Trace sent packets.

no-world-readable—Do not allow any user to read the log file.

size *size*—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named ***trace-file*** reaches this size, it is renamed ***trace-file*.0**. When ***trace-file*** again reaches its maximum size, ***trace-file*.0** is renamed ***trace-file*.1** and ***trace-file*** is renamed ***trace-file*.0**. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum file size, you also must specify a maximum number of trace files with the **files** option.

Syntax: ***x*k** to specify kilobytes, ***x*m** to specify megabytes, or ***x*g** to specify gigabytes

Range: 10 KB through the maximum file size supported on your system

Default: 1 MB

world-readable—Allow any user to read the log file.

Required Privilege Level

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

Related Documentation

- Tracing VPLS Traffic and Operations on page 103

tunnel-services (Routing Instances VPLS)

Syntax

```
tunnel-services {
    devices device-names;
    primary primary-device-name;
}
```

Hierarchy Level

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

Release Information

Statement introduced before Junos OS Release 7.4.

Description

Specify that traffic for particular VPLS routing instances be forwarded to specific virtual tunnel (VT) interfaces, allowing you to load-balance VPLS traffic among all the available VT interfaces on the router.

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

Options

devices *device-names*—Specify the VT interfaces acceptable for use by the VPLS routing instance. If you do not configure this option, all VT interfaces available to the router can be used for de-encapsulating traffic for this instance.

primary *primary-device-name*—Specify the primary VT interface to be used by the VPLS routing instance. The VT interface specified is used to de-encapsulate all VPLS traffic from the MPLS core network for this routing instance. If the VT interface specified is unavailable, then one of the other acceptable VT interfaces is used for handling the VPLS traffic. If you do not configure this option, any acceptable VT interface can be used to de-encapsulate VPLS traffic from the core.

Required Privilege Level

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

Related Documentation

- Specifying the VT Interfaces Used by VPLS Routing Instances on page 38

vlan-id

Syntax vlan-id *number*;

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

Release Information Statement introduced before Junos OS Release 7.4.

Description For Fast Ethernet and Gigabit Ethernet interfaces only, bind an 802.1Q VLAN tag ID to a logical interface.

Options *number*—A valid VLAN identifier.

Range: For 4-port Fast Ethernet PICs configured to handle VPLS traffic, 512 through 1023. For 1-port and 10-port Gigabit Ethernet PICs configured to handle VPLS traffic, 512 through 4094.

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

Related Documentation
- Configuring Interfaces for VPLS Routing on page 47

vlan-id-list (Interface in VPLS)

Syntax vlan-id-list [*numbers number–number*];

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

Release Information Statement introduced for VPLS in Junos OS Release 10.2.

Description Configure a logical interface to forward packets and learn MAC addresses within each VPLS routing instance configured with a VLAN ID that matches a VLAN ID specified in the list. VLAN IDs can be entered individually using a space to separate each ID, entered as an inclusive list separating the starting VLAN ID and ending VLAN ID with a hyphen, or a combination of both.

Options *number number*—Individual VLAN IDs separated by a space.

number-number—Starting VLAN ID and ending VLAN ID in an inclusive range.

Range: 1 through 4095

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

Related Documentation
- Configuring Interfaces for VPLS Routing on page 47
- Configuring VLAN IDs for Logical Interfaces on page 51

vlan-tagging

Syntax vlan-tagging;

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

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.2 for ACX Series Universal Access Routers.
Statement introduced in Junos OS Release 13.2 for PTX Series Routers.
Statement introduced in Junos OS Release 14.1X53-D10 for the QFX Series.

Description For Fast Ethernet and Gigabit Ethernet interfaces, aggregated Ethernet interfaces configured for VPLS, and pseudowire subscriber interfaces, enable the reception and transmission of 802.1Q VLAN-tagged frames on the interface.

NOTE: On EX Series switches except for EX4300 and EX9200 switches, the **vlan-tagging** and **family ethernet-switching** statements cannot be configured on the same interface. Interfaces on EX2200, EX3200, EX3300, EX4200, and EX4500 switches are set to **family ethernet-switching** by the default factory configuration. EX6200 and EX8200 switch interfaces do not have a default **family** setting.

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

Related Documentation

- *802.1Q VLANs Overview*
- *vlan-id*
- *Configuring a Layer 3 Subinterface (CLI Procedure)*
- *Configuring Tagged Aggregated Ethernet Interfaces*
- *Example: Configuring Layer 3 Subinterfaces for a Distribution Switch and an Access Switch*

vpls (Interfaces)

Syntax vpls;

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

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify the VPLS protocol family information for the logical interface.

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

Related Documentation

- Configuring Interfaces for VPLS Routing on page 47

vpls (Routing Instance)

Syntax

```
vpls {
    mac-pinning;
    active-interface {
        any;
        primary interface-name;
    }
    community COMM;
    connectivity-type (ce | irb);
    control-word;
    encapsulation-type ethernet;
    ignore-encapsulation-mismatch;
    ignore-mtu-mismatch;
    import-labeled-routes [ routing-instance-name ];
    interface interface-name;
    interface-mac-limit limit;
    label-block-size size;
    mac-flush [ explicit-mac-flush-message-options ];
    mac-table-aging-time time;
    mac-table-size size;
    mesh-group mesh-group-name {
        interface interface-name;
        l2vpn-id (as-number:id | ip-address:id);
        local-switching;
        mac-flush [ explicit-mac-flush-message-options ];
        neighbor address {...};
        peer-as all;
        pseudowire-status-tlv hot-standby-vc-on;
        route-distinguisher (as-number:id | ip-address:id);
        vpls-id number;
        vrf-export [ policy-names ];
        vrf-import [ policy-names ];
        vrf-target {
            community;
            import community-name;
            export community-name;
        }
    }
    mtu mtu;
    no-control-word;
    no-tunnel-services;
    site site-name {
        active-interface interface-name {
            any;
            primary preference-value;
        }
        best-site;
        interface interface-name {
            interface-mac-limit limit;
        }
        mesh-group mesh-group-name;
        multi-homing;
        site-identifier identifier;
```

```
    site-preference preference-value {
      backup;
      primary;
    }
  }
  site-range number;
  traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
  }
  tunnel-services {
    devices device-names;
    primary primary-device-name;
  }
}
```

Hierarchy Level

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

Release Information

Statement introduced before Junos OS Release 7.4.
mac-flush option introduced in Junos OS Release 10.0.
hot-standby-vc-on, **import-labeled-routes [** *routing-instance-name* **]**, and **interface** *interface* options introduced in Junos OS Release 15.1R2.

Description

Configure a virtual private LAN service (VPLS) routing instance.

The remaining statements are explained separately.

Required Privilege Level

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

Related Documentation

- Configuring VPLS Routing Instances on page 21

vpls-id

Syntax vpls-id *vpls-id*;

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

Release Information Statement introduced in Junos OS Release 8.4.

Description Identify the virtual circuit identifier used for the VPLS routing instance or mesh group. This statement is a part of the configuration to enable LDP signaling for VPLS.

Options ***vpls-id***—Specify a valid identifier for the VPLS routing instance.

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

Related Documentation
- Configuring LDP Signaling for VPLS on page 28

CHAPTER 14

Operational Commands

- Operational-Mode Commands on page 169

Operational-Mode Commands

- Overview of Junos OS CLI Operational Mode Commands on page 169
- Example: Running Operational Mode Commands on Logical Systems on page 172
- Example: Viewing BGP Trace Files on Logical Systems on page 173
- Example: Configuring System Logging on Logical Systems on page 178

Overview of Junos OS CLI Operational Mode Commands

This topic provides an overview of Junos OS CLI operational mode commands and contains the following sections:

- CLI Command Categories on page 169
- Commonly Used Operational Mode Commands on page 170

CLI Command Categories

When you log in to a device running Junos OS and the CLI starts, there are several broad groups of CLI commands:

- Commands for controlling the CLI environment—Some set commands in the **set** hierarchy configure the CLI display screen. For information about these commands, see *Understanding the Junos OS CLI Modes, Commands, and Statement Hierarchies*.
- Commands for monitoring and troubleshooting—The following commands display information and statistics about the software and test network connectivity. Detailed command descriptions are provided in the *Junos OS Interfaces Command Reference*.
  - **clear**—Clear statistics and protocol database information.
  - **mtrace**—Trace mtrace packets from source to receiver.
  - **monitor**—Perform real-time debugging of various software components, including the routing protocols and interfaces.
  - **ping**—Determine the reachability of a remote network host.

  - **show**—Display the current configuration and information about interfaces, routing protocols, routing tables, routing policy filters, system alarms, and the chassis.
  - **test**—Test the configuration and application of policy filters and autonomous system (AS) path regular expressions.
  - **traceroute**—Trace the route to a remote network host.

- Commands for connecting to other network systems—The **ssh** command opens Secure Shell connections, and the **telnet** command opens telnet sessions to other hosts on the network. For information about these commands, see the CLI Explorer.
- Commands for copying files—The **copy** command copies files from one location on the router or switch to another, from the router or switch to a remote system, or from a remote system to the router or switch. For information about these commands, see the CLI Explorer.
- Commands for restarting software processes—The commands in the **restart** hierarchy restart the various Junos OS processes, including the routing protocol, interface, and SNMP. For information about these commands, see the CLI Explorer.
- A command—**request**—for performing system-level operations, including stopping and rebooting the router or switch and loading Junos OS images. For information about this command, see the CLI Explorer.
- A command—**start**—to exit the CLI and start a UNIX shell. For information about this command, see the CLI Explorer.
- A command—**configure**—for entering configuration mode, which provides a series of commands that configure Junos OS, including the routing protocols, interfaces, network management, and user access. For information about the CLI configuration commands, see *Understanding Junos OS CLI Configuration Mode*.
- A command—**quit**—to exit the CLI. For information about this command, see the CLI Explorer.
- For more information about the CLI operational mode commands, see the CLI Explorer.
  .

Commonly Used Operational Mode Commands

Table 6 on page 170 lists some operational commands you may find useful for monitoring router or switch operation. For a complete description of operational commands, see the Junos OS command references.

NOTE: The QFX3500 switch does not support the IS-IS, OSPF, BGP, MPLS, and RSVP protocols.

Table 6: Commonly Used Operational Mode Commands

Items to Check	Description	Command
Software version	Versions of software running on the router or switch	**show version**

Table 6: Commonly Used Operational Mode Commands *(continued)*

Items to Check	Description	Command
Log files	Contents of the log files	**monitor**
	Log files and their contents and recent user logins	**show log**
Remote systems	Host reachability and network connectivity	**ping**
	Route to a network system	**traceroute**
Configuration	Current system configuration	**show configuration**
Manipulate files	List of files and directories on the router or switch	**file list**
	Contents of a file	**file show**
Interface information	Detailed information about interfaces	**show interfaces**
Chassis	Chassis alarm status	**show chassis alarms**
	Information currently on craft display	**show chassis craft-interface**
	Router or switch environment information	**show chassis environment**
	Hardware inventory	**show chassis hardware**
Routing table information	Information about entries in the routing tables	**show route**
Forwarding table information	Information about data in the kernel's forwarding table	**show route forwarding-table**
IS-IS	Adjacent routers or switches	**show isis adjacency**
OSPF	Display standard information about OSPF neighbors	**show ospf neighbor**
BGP	Display information about BGP neighbors	**show bgp neighbor**
MPLS	Status of interfaces on which MPLS is running	**show mpls interface**
	Configured LSPs on the router or switch, as well as all ingress, transit, and egress LSPs	**show mpls lsp**
	Routes that form a label-switched path	**show route label-switched-path**
RSVP	Status of interfaces on which RSVP is running	**show rsvp interface**
	Currently active RSVP sessions	**show rsvp session**
	RSVP packet and error counters	**show rsvp statistics**

Example: Running Operational Mode Commands on Logical Systems

This example shows how to set the CLI to a specified logical system view, run operational-mode commands for the logical system, and then return to the main router view.

- Requirements on page 172
- Overview on page 172
- Configuration on page 172

Requirements

You must have the **view** privilege for the logical system.

Overview

For some operational-mode commands, you can include a **logical-system** option to narrow the output of the command or to limit the operation of the command to the specified logical system. For example, the **show route** command has a **logical-system** option. To run this command on a logical system called LS3, you can use **show route logical-system LS3**. However, some commands, such as **show interfaces**, do not have a **logical-system** option. For commands like this, you need another approach.

You can place yourself into the context of a specific logical system. To configure a logical system context, issue the **set cli logical-system *logical-system-name*** command.

When the CLI is in logical system context mode and you enter an operational-mode command, the output of the command displays information related to the logical system only.

Configuration

Step-by-Step Procedure

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

To set the CLI to a specific logical system context:

1. From the main router, configure the logical system.

   ```
   [edit]
   user@host# set logical-systems LS3
   ```

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

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

3. Set the CLI to view the logical system.

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

   user@host:LS3>
   ```

4. Run an operational-mode command.

```
user@host:LS3> show interfaces terse
Interface               Admin Link Proto    Local                 Remote
lt-1/2/0
lt-1/2/0.3              up    up   inet     10.0.2.1/30
```

5. Enter configuration mode to edit the logical system configuration.

```
user@host:LS3> edit
Entering configuration mode

user@host:LS3#
```

6. Exit configuration mode to return to operational mode.

```
user@host:LS3# exit
Exiting configuration mode
```

7. Clear the logical system view to return to the main router view.

```
user@host:LS3> clear cli logical-system
Cleared default logical system

user@host>
```

8. To achieve the same effect when using a Junos XML protocol client application, include the <**set-logical-system**> tag.

```
<rpc>
<set-logical-system>
<logical-system>LS1</logical-system>
</set-logical-system>
</rpc>
```

Example: Viewing BGP Trace Files on Logical Systems

This example shows how to list and view files that are stored on a logical system.

- Requirements on page 173
- Overview on page 174
- Configuration on page 174
- Verification on page 178

Requirements

- You must have the **view** privilege for the logical system.
- Configure a network, such as the BGP network shown in *Example: Configuring Internal BGP Peering Sessions on Logical Systems*.

Overview

Logical systems have their individual directory structure created in the **/var/logical-systems/*logical-system-name*** directory. It contains the following subdirectories:

- **/config**—Contains the active configuration specific to the logical system.
- **/log**—Contains system log and tracing files specific to the logical system.

  To maintain backward compatibility for the log files with previous versions of Junos OS, a symbolic link (symlink) from the **/var/logs/*logical-system-name*** directory to the **/var/logical-systems/*logical-system-name*** directory is created when a logical system is configured.
- **/tmp**—Contains temporary files specific to the logical system.

The file system for each logical system enables logical system users to view trace logs and modify logical system files. Logical system administrators have full access to view and modify all files specific to the logical system.

Logical system users and administrators can save and load configuration files at the logical-system level using the **save** and **load** configuration mode commands. In addition, they can also issue the **show log**, **monitor**, and **file** operational mode commands at the logical-system level.

This example shows how to configure and view a BGP trace file on a logical system. The steps can be adapted to apply to trace operations for any Junos OS hierarchy level that supports trace operations.

TIP: To view a list of hierarchy levels that support tracing operations, enter the **help apropos traceoptions** command in configuration mode.

Configuration

- Configuring Trace Operations on page 175
- Viewing the Trace File on page 175
- Deactivating and Reactivating Trace Logging on page 177
- Results on page 178

CLI Quick Configuration

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

```
set logical-systems A protocols bgp group internal-peers traceoptions file bgp-log
set logical-systems A protocols bgp group internal-peers traceoptions file size 10k
set logical-systems A protocols bgp group internal-peers traceoptions file files 2
set logical-systems A protocols bgp group internal-peers traceoptions flag update detail
```

Configuring Trace Operations

Step-by-Step Procedure

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

To configure the trace operations:

1. Configure trace operations on the logical system.

```
[edit logical-systems A protocols bgp group internal-peers]
user@host# set traceoptions file bgp-log
user@host# set traceoptions file size 10k
user@host# set traceoptions file files 2
user@host# set traceoptions flag update detail
```

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

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

Viewing the Trace File

Step-by-Step Procedure

To view the trace file:

1. In operational mode on the main router, list the directories on the logical system.

```
user@host> file list /var/logical-systems/A
/var/logical-systems/A:
config/
log/
tmp/
```

2. In operational mode on the main router, list the log files on the logical system.

```
user@host> file list /var/logical-systems/A/log/
/var/logical-systems/A/log:
bgp-log
```

3. View the contents of the **bgp-log** file.

```
user@host> file show /var/logical-systems/A/log/bgp-log
Aug 10 17:12:01 trace_on: Tracing to "/var/log/A/bgp-log" started
Aug 10 17:14:22.826182 bgp_peer_mgmt_clear:5829: NOTIFICATION sent to
192.163.6.4 (Internal AS 17): code 6 (Cease) subcode 4 (Administratively
Reset), Reason: Management session cleared BGP neighbor
Aug 10 17:14:22.826445 bgp_send: sending 21 bytes to 192.163.6.4 (Internal
AS 17)
Aug 10 17:14:22.826499
Aug 10 17:14:22.826499 BGP SEND 192.168.6.5+64965 -> 192.163.6.4+179
Aug 10 17:14:22.826559 BGP SEND message type 3 (Notification) length 21
Aug 10 17:14:22.826598 BGP SEND Notification code 6 (Cease) subcode 4
(Administratively Reset)
Aug 10 17:14:22.831756 bgp_peer_mgmt_clear:5829: NOTIFICATION sent to
192.168.40.4 (Internal AS 17): code 6 (Cease) subcode 4 (Administratively
Reset), Reason: Management session cleared BGP neighbor
Aug 10 17:14:22.831851 bgp_send: sending 21 bytes to 192.168.40.4 (Internal
 AS 17)
Aug 10 17:14:22.831901
Aug 10 17:14:22.831901 BGP SEND 192.168.6.5+53889 -> 192.168.40.4+179
```

```
Aug 10 17:14:22.831959 BGP SEND message type 3 (Notification) length 21
Aug 10 17:14:22.831999 BGP SEND Notification code 6 (Cease) subcode 4
(Administratively Reset)
...
```

4. Filter the output of the log file.

```
user@host> file show /var/logical-systems/A/log/bgp-log | match "flags 0x40"
Aug 10 17:14:54.867460 BGP SEND flags 0x40 code Origin(1): IGP
Aug 10 17:14:54.867595 BGP SEND flags 0x40 code ASPath(2) length 0: <null>
Aug 10 17:14:54.867650 BGP SEND flags 0x40 code NextHop(3): 192.168.6.5
Aug 10 17:14:54.867692 BGP SEND flags 0x40 code LocalPref(5): 100
Aug 10 17:14:54.884529 BGP RECV flags 0x40 code Origin(1): IGP
Aug 10 17:14:54.884581 BGP RECV flags 0x40 code ASPath(2) length 0: <null>
Aug 10 17:14:54.884628 BGP RECV flags 0x40 code NextHop(3): 192.163.6.4
Aug 10 17:14:54.884667 BGP RECV flags 0x40 code LocalPref(5): 100
Aug 10 17:14:54.911377 BGP RECV flags 0x40 code Origin(1): IGP
Aug 10 17:14:54.911422 BGP RECV flags 0x40 code ASPath(2) length 0: <null>
Aug 10 17:14:54.911466 BGP RECV flags 0x40 code NextHop(3): 192.168.40.4
Aug 10 17:14:54.911507 BGP RECV flags 0x40 code LocalPref(5): 100
Aug 10 17:14:54.916008 BGP SEND flags 0x40 code Origin(1): IGP
Aug 10 17:14:54.916054 BGP SEND flags 0x40 code ASPath(2) length 0: <null>
Aug 10 17:14:54.916100 BGP SEND flags 0x40 code NextHop(3): 192.168.6.5
Aug 10 17:14:54.916143 BGP SEND flags 0x40 code LocalPref(5): 100
Aug 10 17:14:54.920304 BGP RECV flags 0x40 code Origin(1): IGP
Aug 10 17:14:54.920348 BGP RECV flags 0x40 code ASPath(2) length 0: <null>
Aug 10 17:14:54.920393 BGP RECV flags 0x40 code NextHop(3): 10.0.0.10
Aug 10 17:14:54.920434 BGP RECV flags 0x40 code LocalPref(5): 100
```

5. View the tracing operations in real time.

```
user@host> clear bgp neighbor logical-system A
Cleared 2 connections
```

CAUTION: Clearing the BGP neighbor table is disruptive in a production environment.

6. Run the **monitor start** command with an optional **match** condition.

```
user@host> monitor start A/bgp-log | match 0.0.0.0/0
Aug 10 19:21:40.773467 BGP RECV         0.0.0.0/0
Aug 10 19:21:40.773685 bgp_rcv_nlri: 0.0.0.0/0
Aug 10 19:21:40.773778 bgp_rcv_nlri: 0.0.0.0/0 belongs to meshgroup
Aug 10 19:21:40.773832 bgp_rcv_nlri: 0.0.0.0/0 qualified bnp->ribact 0x0
12afcb 0x0
```

7. Pause the **monitor** command by pressing Esc-Q.
   To unpause the output, press Esc-Q again.

8. Halt the **monitor** command by pressing Enter and typing **monitor stop**.

```
[Enter]
user@host> monitor stop
```

9. When you are finished troubleshooting, consider deactivating trace logging to avoid any unnecessary impact to system resources.

```
[edit protocols bgp group internal-peers]
user@host:A# deactivate traceoptions
user@host:A# commit
```

When configuration is deactivated, it appears in the configuration with the **inactive** tag.To reactivate trace operations, use the **activate** configuration-mode statement.

```
[edit protocols bgp group internal-peers]
user@host:A# show

type internal;
inactive: traceoptions {
    file bgp-log size 10k files 2;
    flag update detail;
    flag all;
}
local-address 192.168.6.5;
export send-direct;
neighbor 192.163.6.4;
neighbor 192.168.40.4;
```

10. To reactivate trace operations, use the **activate** configuration-mode statement.

```
[edit protocols bgp group internal-peers]
user@host:A# activate traceoptions
user@host:A# commit
```

Deactivating and Reactivating Trace Logging

Step-by-Step Procedure

To deactivate and reactivate the trace file:

1. When you are finished troubleshooting, consider deactivating trace logging to avoid an unnecessary impact to system resources.

```
[edit protocols bgp group internal-peers]
user@host:A# deactivate traceoptions
user@host:A# commit
```

When configuration is deactivated, the statement appears in the configuration with the **inactive** tag.

```
[edit protocols bgp group internal-peers]
user@host:A# show

type internal;
inactive: traceoptions {
    file bgp-log size 10k files 2;
    flag update detail;
    flag all;
}
local-address 192.168.6.5;
export send-direct;
neighbor 192.163.6.4;
neighbor 192.168.40.4;
```

2. To reactivate logging, use the **activate** configuration-mode statement.

```
[edit protocols bgp group internal-peers]
user@host:A# activate traceoptions
user@host:A# commit
```

Results

From configuration mode, confirm your configuration by entering the **show logical-systems A protocols bgp group internal-peers** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show logical-systems A protocols bgp group internal-peers
traceoptions {
    file bgp-log size 10k files 2;
    flag update detail;
}
```

Verification

Confirm that the configuration is working properly.

Verifying That the Trace Log File Is Operating

Purpose Make sure that events are being written to the log file.

Action

```
user@host:A> show log bgp-log
Aug 12 11:20:57 trace_on: Tracing to "/var/log/A/bgp-log" started
```

Example: Configuring System Logging on Logical Systems

This example shows how to configure system logging on logical systems and how to view the logs.

- Requirements on page 178
- Overview on page 179
- Configuration on page 179
- Verification on page 180

Requirements

This example has the following requirements:

- You must have the **view** privilege for the logical system.
- Junos OS Release 11.4 or later.

Overview

Each logical systems has its individual directory structure created in the **/var/logical-systems/*logical-system-name*** directory. This directory contains the following subdirectories:

- **/config**—Contains the active configuration specific to the logical system.
- **/log**—Contains system log and tracing files specific to the logical system.

  To maintain backward compatibility for the log files with previous versions of Junos OS, a symbolic link (symlink) from the **/var/log/*logical-system-name*** directory to the **/var/logical-systems/*logical-system-name*** directory is created when a logical system is configured.
- **/tmp**—Contains temporary files specific to the logical system.

The file system for each logical system enables logical system users to view trace logs and modify logical system files. Logical system administrators have full access to view and modify all files specific to the logical system.

Logical system users and administrators can save and load configuration files at the logical system level using the **save** and **load** configuration mode commands. In addition, they can issue the **show log**, **monitor**, and **file** operational mode commands at the logical system level.

This example shows how to configure system logging on a logical system.

Configuration

CLI Quick Configuration

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

```
set logical-systems lsys1 system syslog host 10.209.10.69 ftp critical
set logical-systems lsys1 system syslog allow-duplicates
set logical-systems lsys1 system syslog file lsys1-file1 daemon error
set logical-systems lsys1 system syslog file lsys1-file1 firewall critical
```

Step-by-Step Procedure

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

To configure system logging:

1. Configure trace operations on the logical system.

```
[edit logical-systems lsys1 system syslog]
user@host# set host 10.209.10.69 ftp critical
user@host# set allow-duplicates
user@host# set file lsys1-file1 daemon error
user@host# set file lsys1-file1 firewall critical
```

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

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

Results

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

```
user@host# show logical-systems
lsys1 {
  system {
    syslog {
      host 10.209.10.69 {
        ftp critical;
      }
      allow-duplicates;
      file lsys1-file1 {
        daemon error;
        firewall critical;
      }
    }
  }
}
```

Verification

Confirm that the configuration is working properly.

Verifying That the System Log File Is Operating

Purpose Make sure that events are being written to the log file.

Action

TIP: To make entries in the system log, you can use the **start shell** command and then use the **logger** shell command. For example: **logger -e "firewall_crit" -p firewall.crit -l lsys1 TEST**

```
user@host> show log lsys1/lsys1-file1
Sep 7 14:15:46 host clear-log[2752]: logfile cleared
Sep  7 14:19:04  host logger: %-: firewall_crit: TEST
...

user@host> file show /var/logical-systems/lsys1/log/lsys1-file1
Sep  7 14:19:04  host logger: %-: firewall_crit: TEST
...
```

Related Documentation

- *Introduction to Logical Systems*