Junos® OS

VPWS Feature Guide for Routing Devices

Release

14.1

Published: 2014-07-21

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

Junos® OS VPWS Feature Guide for Routing Devices
14.1

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

Table of Contents

List of Figures

List of Tables

About the Documentation

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:

- MX Series
- T Series
- M Series
- ACX Series
- PTX 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 the *CLI User Guide*.

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 xvi defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

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

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

Convention	Description	Examples
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**.
> (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 at the Juniper Networks Technical Documentation 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 https://www.juniper.net/cgi-bin/docbugreport/.
- 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 JNASC 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

CHAPTER 1

Introduction to VPWS and Layer 2 Circuits

Understanding VPWS

Virtual private wire service (VPWS) Layer 2 VPNs employ Layer 2 services over MPLS to build a topology of point-to-point connections that connect end customer sites in a VPN. These Layer 2 VPNs provide an alternative to private networks that have been provisioned by means of dedicated leased lines or by means of Layer 2 virtual circuits that employ ATM or Frame Relay. The service provisioned with these Layer 2 VPNs is known as VPWS. You configure a VPWS *instance* on each associated edge device for each VPWS Layer 2 VPN.

Traditional VPNs over Layer 2 circuits require the provisioning and maintenance of separate networks for IP and for VPN services. In contrast, VPWS enables the sharing of a provider's core network infrastructure between IP and Layer 2 VPN services, reducing the cost of providing those services.

Junos OS supports two types of VPWS Layer 2 VPNs:

- Kompella Layer 2 VPNs, which use BGP for autodiscovery and signaling.
- FEC 129 BGP autodiscovery for VPWS, which uses BGP for autodiscovery and LDP as the signaling protocol.

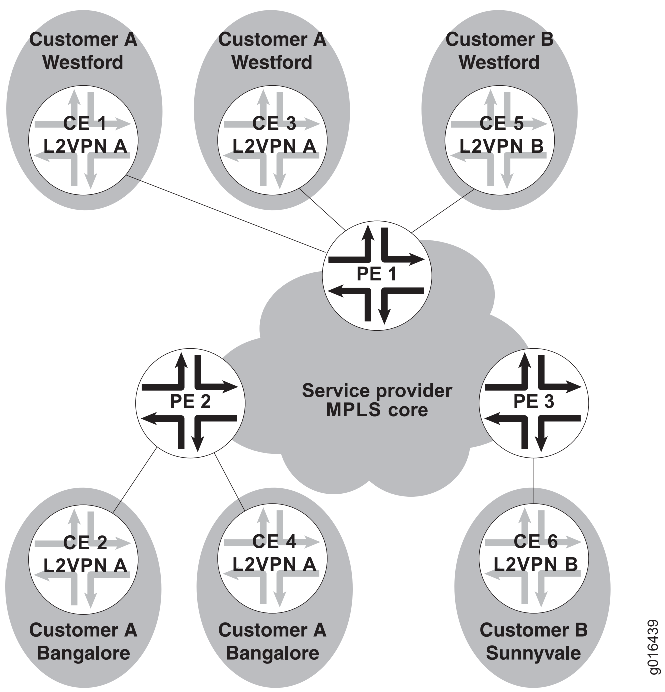
FEC 129 BGP autodiscovery for VPWS requires the **l2vpn-id**, **source-attachment-identifier**, and **target-attachment-identifier** statements. Kompella Layer 2 VPNs require the **site-identifier** and **remote-site-id** statements.

NOTE: VPWS creates pseudowires that emulate Layer 2 circuits. A virtual private LAN service (VPLS) network is similar to VPWS, but provides point-to-multipoint traffic forwarding in contrast to the VPWS Layer 2 VPN's point-to-point traffic forwarding. If you need point-to-multipoint service instead of point-to-point service, consider using VPLS instead of VPWS.

A VPWS Layer 2 VPN can have either a full-mesh or a hub-and-spoke topology. The tunneling mechanism in the core network typically is MPLS. However, VPWS can also use other tunneling protocols, such as GRE. VPWS is similar to Martini Layer 2 services over MPLS, and employs a similar encapsulation scheme for forwarding traffic.

Figure 1 on page 4 illustrates an example of a simple VPWS Layer 2 VPN topology.

Figure 1: VPWS Sample Topology



In this example, the service provider offers VPWS services to Customer A and Customer B. Customer A wants to create a full mesh of point-to-point links between Westford and Bangalore. Customer B needs only a single point-to-point link between Westford and Sunnyvale. The service provider uses BGP and MPLS signaling in the core, and creates a set of unidirectional pseudowires at each provider edge (PE) device to separately cross-connect each customer's Layer 2 circuits.

In order to provision this service, the provider configures two VPWS Layer 2 VPNs, Layer 2 VPN A and Layer 2 VPN B. The circuit cross-connect (CCC) encapsulation type (**ethernet-ccc** or **vlan-ccc**) is configured for each VPWS Layer 2 VPN. All interfaces in a given VPWS Layer 2 VPN must be configured with the VPWS Layer 2 VPN's encapsulation type.

Local and remote site information for the interfaces identifies the cross-connect. Local cross-connects are supported when the interfaces that are connected belong to two different sites configured in the same VPWS instance and on the same PE device.

BGP advertises reachability for the VPNs. The BGP configuration is similar to that used for other VPN services, such as Layer 3 VPNs and VPLS. MPLS is configured to set up base LSPs to the remote PE devices similarly to the other VPN services.

Junos OS provides VPWS support the following configuration methods:

- Pseudowires are manually configured using Forwarding Equivalence Class (FEC) 128.
- Pseudowires are signaled by LDP using FEC 129. This arrangement reduces the configuration burden that is associated with statically configured Layer 2 circuits while still using LDP as the underlying signaling protocol.

Supported and Unsupported Features

Junos OS supports the following features with VPWS :

- Intra-AS VPWS functionality using BGP for autodiscovery and FEC 129 LDP for pseudowire signaling.
- Graceful Routing Engine switchover.
- Operation, administration, and maintenance (OAM) mechanisms, including Bidirectional Forwarding Detection and MPLS ping.
- FEC 128 LDP signaling with static configuration (in Junos OS this is configured within **protocols l2circuit**). With this option, there is no BGP autodiscovery.

Junos OS does not support the following VPWS functionality:

- Multihoming of customer sites to multiple PE devices using the BGP site model of multihoming.
- Terminating FEC 129 VPWS into a mesh group of an FEC 129 VPLS instance.
- Intra-AS VPWS functionality using BGP for autodiscovery and FEC 128 LDP for pseudowire signaling.
- FEC 129 VPWS without BGP autodiscovery.
- Static configuration of VPWS with FEC 129 signaling.
- Nonstop active routing.
- Multi-segment pseudowires.
- Interworking of FEC 128 and FEC 129 VPWS.
- Statically configured Layer 2 circuit-style pseudowire redundancy.
- Inter-AS deployments.

Related Documentation

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

Understanding FEC 129 BGP Autodiscovery for VPWS

The major functional components in a VPWS with FEC 129 are BGP, LDP, and the Layer 2 VPN module of Junos OS. BGP is responsible for distributing the local autodiscovery routes created on each PE device to all other PE devices. LDP is responsible for using the autodiscovery information provided by BGP to set up targeted LDP sessions over which to signal the pseudowires. The Layer 2 VPN is the glue that binds the BGP and LDP functionalities together.

Supported Standards in FEC 129 BGP Autodiscovery for VPWS

The relevant RFCs for this feature are as follows:

- RFC 4447, *Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)*
- RFC 6074, *Provisioning, Auto-Discovery, and Signaling in Layer 2 Virtual Private Networks (L2VPNs)*

Routes and Routing Table Interaction in FEC 129 BGP Autodiscovery for VPWS

BGP, LDP, and Layer 2 VPNs interact through different types of routes installed in the *instance*.l2vpn.0 table. Theroutes that are present in the table are autodiscovery routes and pseudowire routes.

- Autodiscovery routes are used by BGP to allow autodiscovery of remote source access individual identifiers (SAIIs) (the sources of the point-to-point pseudowires) and PE device addresses. Autodiscovery routes are advertised when you configure the **l2vpn auto-discovery-only** address family.

  The format of the autodiscovery routes is a combination of the route distinguisher and the SAII. For example: 10.255.0.1:100:0.0.0.1/96 AD.

  Table 3 on page 6 lists the route elements and the number of associated bytes allocated to each element.

Table 3: Autodiscovery Route Format

Route Element	Bytes
RD	8 bytes
SAII	4 bytes

  The **l2vpn-id** of the FEC 129 VPWS instance is attached to the route in a BGP extended community. One autodiscovery route is advertised for each source attachment identifier (SAI) in the instance.

- Pseudowire routes are installed by the Layer 2 VPN (local) and LDP (remote) to represent the bidirectional components of the pseudowire. For example: NoCtrlWord:5:100:200:2:0.0.0.1/176. The format of the routes is described in Table 4 on page 7.

Table 4: Pseudowire Route Format

Field Name	Field Description
Pseudowire type + control word bit	2 bytes
Remote PE address	4 bytes
Attachment group identifier (AGI) The AGI field of the pseudowire route is always set to the **l2vpn-id** of the instance.	8 bytes
SAII	4 bytes
Target attachment individual identifier (TAII)	4 bytes

Layer 2 VPN Behavior in FEC 129 BGP Autodiscovery for VPWS

A Layer 2 VPN installs a locally generated autodiscovery route into the instance.l2vpn.0 table for every SAII configured in an FEC 129 VPWS instance. The extended community containing the **l2vpn-id** is attached when the route is added to the instance.l2vpn.0 table.

For each autodiscovered SAII from a remote neighbor where the **l2vpn-id** matches the local **l2vpn-id** and the received SAII matches a locally configured TAII, the Layer 2 VPN obtains an MPLS label and generates a pseudowire route and adds it to the instance.l2vpn.0 table. The remote PE address is copied from the BGP protocol next hop for the autodiscovery route.

The Layer 2 VPN module of Junos OS is responsible for installing the forwarding routes into the mpls.0 table as usual.

BGP Autodiscovery Behavior in FEC 129 BGP Autodiscovery for VPWS

Local autodiscovery routes installed by the Layer 2 VPN in the instance.l2vpn.0 table are advertised by BGP to remote PE devices s**l2vpn auto-discovery-only** address family according to the instance and BGP export policies.

On the receiving side, BGP accepts autodiscovery routes from remote peers and installs them in the local bgp.l2vpn.0 table, if they are allowed by inbound policy. The route is installed, and a secondary route is imported into the instance.l2vpn.0 table when an import route target match between the route and instance is found.

LDP Signaling Behavior in VPWS in FEC 129 BGP Autodiscovery for VPWS

LDP listens for routes from instance.l2vpn.0 for any instance configured for FEC 129 VPWS. These routes are identified by the **instance-type l2vpn** statement in the routing instance and the presence of the **l2vpn-id** statement.

When a BGP autodiscovery route is installed, LDP sets up a targeted session with the remote peer, where the peer address is identified as the protocol next hop of the BGP autodiscovery route.

When a pseudowire route is installed in the instance.l2vpn.0 table, LDP uses the parameters associated with the route to signal the creation of the pseudowire using FEC 129. Upon receiving an FEC 129 label mapping message from a remote peer, LDP installs the pseudowire route in the ldp.l2vpn.0 table.

Upon a successful **l2vpn-id** match with a configured FEC 129 VPWS instance, a secondary pseudowire route is imported to the instance.l2vpn.0 table. If an outgoing pseudowire has not already been set up when the incoming pseudowire signaling is received, LDP initiates the outgoing pseudowire creation as well.

Related Documentation

- Understanding VPWS on page 3
- Example: Configuring FEC 129 BGP Autodiscovery for VPWS on page 73

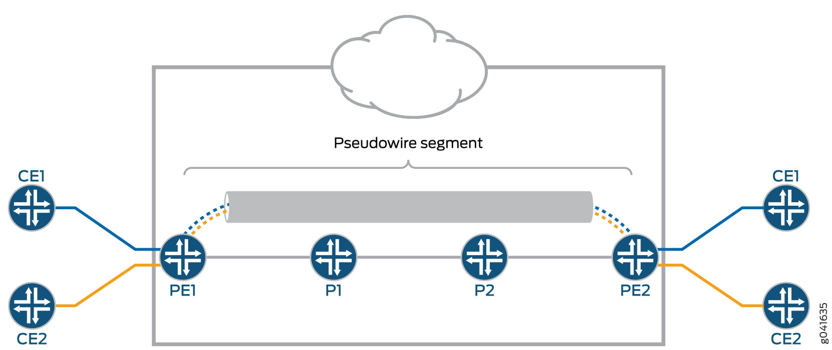
Understanding Multisegment Pseudowire for FEC 129

Understanding Multisegment Pseudowire

A pseudowire is a Layer 2 circuit or service that emulates the essential attributes of a telecommunications service, such as a T1 line, over an MPLS packet-switched network (PSN). The pseudowire is intended to provide only the minimum necessary functionality to emulate the wire with the required resiliency requirements for the given service definition.

When a pseudowire originates and terminates on the edge of the same PSN, the pseudowire label is unchanged between the originating and terminating provider edge (T-PE) devices. This is called a single-segment pseudowire (SS-PW). Figure 2 on page 8 illustrates an SS-PW established between two PE routers. The pseudowires between the PE1 and PE2 routers are located within the same autonomous system (AS).

Figure 2: L2VPN Pseudowire

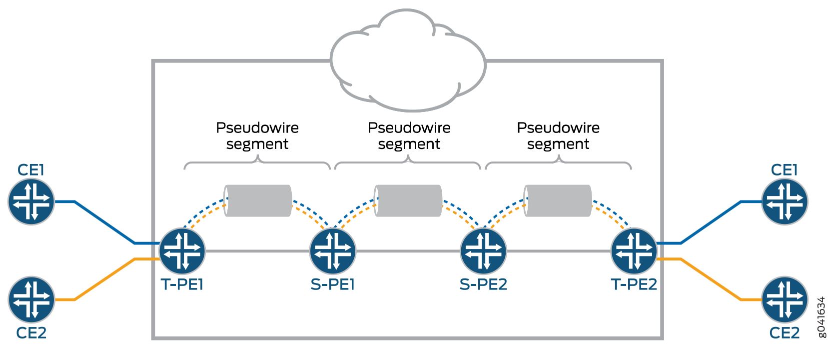


In cases where it is impossible to establish a single pseudowire from a local to a remote PE router, either because it is unfeasible or undesirable to establish a single control plane between the two PEs, a multisegment pseudowire (MS-PW) is used.

An MS-PW is a set of two or more contiguous SS-PWs that are made to function as a single point-to-point pseudowire. It is also known as switched pseudowire. MS-PWs can go across different regions or network domains. A region can be considered as an interior gateway protocol (IGP) area or a BGP autonomous system that belongs to the same or different administrative domain. An MS-PW spans multiple cores or ASs of the same or different carrier networks. A Layer 2 VPN MS-PW can include up to 254 pseudowire segments.

Figure 3 on page 9 illustrates a set of two or more pseudowire segments that function as a single pseudowire. The end routers are called terminating PE (T-PE) routers, and the switching routers are called switching PE (S-PE) routers. The S-PE router terminates the tunnels of the preceding and succeeding pseudowire segments in an MS-PW. The S-PE router can switch the control and data planes of the preceding and succeeding pseudowire segments of the MS-PW. An MS-PW is declared to be up when all the single-segment pseudowires are up.

Figure 3: Multisegment Pseudowire



Using FEC 129 for Multisegment Pseudowire

Currently, there are two types of attachment circuit identifiers (AIIs) defined under FEC 129:

- Type 1 AII
- Type 2 AII

The support of an MS-PW for FEC 129 uses type 2 AII. A type 2 AII is globally unique by definition of RFC 5003.

Single-segment pseudowires (SS-PWs) using FEC 129 on an MPLS PSN can use both type 1 and type 2 AIIs. For an MS-PW using FEC 129, a pseudowire itself is identified as a pair of endpoints. This requires that the pseudowire endpoints be uniquely identified.

In the case of a dynamically placed MS-PW, there is a requirement for the identifiers of attachment circuits to be globally unique, for the purposes of reachability and manageability of the pseudowire. Thus, individual globally unique addresses are allocated to all the attachment circuits and S-PEs that make up an MS-PW.

Type 2 AII is composed of three fields:

- Global_ID—Global identification, which is usually the AS number.
- Prefix—IPv4 address, which is usually the router ID.

- AC_ID—Local attachment circuit, which is a user-configurable value.

Since type 2 AII already contains the T-PE's IP address and it is globally unique from the FEC 129 pseudowire signaling point of view, the combination (AGI, SAII, TAII) uniquely identifies an MS-PW across all interconnected pseudowire domains.

Establishing a Multisegment Pseudowire Overview

An MS-PW is established by dynamically and automatically selecting the predefined S-PEs and placing the MS-PW between two T-PE devices.

When S-PEs are dynamically selected, each S-PE is automatically discovered and selected using the BGP autodiscovery feature, without the requirement of provisioning the FEC 129 pseudowire-related information on all the S-PEs. BGP is used to propagate pseudowire address information throughout the PSN.

Since there is no manual provisioning of FEC 129 pseudowire information on the S-PEs, the Attachment Group Identifier (AGI) and Attachment Individual Identifier (AII) are reused automatically, and choosing the same set of S-PEs for the pseudowire in both the forwarding and reverse direction is achieved through the active and passive role of each T-PE device.

- Active—The T-PE initiates an LDP label mapping message.
- Passive—The T-PE does not initiate an LDP label mapping message until it receives a label mapping message initiated by the active T-PE. The passive T-PE sends its label mapping message to the same S-PE from where it received the label mapping message originated from its active T-PE. This ensures that the same set of S-PEs are used in the reverse direction.

Pseudowire Status Support for Multisegment Pseudowire

- Pseudowire Status Behavior on T-PE on page 10
- Pseudowire Status Behavior on S-PE on page 11

Pseudowire Status Behavior on T-PE

The following pseudowire status messages are relevant on the T-PE:

- 0x00000010—Local PSN-facing pseudowire (egress) transmit fault.
- 0x00000001—Generic nonforwarding fault code. This is set as the local fault code. The local fault code is set at the local T-PE, and LDP sends a pseudowire status TLV message with the same fault code to the remote T-PE.
- Fault codes are bit-wise OR’ed and stored as remote pseudowire status codes.

Pseudowire Status Behavior on S-PE

The S-PE initiates the pseudowire status messages that indicate the pseudowire faults. The SP-PE in the pseudowire notification message hints where the fault was originated.

- When a local fault is detected by the S-PE, a pseudowire status message is sent in both directions along the pseudowire. Since there are no attachment circuits on an S-PE, only the following status messages are relevant:
  - 0x00000008—Local PSN-facing pseudowire (ingress) receive fault.
  - 0x00000010—Local PSN-facing pseudowire (egress) transmit fault.
- To indicate which SS-PW is at fault, an LDP SP-PE TLV is attached with the pseudowire status code in the LDP notification message. The pseudowire status is passed along from one pseudowire to another unchanged by the control plane switching function.
- If an S-PE initiates a pseudowire status notification message with one particular pseudowire status bit, then for the pseudowire status code an S-PE receives, the same bit is processed locally and not forwarded until the S-PE's original status error is cleared.
- An S-PE keeps only two pseudowire status codes for each SS-PW it is involved in – local pseudowire status code and remote pseudowire status code. The value of the remote pseudowire status code is the result of logic or operation of the pseudowire status codes in the chain of SS-PWs preceding this segment. This status code is incrementally updated by each S-PE upon receipt and communicated to the next S-PE. The local pseudowire status is generated locally based on its local pseudowire status.
- Only transmit fault is detected at the SP-PE. When there is no MPLS LSP to reach the next segment, a local transmit fault is detected. The transmit fault is sent to the next downstream segment, and the receive fault is sent to the upstream segment.
- Remote failures received on an S-PE are just passed along the MS-PW unchanged. Local failures are sent to both segments of the pseudowire that the S-PE is involved in.

Pseudowire TLV Support for MS-PW

MS-PW provides the following support for the LDP SP-PE TLV (RFC 6073):

- The LDP SP-PE TLVs for an MS-PW include:
  - Local IP address
  - Remote IP address
- An SP-PE adds the LDP SP-PE TLV to the label mapping message. Each SP-PE appends the local LDP SP-PE TLV to the SP-PE list it received from the other segment.
- The pseudowire status notification message includes the LDP SP-PE TLV when the notification is generated at the SP-PE.

Supported and Unsupported Features

Junos OS supports the following features with MS-PW:

- MPLS PSN for each SS-PW that builds up the MS-PW.
- The same pseudowire encapsulation for each SS-PW in an MS-PW – Ethernet or VLAN-CCC.
- The generalized PWid FEC with T-LDP as an end-to-end pseudowire signaling protocol to set up each SS-PW.
- MP-BGP to autodiscover the two endpoint PEs for each SS-PW associated with the MS-PW.
- Standard MPLS operation to stitch two side-by-side SS-PWs to form an MS-PW.
- Automatic discovery of S-PE so that the MS-PW can be dynamically placed.
- Minimum provisioning of S-PE.
- Operation, administration, and maintenance (OAM) mechanisms, including end-to-end MPLS ping or end-to-any-S-PE MPLS ping, MPLS path trace, end-to-end VCCV, and Bidirectional Forwarding Detection (BFD).
- Pseudowire switching point (SP) PE TLV for the MS-PW.
- Composite next hop on MS-PW.
- Pseudowire status TLV for MS-PW.

Junos OS does **not** support the following MS-PW functionality:

- Mix of LDP FEC 128 and LDP FEC 129.
- Static pseudowire where each label is provisioned staticially.
- Graceful Routing Engine switchover.
- Nonstop active routing.
- Multihoming.
- Partial connectivity verification (originating from an S-PE) in OAM.

Related Documentation

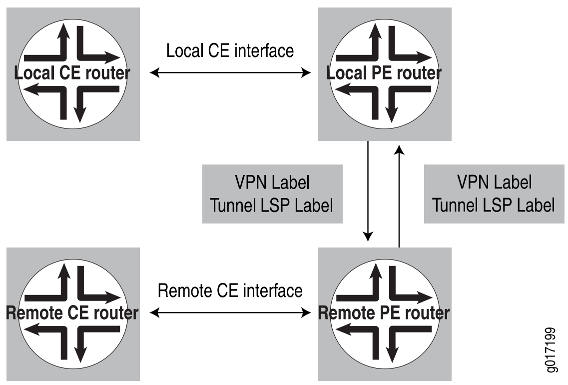
- Example: Configuring a Multisegment Pseudowire on page 89
- *Example: Configuring FEC 129 BGP Autodiscovery for VPWS*
- *Use Case for Configuring a Multisegment Pseudowire*

Layer 2 Circuit Overview

A Layer 2 circuit is a point-to-point Layer 2 connection transported using MPLS or other tunneling technology on the service provider’s network. A Layer 2 circuit is similar to a circuit cross-connect (CCC), except that multiple virtual circuits (VCs) are transported over a single shared label-switched path (LSP) tunnel between two provider edge (PE) routers. In contrast, each CCC requires a separate dedicated LSP.

The Junos OS implementation of Layer 2 circuits supports only the remote form of a Layer 2 circuit; that is, a connection from a local customer edge (CE) router to a remote CE router. Figure 4 on page 13 illustrates the components of a Layer 2 circuit.

Figure 4: Components of a Layer 2 Circuit



To establish a Layer 2 circuit, the Link Integrity Protocol (LIP) is used as the signaling protocol to advertise the ingress label to the remote PE routers. For this purpose, a targeted remote LDP neighbor session is established using the extended discovery mechanism described in LDP, and the session is brought up to the remote PE loopback IP address. Because LDP looks at the Layer 2 circuit configuration and initiates extended neighbor discovery for all the Layer 2 circuit neighbors (the remote PEs), no new configuration is necessary in LDP. Each Layer 2 circuit is represented by the logical interface connecting the local PE router to the local customer edge (CE) router. Note that LDP must be enabled on the lo0.0 interface for extended neighbor discovery to function correctly.

Packets are sent to remote CE routers over an egress VPN label advertised by the remote PE router, using a targeted LDP session. The VPN label is sent over an LDP LSP to the remote PE router connected to the remote CE router. Return traffic from the remote CE router destined to the local CE router is sent using an ingress VPN label advertised by the local PE router, which is also sent over the LDP LSP to the local PE router from the remote PE router.

Related Documentation

- *Layer 3 VPN Overview*
- *Layer 2 VPN Overview*
- *Layer 2 VPN Applications*
- Applications for Interconnecting a Layer 2 Circuit with a Layer 2 Circuit on page 176
- Applications for Interconnecting a Layer 2 Circuit with a Layer 3 VPN on page 192
- Example: Interconnecting a Layer 2 Circuit with a Layer 2 Circuit on page 176
- Example: Interconnecting a Layer 2 Circuit with a Layer 3 VPN on page 192
- Example: Interconnecting a Layer 2 Circuit with a Layer 2 VPN on page 167

Layer 2 Circuit Bandwidth Accounting and Call Admission Control

The sections that follow discuss Layer 2 circuit bandwidth accounting and call admission control (CAC):

Bandwidth Accounting and Call Admission Control Overview

Some network environments require that a certain level of service be guaranteed across the entire length of a path transiting a service provider's network. For Layer 2 circuits transiting an MPLS core network, a customer requirement might be to assure that guarantees for bandwidth and class of service (CoS) be maintained across the core network. For example, an Asynchronous Transfer Mode (ATM) circuit can provide service guarantees for each traffic class. A Layer 2 circuit configured to transport that ATM circuit across the network could be expected to provide the same service guarantees.

Providing this type of service guarantee requires the following:

- The LSPs in the MPLS core network must be able to provide service guarantees for bandwidth, rerouting, and route failures. You accomplish these guarantees by configuring multiclass LSPs. For more information about multiclass LSPs, see *Configuring Multiclass LSPs*.
- The service guarantee must be maintained across the entire length of the link as it transits the service provider's network. Different Layer 2 circuits could have different bandwidth requirements. However, many Layer 2 circuits could be transported over the same E-LSP in the MPLS core network.
- CAC ensures that the LSP has sufficient bandwidth to accommodate the Layer 2 circuit. If there is not enough bandwidth over a particular LSP, the Layer 2 circuit is prevented from using that LSP.

Selecting an LSP Based on the Bandwidth Constraint

CAC of Layer 2 circuits is based on the bandwidth constraint. You must configure this constraint for each Layer 2 circuit interface. If there is a bandwidth constraint configured for a Layer 2 circuit, CAC bases the final selection of which LSP-forwarding next hop to use on the following:

- If multiple LSPs meet the bandwidth requirements, the first LSP found that can satisfy the bandwidth requirements for the Layer 2 circuit is selected.
- If there is more than one next hop mapped to the same LSP, then all the next hops that map to that LSP and pass CAC constraints are installed. This allows the Layer 2 circuit routes to restore themselves quickly in case of failure.

- The available bandwidth on the selected LSP is decremented by the bandwidth required for each Layer 2 circuit. Similarly, when the Layer 2 circuit route is changed or deleted (for example, when the route is disassociated from that particular LSP), the bandwidth on the corresponding LSP is incremented.
- There are no priorities among different Layer 2 circuits competing for the same LSP next hop in the core network.
- When an LSP's bandwidth changes, the Layer 2 circuits using that LSP repeat the CAC process again.

  If the LSP bandwidth increases, some Layer 2 circuits that were not established might now successfully resolve over the LSP. Similarly, if the bandwidth of the LSP decreases, some Layer 2 circuits that were previously up might now be declared down because of insufficient bandwidth on the LSP.
- When no LSP is found to meet the bandwidth requirements of the Layer 2 circuit, it is considered to be a CAC failure, and an error is reported.

LSP Path Protection and CAC

CAC can take into account LSPs that have been configured with an MPLS path protection feature, such as secondary paths, fast reroute, or node and link protection. CAC can consider the bandwidth available on these auxiliary links and can accept the backup connection as valid if the main connection fails. However, there are limitations on how the path protection feature must be configured to prevent CAC from taking down the Layer 2 circuit when the LSP it is using is switched to a backup route.

For more information about MPLS path protection features, see the *MPLS and Traffic Protection*.

The sections that follow discuss the path protection features that can be used in conjunction with CAC and how they must be configured:

- Secondary Paths and CAC on page 15
- Fast Reroute and CAC on page 16
- Link and Node Protection and CAC on page 16

Secondary Paths and CAC

The following describes the ways in which secondary paths would interact with Layer 2 circuit CAC:

- If an LSP is configured with both primary and secondary paths, if the paths have the same bandwidth, and if this bandwidth is enough to accommodate the Layer 2 circuit, the Layer 2 circuit route installs both next hops in the forwarding table.

  CAC allows the Layer 2 circuit to be switched to the secondary path if the primary path fails.
- If the LSP has primary and secondary paths configured with different bandwidths, each path must run through CAC independently. If the active path for that LSP passes CAC constraints successfully, then that next hop is installed and the corresponding LSP is

selected to transport the Layer 2 circuit traffic. The LSP's secondary paths are then checked for CAC, and installed if there is sufficient bandwidth.

However, if the active path for the LSP fails to meet the CAC constraints, then that LSP is not selected and the system looks for a different LSP to transport the Layer 2 circuit.

For example, an LSP has an active primary path with 30 megabits of bandwidth and a secondary path with 10 megabits of bandwidth. The Layer 2 circuit requires 15 megabits of bandwidth. The secondary path fails CAC, and only the next hop corresponding to the primary path is installed for the Layer 2 circuit route. The path protection originally provided by the secondary path is no longer available.

Fast Reroute and CAC

No CAC is done for fast reroute detours. However, as long as the protected path satisfies the CAC bandwidth constraints, the detour next hop is also selected and installed.

Link and Node Protection and CAC

You can configure CAC on Layer 2 circuit-based LSPs with bandwidth constraints and also enable link and node protection. However, if the primary LSP fails, CAC might not be applied to the bypass LSP, meaning the bypass LSP might not meet the bandwidth constraint for the Layer 2 circuit. To minimize the risk of loosing traffic, the Layer 2 circuit continues to use the non-CAC bypass LSP while an attempt is made to establish a new Layer 2 circuit route over an LSP that does support CAC.

Layer 2 Circuits Trunk Mode

Using Layer 2 circuit trunk mode, you can configure Layer 2 circuits to carry ATM trunks, providing a way to link ATM switches over an MPLS core network.

Layer 2 circuit trunk mode allows you to configure the following CoS features:

- CoS queues in Layer 2 circuit trunk mode—For ATM2 IQ interfaces, you can configure ATM CoS queues for Layer 2 circuit trunk mode.
- Layer 2 circuit trunk mode scheduling—For ATM2 IQ interfaces configured to use Layer 2 circuit trunk mode, you can share a scheduler among 32 trunks on an ATM port.
- Two early packet discard (EPD) thresholds per queue—For ATM2 IQ interfaces configured to use Layer 2 circuit trunk mode, you can set two EPD thresholds that depend on the packet-loss priorities (PLPs) of the packets.

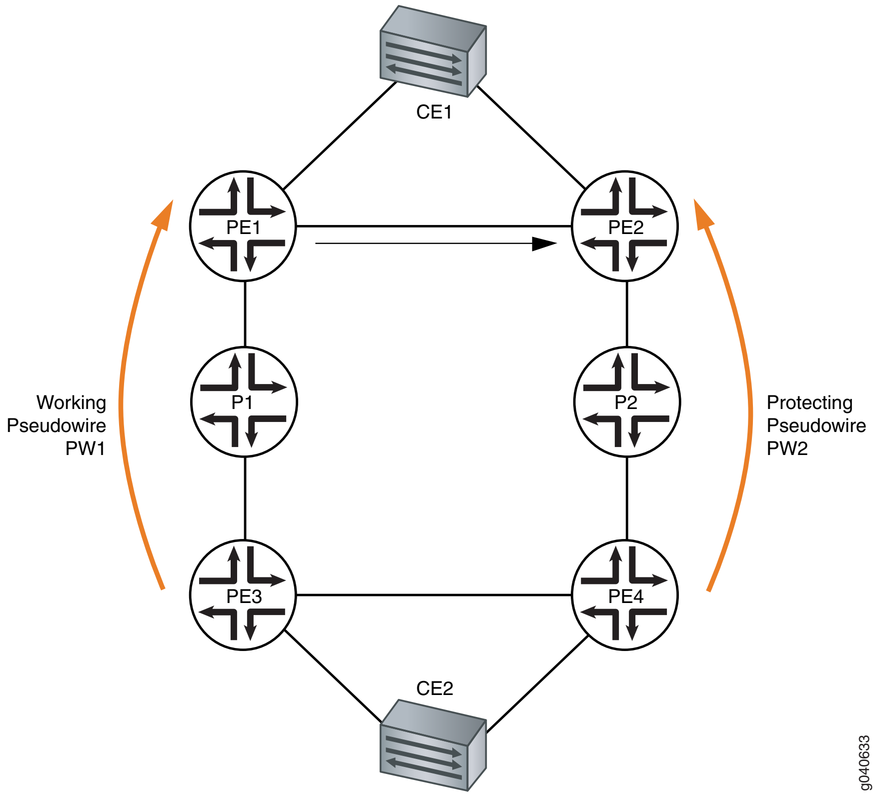
For a detailed overview and configuration documentation, see the *ATM Interfaces Feature Guide for Routing Devices* and *Junos OS Class of Service Library for Routing Devices*.

Related Documentation
- *MPLS and Traffic Protection*

Egress Protection LSPs for Layer 2 Circuits

An egress protection LSP provides link protection for link between PE routers and CE devices as illustrated in Figure 5 on page 17.

Figure 5: Egress Protection LSP



Device CE1 is multihomed to router PE1 and router PE2. Device CE2 is multihomed to router PE3 and router PE4. There are two paths connecting devices CE1 and CE2. The working path is CE2-PE3-P1-PE1-CE1, using pseudowire PW1. The protecting path is CE2-PE4-P2-PE2-CE1, using pseudowire PW2. Normally, traffic flows through the working path. When the end-to-end OAM between devices CE1 and CE2 detects a failure on the working path, traffic will be switched from the working path to the protecting path.

In the topology shown in Figure 5 on page 17, if there was a link or node failure in the core network (for example, a link failure from router P1 to PE1, from router PE3 to P1, or a node failure of router P1), MPLS fast reroute can be triggered on the transport LSPs between router PE3 and router PE1 to repair the connection within tens of milliseconds. Egress protection LSPs address the problem of when a link failure occurs at the edge of the network (for example, a link failure on router PE1 to device CE1).

An egress protection LSP has been configured from router PE1 to router PE2. In the event of a link failure between router PE1 and device CE1, traffic can be switched to the egress protection LSP. Traffic from device CE2 can now be routed through path PE3-P1-PE1-PE2 to reach device CE1.

FAT Flow Labels Overview

A pseudowire is a Layer 2 circuit or service that emulates the essential attributes of a telecommunications service, such as a T1 line, over an MPLS packet-switched network (PSN). The pseudowire is intended to provide only the minimum necessary functionality to emulate the wire with the required resiliency requirements for the given service definition.

In an MPLS network, the flow-aware transport of pseudowires (FAT) flow label, as described in RFC 6391, *Flow-Aware Transport of Pseudowires over an MPLS Packet Switched Network*, is used for load-balancing traffic across LDP-signaled pseudowires for virtual private LAN service (VPLS) and virtual private wire service (VPWS).

When the pseudowire is configured to use the FAT flow labels for load balancing, packets arriving at the ingress router are processed in the following sequence across the path of the pseudowire:

- The ingress router uses the contents of the inbound packet in the hash-key algorithm to calculate the flow-label value.
- The ingress router pushes the flow label to the label stack of the packet.
- The transit routers perform load balancing based only on the label stack.
- The egress router pops the flow label and forwards the packet to its destination.

For load balancing to work based on a flow-label configuration, a version of LDP that supports signaling extensions to use the flow label with pseudowires must be enabled on all routers. The LDP-signaling configuration is identical for VPLS and VPWS pseudowires.

FAT flow labels are supported on the following LDP-signaled forwarding-equivalence classes (FECs) for VPWS and VPLS pseudowires:

- FEC 128 for VPWS—LDP-signaled VPWS with neighbors that are statically configured (BGP autodiscovery is not supported).
- FEC 128 for VPLS—LDP-signaled VPLS with neighbors that are statically configured (BGP autodiscovery is not supported).
- FEC 129 for VPWS—LDP-signaled VPWS with BGP autodiscovery of neighbors.
- FEC 129 for VPLS—LDP-signaled VPLS with BGP autodiscovery of neighbors.

The interface parameter (Sub-TLV) is used both for FEC 128 and FEC 129 pseudowires. The sub-TLV defined for LDP contains the transmit (T) and receive (R) bits. The T bit advertises the ability to push the flow label. The R bit advertises the ability to pop the flow labels. By default, the signaling behavior of the provider edge (PE) router for any of these pseudowires is to advertise the T and R bits in the label set to 0.

The **flow-label-transmit** and **flow-label-receive** configuration statements provide the ability to set the T bit and R bit advertisement to 1 in the Sub-TLV field, which is part of the interface parameters of the FEC for the LDP label-mapping message. You can use these statements to control the pushing of the load-balancing label and the advertisement of the label to the routing peers in the control plane.

Alternatively, for FEC 128 VPWS pseudowires only, you can configure the following statements to statically configure flow label push and pop operations:

- **flow-label-receive-static** to statically pop the flow label on the pseudowire packets received from the remote PE router.
- **flow-label-transmit-static** to statically push the flow label on the pseudowire packets sent to the remote PE router.

Related Documentation

- Configuring the FAT Flow Label for FEC 128 VPWS Pseudowires for Load-Balancing MPLS Traffic on page 67
- *Configuring the FAT Flow Label for FEC 128 VPLS Pseudowires for Load-Balancing MPLS Traffic*
- Configuring the FAT Flow Label for FEC 129 VPWS Pseudowires for Load-Balancing MPLS Traffic on page 69
- *Configuring the FAT Flow Label for FEC 129 VPLS Pseudowires for Load-Balancing MPLS Traffic*

CHAPTER 2

Introduction to Configuring Layer 2 Circuits

Configuring LDP for Layer 2 Circuits

Use LDP as the signaling protocol to advertise ingress labels to the remote PE routers. When configured, LDP examines the Layer 2 circuit configuration and initiates extended neighbor discovery for all the Layer 2 circuit neighbors (for example, remote PEs). This process is similar to how LDP works when tunneled over RSVP. You must run LDP on the **lo0.0** interface for extended neighbor discovery to function correctly.

For detailed information about how to configure LDP, see the *Junos OS MPLS Applications Library for Routing Devices*.

PART 2

Configuration

CHAPTER 3

Configuring Layer 2 Circuits

Configuring BFD for VCCV for Layer 2 Circuits

Bidirectional Forwarding Detection (BFD) support for virtual circuit connection verification (VCCV) allows you to configure a control channel for a pseudowire, in addition to the corresponding operations and management functions to be used over that control channel. BFD provides a low resource mechanism for the continuous monitoring of the pseudowire data path and for detecting data plane failures. This feature provides support for asynchronous mode BFD for VCCV as described in RFC 5885, *Bidirectional Forwarding Detection (BFD) for the Pseudowire Virtual Circuit Connectivity Verification (VCCV)*. You can also use a ping to detect pseudowire failures. However, the processing resources required for a ping are greater than what is needed for BFD. In addition, BFD is capable of detecting data plane failure faster than VCCV ping. BFD for pseudowires is supported for Layer 2 circuits (LDP-based).

Before you begin:

- Configure the device interfaces.

To configure BFD for VCCV:

1. Specify the threshold for the adaptation of the BFD session detection time.

   [edit protocols l2circuit neighbor *IP-address* interface *interface-name* oam bfd-liveness-detection]
   user@host# **set detection-time threshold *milliseconds***

   For example, to set a detection time threshold of 40 milliseconds for OAM BFD liveness detection:

   [edit protocols l2circuit neighbor 120.120.120.120 interface ge-1/1/9.0 oam bfd-liveness-detection]
   user@host# **set detection-time threshold 40**

2. Configure the virtual circuit ID for the Layer 2 circuit protocol.

   [edit protocols l2circuit neighbor *IP-address* interface *interface-name*]
   user@host# **set virtual-circuit-id *virtual-circuit-id***

   For example, to set the virtual circuit ID as 1 for OAM BFD liveness detection:

   [edit protocols l2circuit neighbor 120.120.120.120 interface ge-1/1/9.0 oam bfd-liveness-detection]
   user@host# **set virtual-circuit-id 1**

3. Configure the minimum interval after which the local routing device transmits hello packets and then expects to receive a reply from a neighbor with which it has established a BFD session for the Layer 2 circuit.

   [edit protocols l2circuit neighbor *IP-address* interface *interface-name* oam bfd-liveness-detection]
   user@host# **set minimum-interval *milliseconds***

   For example, to set a minimum interval of 300 milliseconds for OAM BFD liveness detection:

   [edit protocols l2circuit neighbor 120.120.120.120 interface ge-1/1/9.0 oam bfd-liveness-detection] user@host# **set minimum-interval 300**

4. Configure the minimum interval after which the local routing device must receive a reply from a neighbor with which it has established a BFD session for the Layer 2 circuit protocol.

   [edit protocols l2circuit neighbor *IP-address* interface *interface-name* oam
     bfd-liveness-detection]
   user@host# **set minimum-receive-interval** ***milliseconds***

   For example, to set a minimum receive interval of 10 milliseconds for OAM BFD liveness detection:

   [edit protocols l2circuit neighbor 120.120.120.120 interface ge-1/1/9.0 oam
     bfd-liveness-detection]
   user@host# **set minimum-receive-interval 10**

5. Configure the number of hello packets not received by a neighbor that causes the originating interface to be declared down for the Layer 2 circuit protocol.

   [edit protocols l2circuit neighbor *IP-address* interface *interface-name* oam
     bfd-liveness-detection]
   user@host# **set multiplier** ***number***

   For example, to set the multiplier as 3 for OAM BFD liveness detection:

   [edit protocols l2circuit neighbor 120.120.120.120 interface ge-1/1/9.0 oam
     bfd-liveness-detection]
   user@host# **set multiplier 3**

6. Configure to disable adaptation.

   [edit protocols l2circuit neighbor *IP-address* interface *interface-name* oam
     bfd-liveness-detection]
   user@host# **set no-adaptation**

7. Configure the minimum interval at which the local routing device transmits hello packets to a neighbor with which it has established a BFD session.

   [edit protocols l2circuit neighbor *IP-address* interface *interface-name* oam
     bfd-liveness-detection transmit-interval]
   user@host# **set minimum-interval** ***milliseconds***

   For example, to set a minimum transmit interval of 5 milliseconds for OAM BFD liveness detection:

   [edit protocols l2circuit neighbor 120.120.120.120 interface ge-1/1/9.0 oam
     bfd-liveness-detection transmit-intreval]
   user@host# **set minimum-interval 5**

8. Specify the threshold for the adaptation of the BFD session transmit interval.

   [edit protocols l2circuit neighbor *IP-address* interface *interface-name* oam
     bfd-liveness-detection transmit-interval]
   user@host# **set threshold** ***milliseconds***

   For example, to set a transmit interval threshold of 30 milliseconds for OAM BFD liveness detection:

   [edit protocols l2circuit neighbor 120.120.120.120 interface ge-1/1/9.0 oam
     bfd-liveness-detection transmit-intreval]
   user@host# **set threshold 30**

Related Documentation

- Example: Configuring BFD for VCCV for Layer 2 Circuits on page 131

Configuring an IGP on the PE and P Routers

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

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

Related Documentation

- *Example: Configuring IS-IS*
- *Examples: Configuring Static Routes*
- *OSPF Feature Guide for Routing Devices*

Configuring IBGP Sessions Between PE Routers in VPNs

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

Configure an IBGP session for the VPN as follows:

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

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

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

The **family** statement allows you to configure the IBGP session for Layer 2 VPNs, VPLS, EVPNs or for Layer 3 VPNs.

- To configure an IBGP session for Layer 2 VPNs and VPLS, include the **signaling** statement at the **[edit protocols bgp group *group-name* family l2vpn]** hierarchy level:

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

- To configure an IBGP session for EVPNs, include the **signaling** statement at the **[edit protocols bgp group *group-name* family evpn]** hierarchy level:

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

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

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

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

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

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

Related Documentation

- Configuring an IGP on the PE and P Routers on page 28
- Configuring a Signaling Protocol and LSPs for VPNs on page 29

Configuring a Signaling Protocol and LSPs for VPNs

For VPNs to function, you must enable a signaling protocol, either the LDP or RSVP on the provider edge (PE) routers and on the provider (P) routers. You also need to configure label-switched paths (LSPs) between the ingress and egress routers. In a typical VPN configuration, you need to configure LSPs from each PE router to all of the other PE routers participating in the VPN in a full mesh.

NOTE: As with any configuration involving MPLS, you cannot configure any of the core-facing interfaces on the PE routers over dense Fast Ethernet PICs.

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

- Using LDP for VPN Signaling on page 30
- Using RSVP for VPN Signaling on page 31

Using LDP for VPN Signaling

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

1. Configure LDP on the interfaces in the core of the service provider's network by including the **ldp** statement at the **[edit protocols]** hierarchy level.

   You need to configure LDP only on the interfaces between PE routers or between PE and P routers. You can think of these as the "core-facing" interfaces. You do not need to configure LDP on the interface between the PE and customer edge (CE) routers.

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

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

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

3. Configure OSPF or IS-IS on each PE and P router.

   You configure these protocols at the master instance of the routing protocol, not within the routing instance used for the VPN.

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

     ```
     [edit]
     protocols {
       ospf {
         area 0.0.0.0 {
           interface type-fpc/pic/port;
     ```



```
      }
    }
  }
```

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

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

For more information about configuring OSPF and IS-IS, see the *OSPF Feature Guide for Routing Devices* and *IS-IS Feature Guide for Routing Devices*.

Using RSVP for VPN Signaling

To use RSVP for VPN signaling, perform the following steps:

1. On each PE router, configure traffic engineering.

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

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

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

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

2. On each PE and P router, enable RSVP on the interfaces that participate in the label-switched path (LSP).

   On the PE router, these interfaces are the ingress and egress points to the LSP. On the P router, these interfaces connect the LSP between the PE routers. Do not enable RSVP on the interface between the PE and the CE routers, because this interface is not part of the LSP.

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

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

3. On each PE router, configure an MPLS LSP to the PE router that is the LSP's egress point.

   To do this, include the **interface** and **label-switched-path** statements at the **[edit protocols mpls]** hierarchy level:

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

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

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

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

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

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

   Include one **interface** statement for each connection to the LSP.

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

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

5. Enable MPLS on the interface between the PE and CE routers by including the **interface** statement at the **[edit mpls]** hierarchy level.

   Doing this allows the PE router to assign an MPLS label to traffic entering the LSP or to remove the label from traffic exiting the LSP.

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

   For information about configuring MPLS, see the *Minimum MPLS Configuration*.

Related Documentation

- *Minimum MPLS Configuration*9

Configuring Routing Instances on PE Routers in VPNs

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

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

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

You can include these statements at the following hierarchy levels:

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

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

- Configuring the Routing Instance Name for a VPN on page 34
- Configuring the Description on page 34
- Configuring the Instance Type on page 34
- Configuring Interfaces for VPN Routing on page 35

- Configuring the Route Distinguisher on page 37
- Configuring Automatic Route Distinguishers on page 38

Configuring the Routing Instance Name for a VPN

The name of the routing instance for a VPN can be a maximum of 128 characters and can contain letters, numbers, and hyphens. In Junos OS Release 9.0 and later, you can no longer specify **default** as the actual routing-instance name. You also cannot use any special characters (! @ # $ % ^ & * , +< > : ;) within the name of a routing instance.

NOTE: In Junos OS Release 9.6 and later, you can include a slash (/) in a routing instance name only if a logical system is not configured. That is, you cannot include the slash character in a routing instance name if a logical system other than the default is explicitly configured.

Specify the routing-instance name with the **routing-instance** statement:

routing-instance *routing-instance-name* {...}

You can include this statement at the following hierarchy levels:

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

Configuring the Description

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

To configure a text description, include the **description** statement:

description *text*;

You can include this statement at the following hierarchy levels:

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

Configuring the Instance Type

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

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

  instance-type l2vpn;

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

  instance-type vpls;

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

  instance-type vrf;

  NOTE: Routing Engine based sampling is not supported on VRF routing instances.

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

  instance-type virtual-router;

You can include this statement at the following hierarchy levels:

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

Configuring Interfaces for VPN Routing

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

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

- General Configuration for VPN Routing on page 35
- Configuring Interfaces for Layer 3 VPNs on page 36
- Configuring Interfaces for Carrier-of-Carriers VPNs on page 36
- Configuring Unicast RPF on VPN Interfaces on page 37

General Configuration for VPN Routing

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

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

interface *interface-name*;

You can include this statement at the following hierarchy levels:

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

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

physical.logical

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

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

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

Configuring Interfaces for Layer 3 VPNs

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

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

Configuring Interfaces for Carrier-of-Carriers VPNs

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

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

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

Configuring Unicast RPF on VPN Interfaces

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

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

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

For information about how to configure unicast RPF on VPN interfaces, see *Configuring Unicast RPF*.

Configuring the Route Distinguisher

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

For Layer 2 VPNs and VPLS, if you have configured the **l2vpn-use-bgp-rules** statement, you must configure a unique route distinguisher for each PE router participating in a specific routing instance.

For other types of VPNs, we recommend that you use a unique route distinguisher for each PE router participating in the routing instance. Although you can use the same route distinguisher on all PE routers for the same VPN routing instance (except for Layer 2 VPNs and VPLS), if you use a unique route distinguisher, you can determine the CE router from which a route originated within the VPN.

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

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

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

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

- ***as-number*:*number***, where ***as-number*** is an autonomous system (AS) number (a 2-byte value) and ***number*** is any 4-byte value. The AS number can be in the range 1 through 65,535. We recommend that you use an Internet Assigned Numbers Authority (IANA)-assigned, nonprivate AS number, preferably the Internet service provider's (ISP's) own or the customer's own AS number.

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

Configuring Automatic Route Distinguishers

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

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

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

You can include this statement at the following hierarchy levels:

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

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

Related Documentation

- Configuring Policies for the VRF Table on PE Routers in VPNs on page 38
- *Configuring BGP Route Target Filtering for VPNs*

Configuring Policies for the VRF Table on PE Routers in VPNs

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

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

Configuring the Route Target

As part of the policy configuration for the VPN routing table, you must define a route target, which defines which VPN the route is a part of. When you configure different types of VPN services (Layer 2 VPNs, Layer 3 VPNs, EVPNs, or VPLS) on the same PE router, be sure to assign unique route target values to avoid the possibility of adding route and signaling information to the wrong VPN routing table.

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

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

You can include this statement at the following hierarchy levels:

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

name is the name of the community.

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

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

Configuring the Route Origin

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

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

To configure a route origin, complete the following steps:

1. Include the **community** statement with the **origin** option:

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

   You can include this statement at the following hierarchy levels:

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

   name is the name of the community.

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

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

2. Include the community in the import policy for the PE router's VRF table by configuring the **community** statement with the ***community-id*** identifier defined in Step 1 at the **[edit policy-options policy-statement *import-policy-name* term *import-term-name* from]** hierarchy level. See "Configuring an Import Policy for the PE Router's VRF Table" on page 40.

   If the policy's **from** clause does not specify a community condition, the **vrf-import** statement in which the policy is applied cannot be committed. The Junos OS commit operation does not pass the validation check.

3. Include the community in the export policy for the PE router's VRF table by configuring the **community** statement with the ***community-id*** identifier defined in Step 1 at the **[edit policy-options policy-statement *export-policy-name* term *export-term-name* then]** hierarchy level. See "Configuring an Export Policy for the PE Router's VRF Table" on page 42.

See *Route Origin for VPNs* for a configuration example.

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

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

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

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

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

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

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

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

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

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

   For more information about creating policies, see the *Routing Policy Feature Guide for Routing Devices*.

2. You can optionally use a regular expression to define a set of communities to be used for the VRF import policy.

   For example you could configure the following using the **community** statement at the **[edit policy-options policy-statement *policy-statement-name*]** hierarchy level:

```
[edit policy-options vrf-import-policy-sample]
community high-priority members *:50
```

   Note that you cannot configure a regular expression as a part of a route target extended community. For more information about how to configure regular expressions for communities, see *Understanding How to Define BGP Communities and Extended Communities* .

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

```
vrf-import import-policy-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*]**

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

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

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

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

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

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

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

   NOTE: Configuring the **community add** statement is a requirement for Layer 2 VPN VRF export policies. If you change the **community add** statement to the **community set** statement, the router at the egress of the Layer 2 VPN link might drop the connection.

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

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

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

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

The ***export-policy-name*** policy evaluates all routes received over the routing protocol session with the CE router. (This session can use the BGP, OSPF, or RIP routing protocols, or static routes.) If the routes match the conditions in the **from** statement, the community target specified in the **then community add** statement is added to them and they are exported to the remote PE routers. The second term in the policy rejects all other routes.

For more information about creating policies, see the *Routing Policy Feature Guide for Routing Devices*.

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

   vrf-export *export-policy-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*]**

Applying Both the VRF Export and the BGP Export Policies

When you apply a VRF export policy as described in "Configuring an Export Policy for the PE Router's VRF Table" on page 42, routes from VPN routing instances are advertised to other PE routers based on this policy, whereas the BGP export policy is ignored.

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

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

- Routes imported into the l3vpn.bgp.0 routing table retain the attributes of the original routes (for example, an OSPF route remains an OSPF route even when it is stored in the l3vpn.bgp.0 routing table). You should be aware of this when you configure an export policy for connections between an IBGP PE router and a PE router, a route reflector and a PE router, or AS boundary router (ASBR) peer routers.

- By default, all routes in the l3vpn.bgp.0 routing table are exported to the IBGP peers. If the last statement of the export policy is deny all and if the export policy does not specifically match on routes in the l3vpn.bgp.0 routing table, no routes are exported.

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

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

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

Configuring a VRF Target

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

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

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

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

```
vrf-target community;
```

You can include this statement at the following hierarchy levels:

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

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

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

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

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



```
    import community-name;
}
```

You can include this statement at the following hierarchy levels:

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

Configuring Local Interface Switching in Layer 2 Circuits

You can configure a virtual circuit entirely on the local router, terminating the circuit on a local interface. Possible uses for this feature include being able to enable switching between Frame Relay DLCIs.

To configure a virtual circuit to terminate locally, include the **local-switching** statement:

```
local-switching {
    interface interface-name {
        description text;
        end-interface {
            interface interface-name;
            no-revert;
            protect-interface interface-name;
        }
        ignore-mtu-mismatch;
        no-revert;
        protect-interface interface-name;
    }
}
```

You can include this statement at the following hierarchy levels:

- **[edit protocols l2circuit]**
- **[edit logical-systems *logical-system-name* protocols l2circuit]**

The following sections describe how to configure local interface switching:

- Configuring the Interfaces for the Local Interface Switch on page 45
- Enabling Local Interface Switching When the MTU Does Not Match on page 46

Configuring the Interfaces for the Local Interface Switch

Local interface switching requires you to configure at least two interfaces:

- Starting interface—Include the **interface** statement at the **[edit protocols l2circuit local-switching]** hierarchy level.
- Ending interface—Include the **end-interface** statement at the **[edit protocols l2circuit local-switching interface *interface-name*]** hierarchy level.

You can also configure virtual circuit interface protection for each local interface:

- Protect interface for the starting interface—Include the **protect-interface** statement at the **[edit protocols l2circuit local-switching interface *interface-name*]** hierarchy level.
- Protect interface for the ending interface—Include the **protect-interface** statement at the **[edit protocols l2circuit local-switching interface *interface-name* end-interface]** hierarchy level.

For more information about how to configure protect interfaces, see "Configuring the Protect Interface" on page 50.

Typically, when the primary interface goes down, the pseudowire starts using the protect interface. By default, when the primary interface comes back online, the interface is switched-over back from the protect interface to the primary interface. To prevent the switchover back to the primary interface, unless the primary interface goes down, include the **no-revert** statement. This prevents loss of traffic during the switchover.

NOTE: If the protect interface fails, the interface is switched-over back to the primary interface, irrespective of whether or not the **no-revert** statement is included in the configuration.

You can configure the **no-revert** statement both for the starting interface and the ending interface.

```
[edit protocols l2circuit local-switching interface interface-name]
no-revert;
end-interface {
  interface interface-name;
  no-revert;
}
```

NOTE: The protect interface must be configured prior to configuring the **no-revert** statement.

Enabling Local Interface Switching When the MTU Does Not Match

You can configure a local switching interface to ignore the MTU configuration set for the associated physical interface. This enables you to bring up a circuit between two logical interfaces that are defined on physical interfaces with different MTU values.

To configure the local switching interface to ignore the MTU configured for the physical interface, include the **ignore-mtu-mismatch** statement:

```
ignore-mtu-mismatch;
```

You can include this statement at the following hierarchy levels:

- **[edit protocols l2circuit local-switching interface *interface-name*]**

- **[edit logical-systems *logical-system-name* protocols l2circuit local-switching interface *interface-name*]**

Configuring Interfaces for Layer 2 Circuits

The following sections describe how to configure interfaces for Layer 2 circuits:

Configuring the Address for the Neighbor of the Layer 2 Circuit

All the Layer 2 circuits using a particular remote PE router designated for remote CE routers are listed under the **neighbor** statement ("neighbor" designates the PE router). Each neighbor is identified by its IP address and is usually the end-point destination for the label-switched path (LSP) tunnel transporting the Layer 2 circuit.

To configure a PE router as a neighbor for a Layer 2 circuit, specify the neighbor address using the **neighbor** statement:

```
neighbor address {
  ...
}
```

You can include this statement at the following hierarchy levels:

- **[edit protocols l2circuit]**
- **[edit logical-systems *logical-system-name* protocols l2circuit]**

Configuring the Neighbor Interface for the Layer 2 Circuit

Each Layer 2 circuit is represented by the logical interface connecting the local provider edge (PE) router to the local customer edge (CE) router. This interface is tied to the Layer 2 circuit neighbor configured in "Configuring the Address for the Neighbor of the Layer 2 Circuit" on page 47.

To configure the interface for a Layer 2 circuit neighbor, include the **interface** statement:

NOTE: The commit operation fails, if the same logical interface is configured for both Layer 2 circuit and ccc connection.

```
interface interface-name {
  bandwidth (bandwidth | ctnumber bandwidth);
  community community-name;
  (control-word | no-control-word);
  description text;
  encapsulation-type type;
  ignore-encapsulation-mismatch;
  ignore-mtu-mismatch;
```

```
    mtu mtu-number;
    no-revert;
    protect-interface interface-name;
    pseudowire-status-tlv;
    psn-tunnel-endpoint address;
    virtual-circuit-id identifier;
  }
```

You can include this statement at the following hierarchy levels:

- **[edit protocols l2circuit neighbor *address*]**
- **[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address*]**

The following sections describe how to configure the interface for the Layer 2 circuit neighbor:

Configuring a Community for the Layer 2 Circuit

To configure a community for a Layer 2 circuit, include the **community** statement:

```
community community-name;
```

You can include this statement at the following hierarchy levels:

- **[edit protocols l2circuit neighbor *address* interface *interface-name*]**
- **[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*]**

For information about how to configure a routing policy for a Layer 2 circuit, see "Configuring Policies for Layer 2 Circuits" on page 57.

Configuring the Control Word for Layer 2 Circuits

To emulate the virtual circuit (VC) encapsulation for Layer 2 circuits, a 4-byte control word is added between the Layer 2 protocol data unit (PDU) being transported and the VC label that is used for demultiplexing. For most protocols, a null control word consisting of all zeroes is sent between Layer 2 circuit neighbors.

However, individual bits are available in a control word that can carry Layer 2 protocol control information. The control information is mapped into the control word, which allows the header of a Layer 2 protocol to be stripped from the frame. The remaining data and control word can be sent over the Layer 2 circuit, and the frame can be reassembled with the proper control information at the egress point of the circuit.

The following Layer 2 protocols map Layer 2 control information into special bit fields in the control word:

- Frame Relay—The control word supports the transport of discard eligible (DE), forward explicit congestion notification (FECN), and backward explicit congestion notification (BECN) information. For configuration information, see “Configuring the Control Word for Frame Relay Interfaces” on page 49.

NOTE: **Frame Relay is not supported on the ACX Series routers.**

- ATM AAL5 mode—The control word supports the transport of sequence number processing, ATM cell loss priority (CLP), and explicit forward congestion indication (EFCI) information. When you configure an AAL5 mode Layer 2 circuit, the control information is carried by default and no additional configuration is needed.
- ATM cell-relay mode—The control word supports sequence number processing only. When you configure a cell-relay mode Layer 2 circuit, the sequence number information is carried by default and no additional configuration is needed.

The Junos OS implementation of sequence number processing for ATM cell-relay mode and AAL5 mode is not the same as that described in Sec. 3.1.2 of the IETF draft *Encapsulation Methods for Transport of Layer 2 Frames Over IP and MPLS Networks*. The differences are as follows:

- A packet with a sequence number of 0 is considered as out of sequence.
- A packet that does not have the next incremental sequence number is considered out of sequence.
- When out-of-sequence packets arrive, the sequence number in the Layer 2 circuit control word increments by one and becomes the expected sequence number for the neighbor.

The following sections discuss how to configure the control word for Layer 2 circuits:

Configuring the Control Word for Frame Relay Interfaces

On interfaces with Frame Relay CCC encapsulation, you can configure Frame Relay control bit translation to support Frame Relay services over IP and MPLS backbones by using CCC, Layer 2 VPNs, and Layer 2 circuits. When you configure translation of Frame Relay control bits, the bits are mapped into the Layer 2 circuit control word and preserved across the IP or MPLS backbone.

For information about how to configure the control bits, see the *Junos OS Network Interfaces Library for Routing Devices* and the *Junos OS, Release 14.1*.

Disabling the Control Word for Layer 2 Circuits

The Junos OS can typically determine whether a neighboring router supports the control word. However, if you want to explicitly disable its use on a specific interface, include the **no-control-word** statement:

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

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

Configuring the Encapsulation Type for the Layer 2 Circuit Neighbor Interface

You can specify the Layer 2 circuit encapsulation type for the interface receiving traffic from a Layer 2 circuit neighbor. The encapsulation type is carried in the LDP-signaling messages exchanged between Layer 2 circuit neighbors when pseudowires are created. The encapsulation type you configure for each Layer 2 circuit neighbor varies depending on the type of networking equipment or the type of Layer 2 protocol you have deployed in your network. If you do not specify an encapsulation type for the Layer 2 circuit, the encapsulation of the CE device interface is used by default.

Specify the encapsulation type for the Layer 2 circuit neighbor interface by including the **encapsulation-type** statement:

```
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);
```

You can include this statement at the following hierarchy levels:

- **[edit protocols l2circuit neighbor *address* interface *interface-name*]**
- **[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*]**

Enabling the Layer 2 Circuit When the Encapsulation Does Not Match

You can configure the Junos OS to allow a Layer 2 circuit to be established even though the encapsulation configured on the CE device interface does not match the encapsulation configured on the Layer 2 circuit interface by including the **ignore-encapsulation-mismatch** statement. You can configure the **ignore-encapsulation-mismatch** statement for the connection to the remote connection by including the statement at the **[edit protocols l2circuit neighbor *address* interface *interface-name*]** hierarchy level or for the local connection by including this statement at the **[edit protocols l2circuit local-switching interface *interface-name*]** hierarchy level.

```
ignore-encapsulation-mismatch;
```

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

Configuring the Protect Interface

You can configure a protect interface for the logical interface linking a virtual circuit to its destination, whether the destination is remote or local. A protect interface provides a backup for the protected interface in case of failure. Network traffic uses the primary interface only so long as the primary interface functions. If the primary interface fails, traffic is switched to the protect interface. The protect interface is optional.

To configure the protect interface, include the **protect-interface** statement:

protect-interface *interface-name*;

NOTE: **The protect interface must be configured prior to configuring the no-revert statement.**

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

For an example of how to configure a protect interface for a Layer 2 circuit, see "Example: Configuring Layer 2 Circuit Protect Interfaces" on page 139.

Configuring the Protect Interface From Switching Over to the Primary Interface

Typically, when the primary interface goes down, the pseudowire starts using the protect interface. By default, when the primary interface comes back online, the interface is switched-over back from the protect interface to the primary interface. To prevent the switchover back to the primary interface, unless the protect interface goes down, include the **no-revert** statement. This prevents loss of traffic during the switchover.

NOTE: **If the protect interface fails, the interface is switched-over back to the primary interface, irrespective of whether or not the no-revert statement is included in the configuration.**

You can configure the **no-revert** statement at the **[edit protocols l2circuit neighbor *address* interface *interface-name*]** hierarchy level:

[edit protocols l2circuit neighbor *address* interface *interface-name*]
no-revert;

Configuring the Pseudowire Status TLV

The pseudowire status type length variable (TLV) is used to communicate the status of a pseudowire back and forth between two PE routers. For Layer 2 circuit configurations, you can configure the PE router to negotiate the pseudowire with its neighbor using the pseudowire status TLV. This same functionality is also available for LDP VPLS neighbor configurations. The pseudowire status TLV is configurable for each pseudowire connection and is disabled by default. The pseudowire status negotiation process assures that a PE router reverts back to the label withdraw method for pseudowire status if its remote PE router neighbor does not support the pseudowire status TLV.

Unlike the control word, a PE router's ability to support the pseudowire status TLV is communicated when the initial label mapping message is sent to its remote PE router. Once the PE router transmits its support for the pseudowire status TLV to its remote PE router, it includes the pseudowire status TLV in every label mapping message sent to the remote PE router. If you disable support for the pseudowire status TLV on the PE router, a label withdraw message is sent to the remote PE router and then a new label mapping message without the pseudowire status TLV follows.

To configure the pseudowire status TLV for the pseudowire to the neighbor PE router, include the **pseudowire-status-tlv** statement:

```
pseudowire-status-tlv;
```

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

Configuring Layer 2 Circuits over Both RSVP and LDP LSPs

You can configure two Layer 2 circuits between the same two routers, and have one Layer 2 circuit traverse an RSVP LSP and the other traverse an LDP LSP. To accomplish this, you need to configure two loopback addresses on the local router. You configure one of the loopback address for the Layer 2 circuit traversing the RSVP LSP. You configure the other loopback address to handle the Layer 2 circuit traversing the LDP LSP. For information about how to configure multiple loop back interfaces, see *Configuring Logical Units on the Loopback Interface for Routing Instances in Layer 3 VPNs*.

You also need to configure a packet switched network (PSN) tunnel endpoint for one of the Layer 2 circuits. It can be either the Layer 2 circuit traversing the RSVP LSP or the one traversing the LDP LSP. The PSN tunnel endpoint address is the destination address for the LSP on the remote router.

To configure the address for the PSN tunnel endpoint, include the **psn-tunnel-endpoint** statement:

```
psn-tunnel-endpoint address;
```

You can include this statement at the following hierarchy levels:

- **[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*]**
- **[edit protocols l2circuit neighbor *address* interface *interface-name*]**

By default, the PSN tunnel endpoint for a Layer 2 circuit is identical to the neighbor address, which is also the same as the LDP neighbor address.

The tunnel endpoints on the remote router do not need to be loopback addresses.

Example: PSN Tunnel Endpoint

The following example illustrates how you might configure a PSN tunnel endpoint:

```
[edit protocols l2circuit]
neighbor 10.255.0.6 {
  interface t1-0/2/2.0 {
    psn-tunnel-endpoint 20.20.20.20;
    virtual-circuit-id 1;
  }
  interface t1-0/2/1.0 {
    virtual-circuit-id 10;
```

```
    }
  }
```

The Layer 2 circuit configured for the **t1-0/2/2.0** interface resolves in the inet3 routing table to **20.20.20.20**. This could be either an RSVP route or a static route with an LSP next hop.

Configuring the Virtual Circuit ID

You configure a virtual circuit ID on each interface. Each virtual circuit ID uniquely identifies the Layer 2 circuit among all the Layer 2 circuits to a specific neighbor. The key to identifying a particular Layer 2 circuit on a PE router is the neighbor address and the virtual circuit ID. An LDP-FEC-to-label binding is associated with a Layer 2 circuit based on the virtual circuit ID in the FEC and the neighbor that sent this binding. The LDP-FEC-to-label binding enables the dissemination of the VPN label used for sending traffic on that Layer 2 circuit to the remote CE device.

You also configure a virtual circuit ID for each redundant pseudowire. A redundant pseudowire is identified by the backup neighbor address and the virtual circuit ID. For more information, see *Configuring Redundant Pseudowires for Layer 2 Circuits and VPLS*.

To configure the virtual circuit ID, include the **virtual-circuit-id** statement:

virtual-circuit-id *identifier*;

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

Configuring the Interface Encapsulation Type for Layer 2 Circuits

The Layer 2 encapsulation type is carried in the LDP forwarding equivalence class (FEC). You can configure either circuit cross-connect (CCC) or translational cross-connect (TCC) encapsulation types for Layer 2 circuits. For more information, see the *Junos OS MPLS Applications Library for Routing Devices* and *Junos OS Network Interfaces Library for Routing Devices*.

NOTE: Some platform and FPC combinations can not pass TCC encapsulated ISO traffic. See *Platforms/FPCs That Cannot Forward TCC Encapsulated ISO Traffic* for details.

To configure the interface encapsulation for a Layer 2 circuit, include the **encapsulation** statement:

encapsulation *encapsulation*;

You can include this statement at the following hierarchy levels:

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

Configuring ATM2 IQ Interfaces for Layer 2 Circuits

You can configure Asynchronous Transfer Mode 2 (ATM2) intelligent queuing (IQ) interfaces for Layer 2 circuits by using Layer 2 circuit ATM Adaptation Layer 5 (AAL5) transport mode, Layer 2 circuit ATM cell relay mode, and the Layer 2 circuit ATM trunk mode.

The configuration statements are as follows:

- **atm-l2circuit-mode aal5**
- **atm-l2circuit-mode cell**
- **atm-l2circuit-mode trunk**

For more information about these statements, see the *Junos OS Administration Library for Routing Devices*. For more information about how to configure ATM2 IQ interfaces, see the *Junos OS Network Interfaces Library for Routing Devices*.

The Junos OS implementation of sequence number processing for Layer 2 circuit ATM cell relay mode and Layer 2 circuit AAL5 mode differs from that described in the Internet draft draft-martini-l2circuit-encap-mpls-11.txt, *Encapsulation Methods for Transport of Layer 2 Frames over MPLS Networks* (expires August 2006).

The Junos OS implementation has the following differences:

1. A packet with a sequence number of 0 is treated as out of sequence.
2. A packet that does not have the next incremental sequence number is considered out of sequence.

When out-of-sequence packets arrive, the expected sequence number for the neighbor is set to the sequence number in the Layer 2 circuit control word.

Configuring the MTU for Layer 2 Interfaces

By default, the MTU used to advertise a Layer 2 pseudowire is determined by taking the interface MTU for the associated physical interface and subtracting the encapsulation overhead for sending IP packets based on the encapsulation. However, encapsulations that support multiple logical interfaces (and multiple Layer 2 pseudowires) rely on the same interface MTU (since they are all associated with the same physical interface). This can prove to be a limitation for VLAN Layer 2 pseudowires using the same Ethernet interface or for Layer 2 pseudowire DLCIs using the same Frame Relay interface.

This can also affect multivendor environments. For example, if you have three PE devices supplied by different vendors and one of the devices only supports an MTU of 1500, even if the other devices support larger MTUs you must configure the MTU as 1500 (the smallest MTU of the three PE devices).

You can explicitly configure which MTU is advertised for a Layer 2 pseudowire, even if the Layer 2 pseudowire is sharing a physical interface with other Layer pseudowires. When you explicitly configure an MTU for a Layer 2 pseudowire, be aware of the following:

- For BGP-based applications such as l2vpn and bgp-vpls, the advertised MTU will be zero unless an MTU value is explicitly set at the **[edit routing-instances *routing-instance-name* protocols (*l2vpn* | *vpls*) site *site-name*]** hierarchy level.
- An explicitly configured MTU is signaled to the remote PE device. The configured MTU is also compared to the MTU received from the remote PE device. If there is a conflict, the Layer 2 pseudowire is taken down.
- If you configure an MTU for an ATM cell relay interface on an ATM II PIC, the configured MTU is used to compute the cell bundle size advertised for that Layer 2 pseudowire, instead of the default interface MTU.
- A configured MTU is used only in the control plane. It is not enforced in the data plane. You need to ensure that the CE device for a given Layer 2 pseudowire uses the correct MTU for data transmission.

The following procedure describes how to configure the MTU for the Layer 2 interface. This information applies to the following Layer 2 technologies:

- Layer 2 VPNs
- Layer 2 Circuits
- VPLS

1. To configure the MTU for a Layer 2 circuit, include the **mtu** statement:

   mtu *mtu-number*;

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

2. To allow a Layer 2 pseudowire to be established even though the MTU configured on the local PE router does not match the MTU configured on the remote PE router, include the **ignore-mtu-mismatch** statement:

```
ignore-mtu-mismatch;
```

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

Related Documentation

- ignore-mtu-mismatch on page 265
- mtu on page 278

Configuring Static Layer 2 Circuits

You can configure static Layer 2 circuit pseudowires. Static pseudowires are designed for networks that do not support LDP or do not have LDP enabled. You configure a static pseudowire by configuring static values for the in and out labels needed to enable a pseudowire connection. The **ignore-mtu-mismatch**, **ignore-vlan-id**, and **ignore-encapsulation-mismatch** statements are not relevant for static pseudowire configurations since the peer router cannot forward this information.

When you configure static pseudowires, you need to manually compare the encapsulation, TDM bit rate, and control word of the router with the remote peer router and ensure that they match, otherwise the static pseudowire might not work.

To configure static Layer 2 circuit pseudowires, include the **static** statement:

```
static {
  incoming-label label;
  outgoing-label label;
  send-oam;
}
```

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

You can configure a static pseudowire as a standalone Layer 2 circuit or in conjunction with a redundant pseudowire. You configure the static pseudowire statement at the **[edit protocols l2circuit neighbor *address* interface *interface-name*]** hierarchy level. You configure the redundant pseudowire at the **[edit protocols l2circuit neighbor *address* interface *interface-name* backup-neighbor *neighbor*]** hierarchy level. If you configure a static pseudowire to a neighbor and also configure a redundant pseudowire, the redundant pseudowire must also be static.

You can enable the ability to ping a static pseudowire by configuring the **send-oam** statement. This functionality applies to the backup neighbor as well. Once you have configured this statement, you can ping the static pseudowire by issuing the **ping mpls l2circuit** command.

For information about how to configure redundant pseudowires, see *Configuring Redundant Pseudowires for Layer 2 Circuits and VPLS*.

Related Documentation

- *Configuring Redundant Pseudowires for Layer 2 Circuits and VPLS*
- ping mpls l2circuit on page 310

Configuring Policies for Layer 2 Circuits

You can configure Junos routing policies to control the flow of packets over Layer 2 circuits. This capability allows you to provide different level of service over a set of equal-cost Layer 2 circuits. For example, you can configure a circuit for high-priority traffic, a circuit for average-priority traffic, and a circuit for low-priority traffic. By configuring Layer 2 circuit policies, you can ensure that higher-value traffic has a greater likelihood of reaching its destination.

The following sections explain how to configure Layer 2 circuit policies:

Configuring the Layer 2 Circuit Community

To configure a community for Layer 2 circuits, include the **community** statement.

```
community community-name {
  members [ community-ids ];
}
```

You can include this statement at the following hierarchy levels:

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

name identifies the community or communities.

community-ids identifies the type of community or extended community:

- A normal community uses the following community ID format:

  as-number*:*community-value

  as-number is the autonomous system (AS) number of the community member.

  community-value is the identifier of the community member. It can be a number from 0 through 65,535.

- An extended community uses the following community ID format:

  type*:*administrator*:*assigned-number

  type is the type of target community. The target community identifies the route's destination.

  administrator is either an AS number or an IP version 4 (IPv4) address prefix, depending on the type of community.

  assigned-number identifies the local provider.

  You also need to configure the community for the Layer 2 circuit interface; see "Configuring a Community for the Layer 2 Circuit" on page 48.

Configuring the Policy Statement for the Layer 2 Circuit Community

To configure a policy to send community traffic over a specific LSP, include the **policy-statement** statement:

```
policy-statement policy-name {
  term term-name {
    from community community-name;
    then {
      install-nexthop (except | lsp lsp-name | lsp-regex lsp-regular-expression);
      accept;
    }
  }
}
```

You can include this statement at the following hierarchy levels:

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

To prevent the installation of any matching next hops, include the **install-nexthop** statement with the **except** option:

```
install-nexthop except;
```

You can include this statement at the following hierarchy levels:

- **[edit policy-options policy-statement *policy-name* term *term-name* then]**
- **[edit logical-systems *logical-system-name* policy-options policy-statement *policy-name* term *term-name* then]**

To assign traffic from a community to a specific LSP, include the **install-nexthop** statement with the **lsp *lsp-name*** option and the **accept** statement:

```
install-nexthop lsp lsp-name;
accept;
```

You can include these statements at the following hierarchy levels:

- **[edit policy-options policy-statement *policy-name* term *term-name* then]**
- **[edit logical-systems *logical-system-name* policy-options policy-statement *policy-name* term *term-name* then]**

You can also use a regular expression to select an LSP from a set of similarly named LSPs for the **install-nexthop** statement. To configure a regular expression, include the **install-nexthop** statement with the **lsp-regex** option and the **accept** statement:

```
install-nexthop lsp-regex lsp-regular-expression;
accept;
```

You can include these statements at the following hierarchy levels:

- **[edit policy-options policy-statement *policy-name* term *term-name* then]**

- **[edit logical-systems *logical-system-name* policy-options policy-statement *policy-name* term *term-name* then]**

Example: Configuring a Policy for a Layer 2 Circuit Community

The following example illustrates how you might configure a regular expression in a Layer 2 circuit policy. You create three LSPs to handle gold-tier traffic from a Layer 2 circuit. The LSPs are named **alpha-gold**, **beta-gold**, and **delta-gold**. You then include the **install-nexthop** statement with the **lsp-regex** option with the LSP regular expression **.*-gold** at the **[edit policy-options policy-statement *policy-name* term *term-name* then]** hierarchy level:

```
[edit policy-options]
policy-statement gold-traffic {
   term to-gold-LSPs {
      from community gold;
      then {
         install-nexthop lsp-regex .*-gold;
         accept;
      }
   }
}
```

The community **gold** Layer 2 circuits can now use any of the **-gold** LSPs. Given equal utilization across the three **-gold** LSPs, LSP selection is made at random.

You need to apply the policy to the forwarding table. To apply a policy to the forwarding table, configure the **export** statement at the **[edit routing-options forwarding-table]** hierarchy level:

```
[edit routing-options forwarding-table]
export policy-name;
```

Verifying the Layer 2 Circuit Policy Configuration

To verify that you have configured a policy for the Layer 2 circuit, issue the **show route table mpls detail** command. It should display the community for ingress routes that corresponds to the Layer 2 circuits, as shown by the following example:

```
user@host> show route table mpls detail
so-1/0/1.0 (1 entry, 1 announced)
*L2VPN  Preference: 7
Next hop: via so-1/0/0.0 weight 1, selected
Label-switched-path to-community-gold
Label operation: Push 100000 Offset: -4
Next hop: via so-1/0/0.0 weight 1
Label-switched-path to-community-silver
Label operation: Push 100000 Offset: -4
Protocol next hop: 10.255.245.45
Push 100000 Offset: -4
Indirect next hop: 85333f0 314
State: <Active Int>
Local AS:   100
Age: 22
Task: Common L2 VC
Announcement bits (2): 0-KRT 1-Common L2 VC
```

```
AS path: I
Communities: 100:1
```

For more information about how to configure routing policies, see the *Routing Policy Feature Guide for Routing Devices*.

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*). 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*. 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*
- l2vpn-use-bgp-rules on page 274
- route-distinguisher on page 294

Configuring ATM Trunking on Layer 2 Circuits

You can configure Layer 2 circuits to transport ATM traffic from directly connected ATM switches across an MPLS core network. Traffic from an ATM switch is received on the local PE router. The ATM cells are given an MPLS label and then sent across the MPLS network to the remote PE router. The receiving router removes the MPLS label from the ATM cell and then forwards the cell the receiving ATM switch.

NOTE: ATM trunking on Layer 2 circuits is supported only on T Series and M320 routers and ATM2 IQ PICs.

Figure 6: ATM Trunking on Layer 2 Circuits

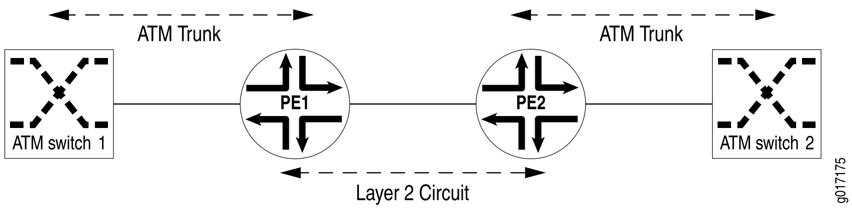


Figure 6 on page 62 illustrates how ATM switches could be linked together by a Layer 2 circuit. The PE1 Router is configured to receive ATM trunk traffic from ATM Switch 1. As each ATM cell is received on the PE1 Router, it is classified by means of the class-of-service (CoS) information in the cell header and then encapsulated as a labeled packet. The CoS information and cell loss priority (CLP) of the ATM cell are copied into the experimental (EXP) bits of the MPLS label. The labeled packet is then transported across the service provider network to the PE2 Router by means of a Layer 2 circuit.

On the PE2 Router, the label is removed and the plain ATM cell is forwarded to ATM Switch 2. The CoS and CLP are extracted from the EXP bits and are then used to select the correct output queue and determine whether the ATM cell should be dropped.

The ATM physical port on the router can support 32 logical trunks when network-to-network interface (NNI) is used and 8 logical trunks when user-to-network interface (UNI) is used. A trunk can carry traffic on 32 virtual path identifiers (VPIs), numbered 0 through 31. Each ATM trunk is associated with an MPLS label and a logical interface. On the ingress router, one or more of these trunks are mapped to a Layer 2 circuit.

The configuration for the Layer 2 circuit between PE routers is conventional. Follow the procedures outlined in this chapter for configuring the circuit. However, there is some specific configuration you need to complete for the Layer 2 circuit to carry traffic from an ATM trunk.

First, enable ATM trunking for Layer 2 circuits. To enable ATM trunking for Layer 2 circuits, specify the **trunk** option for the **atm-l2circuit-mode** statement at the **[edit chassis fpc *number* pic *number*]** hierarchy level:

[edit chassis fpc *number* pic *number*]


```
atm-l2circuit-mode trunk (uni | nni);
```

Specify the **uni** option for UNI trunks and the **nni** option for NNI trunks. The default option is **uni**.

You also need to configure each ATM trunk for a specific logical interface. Each ATM trunk has a trunk identifier in the range from 0 to 31. This configuration step is in addition to the typical configuration steps you follow related to configuring interfaces for Layer 2 circuits, as described in "Configuring Interfaces for Layer 2 Circuits" on page 47.

To associate a specific trunk identifier with a logical interface, include the **trunk-id** statement:

trunk-id *number*;

You can include this statement at the following hierarchy levels:

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

Since ATM trunking is supported on ATM2 IQ PICs only, the only value you can configure for the **pic-type** statement is **atm2**. If you do not configure the **pic-type** statement but you do configure the **trunk** option for the **atm-l2circuit-mode** statement (at the **[chassis fpc *number* pic *number*]** hierarchy level), the **pic-type** statement defaults to **atm2**.

Configuring Bandwidth Allocation and Call Admission Control in Layer 2 Circuits

You can configure bandwidth allocation and call admission control (CAC) on Layer 2 circuits. This feature is available for RSVP-signaled LSPs traversing an MPLS network.

When you enable bandwidth allocation on a Layer 2 circuit, attempts to establish an RSVP-signaled LSP are preceded by a check of the available bandwidth on the network. This check is the CAC. The available bandwidth is compared to the bandwidth requested by the LSP. If there is insufficient bandwidth, the Layer 2 circuit is not established and an error message is generated. To apply CAC to a Layer 2 circuit, a bandwidth constraint must be configured.

You can specify the bandwidth for a Layer 2 circuit without configuring a bandwidth for each class type (queue). To specify the bandwidth allocation for a Layer 2 circuit, include the **bandwidth** statement:

bandwidth *bandwidth*;

Specify the bandwidth in bits per second.

You can include this statement at the following hierarchy levels:

- **[edit protocols l2circuit neighbor *address* interface *interface-name*]**
- **[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*]**

Alternatively, you can configure the bandwidth for each class type on a Layer 2 circuit. If you use this type of configuration, you cannot simultaneously configure the nonclass type of bandwidth configuration for the Layer 2 circuit (the commit operation fails).

To configure the bandwidth for each class type on an Layer 2 circuit, include the **bandwidth** statement:

```
bandwidth {
  ct0 bandwidth;
  ct1 bandwidth;
  ct2 bandwidth;
  ct3 bandwidth;
}
```

You can include this statement at the following hierarchy levels:

- **[edit protocols l2circuit neighbor *address* interface *interface-name*]**
- **[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*]**

Specify the bandwidth for each class type in bits per second. It is not necessary to specify a bandwidth for all four class types.

Reducing APS Switchover Time in Layer 2 Circuits

On M320 routers with Channelized OC3/STM1 Circuit Emulation PIC with SFP, you can configure the **fast-aps-switch** statement at the **[edit interfaces *interface-name* sonet-options aps]** hierarchy level to reduce the Automatic Protection Switching (APS) switchover time in Layer 2 circuits. Configuring this statement reduces the APS switchover time only when the Layer 2 circuit encapsulation type for the interface receiving traffic from a Layer 2 circuit neighbor is Structure Agnostic time-division multiplexing (TDM) over Packet (SAToP).

The **fast-aps-switch** statement must be configured on both working and protect circuits. Additionally, to achieve reduction in APS switchover time:

- Per-packet load balancing must be configured.
- Bidirectional switching mode must be configured.
- If the **fast-aps-switch** statement is configured in revertive APS mode, configure an appropriate value for revert time. We recommend that you configure a revert time of 600 seconds for 672 through 1344 Layer 2 circuits.
- To prevent the logical interfaces in the data path from being shut down, configure appropriate hold-time values on all the interfaces in the data path that support TDM.

NOTE:

- The **fast-aps-switch** statement cannot be configured when the APS **annex-b** option is configured.
- The interfaces that have the **fast-aps-switch** statement configured cannot be used in virtual private LAN service (VPLS) environments.

The following tasks illustrate how to configure Junos OS to reduce APS switchover time.

NOTE: Per-packet load balancing can be configured for a limited set of routes or for all routes. To simplify the steps involved in configuring per-packet load balancing, steps for configuring per-packet load balancing for all routes is covered in this procedure.

- Configuring Per-Packet Load Balancing on page 65
- Configuring Fast APS Switchover on page 65

Configuring Per-Packet Load Balancing

To configure per-packet load balancing for all routes:

1. Configure the **per-packet** option for the **load-balance** statement at the **[edit policy-options policy-statement *policy-name* then]** hierarchy level.

   [edit policy-options policy-statement *policy-name* then]
   user@host# **set load-balance per-packet**

   For example:

   [edit policy-options policy-statement load-balancing-policy then]
   user@host# **set load-balance per-packet**

2. Configure the policy name in the **export** statement at the **[edit routing-options forwarding-table]** hierarchy level.

   [edit routing-options forwarding-table]
   user@host# **set export *policy-name***

   For example:

   [edit routing-options forwarding-table]
   user@host# **set export load-balancing-policy**

Configuring Fast APS Switchover

To configure fast APS switchover:

1. On both the working and protect circuits, configure the **fast-aps-switch** statement at the **[edit interfaces *interface-name* sonet-options aps]** hierarchy level.

   [edit interfaces *interface-name* sonet-options aps]
   user@host# **set fast-aps-switch**

For example:

[edit interfaces cstm1-0/0/0 sonet-options aps]
user@host# **set fast-aps-switch**

[edit interfaces cstm1-0/1/0 sonet-options aps]
user@host# **set fast-aps-switch**

2. Configure bidirectional switching mode on both the working and protect circuits. To do this, configure the **switching-mode bidirectional** statement at the **[edit interfaces *interface-name* sonet-options aps]** hierarchy level on both the working and protect circuits.

   [edit interfaces *interface-name* sonet-options aps]
   user@host# **set switching-mode bidirectional**

   For example:

   [edit interfaces cstm1-0/1/0 sonet-options aps]
   user@host# **set switching-mode bidirectional**

   [edit interfaces cstm1-0/1/0 sonet-options aps]
   user@host# **set switching-mode bidirectional**

3. If APS is configured in revertive mode, configure an appropriate value for revert time on both the working and protect circuits. To do this, configure the **revert-time** statement at the **[edit interfaces *interface-name* sonet-options aps]** hierarchy level on both the working and protect circuits.

   [edit interfaces *interface-name* sonet-options aps]
   user@host# **set revert-time *seconds***

   For example:

   [edit interfaces cstm1-0/0/0 sonet-options aps]
   user@host# **set revert-time 600**

   [edit interfaces cstm1-0/1/0 sonet-options aps]
   user@host# **set revert-time 600**

4. To prevent the logical interfaces in the data path from being shut down, configure appropriate hold-time values on *all* interfaces in the data path that support TDM.

   [edit interfaces *interface-name* hold-time]
   user@host# **set up *seconds* down *seconds***

   For example:

   [edit interfaces cstm1-0/0/0 hold-time]
   user@host# **set up 1 down 400**

Configuring the FAT Flow Label for FEC 128 VPWS Pseudowires for Load-Balancing MPLS Traffic

This topic shows how to configure flow-aware transport of pseudowires (FAT) flow labels for forwarding equivalence class (FEC) 128 virtual private wire service (VPWS) pseudowires for load-balancing MPLS traffic.

FAT flow labels enable load-balancing of MPLS packets across equal-cost multipath (ECMP) paths or link aggregation groups (LAGs) without the need for deep packet inspection of the payload. FAT flow labels can be used for LDP-signaled FEC 128 and FEC 129 pseudowires for virtual private LAN service (VPLS) and VPWS networks.

You can configure FAT flow labels to be signaled by LDP on FEC 128 VPWS pseudowires by including the **flow-label-transmit** and **flow-label-receive** configuration statements at the **[edit protocols l2circuit neighbor *neighbor-id* interface *interface-name*]** hierarchy level. This configuration sets the T bit and R bit advertisement to 1 (the default being 0) in the Sub-TLV field, which is one of the interface parameters of the FEC for the LDP label-mapping message header. These statements signal the pushing and popping of the load-balancing label to the routing peers in the control plane.

Alternatively, you can configure the following statements at the **[edit protocols l2circuit neighbor *neighbor-id* interface *interface-name*]** hierarchy level:

- **flow-label-transmit-static** to statically push the flow label on the pseudowire packets sent to the remote provider edge (PE) router.
- **flow-label-receive-static** to statically pop the flow label on the pseudowire packets received from the remote PE router.

Before you begin:

1. Configure the device interfaces and enable MPLS on all the interfaces.
2. Configure MPLS and an LSP to the remote PE router.
3. Configure OSPF and IS-IS.
4. Configure LDP on the loopback interface and the PE interface connecting to the P (transit) router.

To configure the FAT flow label for an FEC 128-signaled VPLS pseudowire, on the ingress PE router:

1. Configure the neighbors for the Layer 2 circuit.

   [edit protocols l2circuit]
   user@PE1# **set neighbor *neighbor-id***

   All the Layer 2 circuits using a particular remote PE router designated for remote CE routers are listed under the **neighbor** statement. Each neighbor is identified by its IP address and is usually the end-point destination for the label-switched path (LSP) tunnel transporting the Layer 2 circuit.

2. Configure the interface for the Layer 2 circuit neighbor and a unique identifier for the Layer 2 circuit.

   [edit protocols l2circuit neighbor *neighbor-id*]
   user@PE1# **set interface** ***interface-name*** **virtual-circuit-id** ***unique-l2ckt-identifier***

3. NOTE: You can only configure one of the following pairs of statements:

   - **flow-label-transmit** and **flow-label-receive** or,
   - **flow-label-transmit-static** and **flow-label-receive-static**

   Configure the router to signal the capability to push the flow label in the transmit direction to the remote PE router.

   [edit protocols l2circuit neighbor *neighbor-id* interface *interface-name*]
   user@PE1# **set flow-label-transmit**

4. Alternatively, configure the **flow-label-transmit-static** statement to statically push the flow label on the pseudowire packets sent to the remote PE router.

   [edit protocols l2circuit neighbor *neighbor-id* interface *interface-name*]
   user@PE1# **set flow-label-transmit-static**

   If the incoming pseudowire packet is not marked with the flow label, the packet is dropped by the egress PE router.

5. Configure the router to signal the capability to pop the flow label in the receive direction to the remote egress PE router.

   [edit protocols l2circuit neighbor *neighbor-id* interface *interface-name*]
   user@PE1# **set flow-label-receive**

6. Alternatively, configure the **flow-label-receive-static** statement to pop the flow label on the pseudowire packets received from the remote PE router.

   [edit protocols l2circuit neighbor *neighbor-id* interface *interface-name*]
   user@PE1# **set flow-label-receive-static**

   The ingress PE router inserts the flow label in the pseudowire packet, irrespective of the information exchanged in the signaling plane. If the egress PE router cannot handle the pseudowire packet marked with the flow label, the packet is dropped.

7. Verify and commit the configuration.

   For example:

   ```
   [edit protocols l2circuit]
   user@PE1# show
   neighbor 10.255.104.135 {
     interface ge-1/0/8.0 {
       virtual-circuit-id 1;
       flow-label-transmit;
       flow-label-receive;
     }
   }
   ```

   OR:


```
[edit protocols l2circuit]
user@PE1# show
neighbor 10.255.104.135 {
  interface ge-1/0/8.0 {
    virtual-circuit-id 1;
    flow-label-transmit-static;
    flow-label-receive-static;
  }
}
```

8. Repeat the configuration on the remote egress PE router.

Related Documentation

- *Configuring the FAT Flow Label for FEC 128 VPLS Pseudowires for Load-Balancing MPLS Traffic*
- Configuring the FAT Flow Label for FEC 129 VPWS Pseudowires for Load-Balancing MPLS Traffic on page 69
- *Configuring the FAT Flow Label for FEC 129 VPLS Pseudowires for Load-Balancing MPLS Traffic*
- FAT Flow Labels Overview on page 18

Configuring the FAT Flow Label for FEC 129 VPWS Pseudowires for Load-Balancing MPLS Traffic

This topic shows how to configure flow-aware transport of pseudowires (FAT) flow labels for forwarding equivalence class (FEC) 129 virtual private wire service (VPWS) pseudowires.

FAT flow labels enable load-balancing of MPLS packets across equal-cost multipath (ECMP) paths or link aggregation groups (LAGs) without the need for deep packet inspection of the payload. FAT flow labels can be used for LDP-signaled FEC 128 and FEC 129 pseudowires for virtual private LAN service (VPLS) and VPWS networks.

You can configure FAT flow labels to be signaled by LDP on FEC 129 VPWS pseudowires (Layer 2 circuits) by including the **flow-label-transmit** and **flow-label-receive** configuration statements at the **[edit routing-instances *instance-name* protocols l2vpn site *name*]** or the **[edit routing-instances *instance-name* protocols l2vpn site *name* interface *interface-name*]** hierarchy level. This configuration sets the T bit and R bit advertisement to 1 (the default being 0) in the Sub-TLV field, which is one of the interface parameters of the FEC for the LDP label-mapping message header. These statements signal the pushing and popping of the load-balancing label to the routing peers in the control plane.

Before you begin:

1. Configure the device interfaces and enable MPLS on all core-facing interfaces.
2. Configure CCC encapsulation and the CCC address family for interfaces configured as members of the FEC 129 VPWS instance.
3. Configure MPLS and an LSP to the remote provider edge (PE) router.

4. Configure the BGP sessions on the PE devices with the BGP autodiscovery-only address family to allow exchange of the autodiscovery routes.
5. Configure an IGP such as IS-IS or OSPF.
6. Configure LDP on the loopback interface and the core-facing interface.
7. Configure the autonomous system (AS) number.

To configure the FAT flow label for an FEC 129 VPWS pseudowire, on the ingress PE router:

1. Configure the VPWS routing instance.

   LDP listens for routes from instance.l2vpn.0 for any instance configured for FEC 129 VPWS. These routes are identified by the **instance-type l2vpn** statement in the routing instance and the presence of the **l2vpn-id** statement.

   [edit]
   user@PE1# **set routing-instances** ***instance-name*** **instance-type l2vpn**
   user@PE1# **set routing-instances** ***instance-name*** **interface** ***interface-name***
   user@PE1# **set routing-instances** ***instance-name*** **route-distinguisher (*as-number:x:y* | *ip-address:id*)**
   user@PE1# **set routing-instances** ***instance-name*** **l2vpn-id *(as-number:x:y* | *ip-address:id)***
   user@PE1# **set routing-instances** ***instance-name*** **vrf-target** ***community***
   user@PE1# **set routing-instances** ***instance-name*** **protocols l2vpn site** ***site-name*** **source-attachment-identifier** ***identifier***
   user@PE1# **set routing-instances** ***instance-name*** **protocols l2vpn site** ***site-name*** **interface** ***interface-name*** **target-attachment-identifier** ***identifier***

   Because VPWS is a point-to-point service, FEC 129 VPWS routing instances are configured as **instance-type l2vpn**. As with FEC 129 VPLS, FEC 129 VPWS uses the **l2vpn-id** statement to define the Layer 2 VPN of which the routing instance is a member. The presence of the **l2vpn-id** statement designates that FEC 129 LDP-signaling is used for the routing instance.

2. Configure the device to signal the capability to push the flow label in the transmit direction to the remote PE router.

   [edit routing-instances *instance-name* protocols l2vpn site *name* interface *interface-name*]
   user@PE1# **set flow-label-transmit**

3. Configure the device to signal the capability to pop the flow label in the receive direction to the remote PE router.

   [edit routing-instances *instance-name* protocols l2vpn site *name* interface *interface-name*]
   user@PE1# **set flow-label-receive**

4. Alternatively, configure the **flow-label-transmit** and **flow-label-receive** statements directly within the site. When configured within the site, the defined parameters affect any pseudowire originating from that site. When configured under an interface within the site, the defined parameters affect that single specific pseudowire. This enables you to manipulate the parameters across all pseudowires associated with a particular local site in one place in the configuration.

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

```
user@PE1# set flow-label-receive
user@PE1# set flow-label-transmit
```

5. Verify and commit the configuration.

   For example:

```
[edit routing-instances FEC129-VPWS]
user@PE1# show
instance-type l2vpn;
interface ge-0/0/1.600;
route-distinguisher 10.255.255.1:100;
l2vpn-id l2vpn-id:100:100;
vrf-target target:100:100;
protocols {
  l2vpn {
    site ONE {
      source-attachment-identifier 1;
      interface ge-0/0/1.600 {
        target-attachment-identifier 2;
        flow-label-transmit; <<< Applicable only to the pseudowire specific to the
          interface >>>
        flow-label-receive;
      }
    }
    site TWO {
      source-attachment-identifier 3;
      flow-label-transmit; <<< Applicable to all pseudowires within the site >>>
      flow-label-receive;
      interface ge-0/0/2.600 {
        target-attachment-identifier 1;
      }
      interface ge-0/0/2.601 {
        target-attachment-identifier 4;
      }
    }
  }
}
```

6. Repeat the configuration on the remote egress PE router.

Related Documentation

- Configuring the FAT Flow Label for FEC 128 VPWS Pseudowires for Load-Balancing MPLS Traffic on page 67
- *Configuring the FAT Flow Label for FEC 128 VPLS Pseudowires for Load-Balancing MPLS Traffic*
- *Configuring the FAT Flow Label for FEC 129 VPLS Pseudowires for Load-Balancing MPLS Traffic*
- FAT Flow Labels Overview on page 18

CHAPTER 4

VPWS Examples

Example: Configuring FEC 129 BGP Autodiscovery for VPWS

This example shows how to configure the virtual private wire service (VPWS), where remote provider edge (PE) devices are automatically discovered dynamically by BGP, and pseudowires are signaled by LDP using FEC 129. This arrangement reduces the configuration burden that is associated with statically configured Layer 2 circuits while still using LDP as the underlying signaling protocol.

Requirements

This example requires Junos OS Release 13.2 or later on the PE devices.

Overview

Because VPWS is a point-to-point service, FEC 129 VPWS routing instances are configured as **instance-type l2vpn**. As with FEC 129 VPLS, FEC 129 VPWS uses the **l2vpn-id** statement to define the Layer 2 VPN of which the routing instance is a member. The presence of the **l2vpn-id** statement designates that FEC 129 LDP signaling is used for the routing instance. The absence of **l2vpn-id** indicates that BGP signaling is used instead.

The point-to-point nature of VPWS requires that you specify the source access individual identifier (SAII) and the target access individual identifier (TAII). This SAII-TAII pair defines a unique pseudowire between two PE devices.

The SAII is specified with the **source-attachment-identifier** statement within the FEC 129 VPWS routing instance. You configure the source attachment identifier and the interfaces to associate with that source attachment identifier. Under each interface, you can configure the TAII with the **target-attachment-identifier** statement. If the configured target identifier matches a source identifier advertised by a remote PE device by way of a BGP autodiscovery message, the pseudowire between that source-target pair is signaled. If there is no match between an advertised source identifier and the configured target identifier, the pseudowire is not established.

Sample: VPWS Configuration with Multiple Interfaces and Sites

```
routing-instances {
  FEC129-VPWS {
    instance-type l2vpn;
    interface ge-0/0/1.0;
    interface ge-0/0/2.0;
    interface ge-0/0/3.0;
    route-distinguisher 10.255.0.1:200;
    l2vpn-id l2vpn-id:100:200;
    vrf-target target:100:200;
    protocols l2vpn {
      site CUSTOMER-1 {
        source-attachment-identifier 1;
        interface ge-0/0/1.0 {
          target-attachment-identifier 2;
        }
        interface ge-0/0/2.0 {
          target-attachment-identifier 3;
        }
      }
    }
  }
}
```

You can configure multiple interfaces within a site, because each SAII-TAII pair defines a unique pseudowire, as shown with pseudowires 1-2 and 1-3 in the sample configuration. Both the source and target access identifiers are 4-byte numbers and can only be configured in FEC 129 VPWS instances where the **instance-type** is **l2vpn** and the **l2vpn-id** configuration statement is present.

You can specify the source and target identifiers as plain unsigned integers in the range 1 through 4,292,967,295.

The Layer 2 circuit and Layer 2 VPN services allow many optional parameters to be included on a per-pseudowire basis. FEC 129 VPWS allows such parameters as MTU settings, community tagging, and inclusion of a control word, as shown in this sample configuration:

Sample: VPWS Configuration with Optional Configuration Parameters

```
routing-instances {
  FEC129-VPWS {
    instance-type l2vpn;
    interface ge-0/0/1.0;
    interface ge-0/0/2.0;
    interface ge-0/0/3.0;
    route-distinguisher 10.255.0.1:200;
    l2vpn-id l2vpn-id:100:200;
    vrf-target target:100:200;
    protocols l2vpn {
      site CUSTOMER-1 {
        source-attachment-identifier 1;
        community COMM;
        control-word ;
```



```
                encapsulation-type ethernet;
                ignore-encapsulation-mismatch;
                ignore-mtu-mismatch;
                mtu 1500;
                no-control-word;
                interface ge-0/0/1.0 {
                    target-attachment-identifier 2;
                }
                interface ge-0/0/2.0 {
                    target-attachment-identifier 3;
                    community COMM;
                    control-word;
                    encapsulation-type ethernet;
                    ignore-encapsulation-mismatch;
                    ignore-mtu-mismatch;
                    mtu 1500;
                    no-control-word;
                }
            }
        }
    }
}
```

When configured within the site, the defined parameters affect any pseudowire originating from that site. When configured under an interface, the defined parameters affect that single specific pseudowire. This allows you to manipulate the parameters across all pseudowires associated with a particular local site in one place in the configuration.

Like other point-to-point services, the interfaces configured as members of the FEC 129 VPWS instance must be configured for CCC encapsulation and the CCC address family, as shown here:

```
interfaces {
    ge-0/0/1 {
        encapsulation ethernet-ccc;
        unit 0 {
            family ccc;
        }
    }
    ge-0/0/2 {
        encapsulation ethernet-ccc;
        unit 0 {
            family ccc;
        }
    }
    ge-0/0/3 {
        encapsulation ethernet-ccc;
        unit 0 {
            family ccc;
        }
    }
}
```

You can use **vlan-ccc** instead of **ethernet-ccc**.

To support the basic FEC 129 VPWS functionality, the BGP sessions on the PE devices also need to be configured with the BGP **auto-discovery-only** address family to allow exchange of the autodiscovery routes. If traditional BGP VPLS or Layer 2 VPN service is also provisioned on the PE devices, the address family **l2vpn signaling** is also required, as shown here:

```
bgp {
   group pe {
      type internal;
      local-address 10.255.0.1;
      family l2vpn {
         auto-discovery-only;
         signaling;
      }
      neighbor 10.255.0.2;
      neighbor 10.255.0.3;
   }
}
```

The following configuration sample shows an FEC 129 VPWS routing instance with the operation, administration, and maintenance (OAM) (ping and BFD) configuration options:

Sample: VPWS Configuration with OAM

```
routing-instances {
   FEC129-VPWS {
      instance-type l2vpn;
      interface ge-0/0/1.0;
      route-distinguisher 10.255.0.1:200;
      l2vpn-id l2vpn-id:100:200;
      vrf-target target:100:200;
      protocols l2vpn {
         oam {
            ping-interval 600;
            bfd-liveness-detection {
               minimum-interval 200;
            }
         }
         site CUSTOMER {
            source-attachment-identifier 1;
            oam {
               ping-interval 600;
               bfd-liveness-detection {
                  minimum-interval 200;
               }
            }
            interface ge-0/0/1.0 {
               oam {
                  ping-interval 600;
                  bfd-liveness-detection {
                     minimum-interval 200;
                  }
               }
               target-attachment-identifier 2;
            }
         }
      }
   }
```



```
}
```

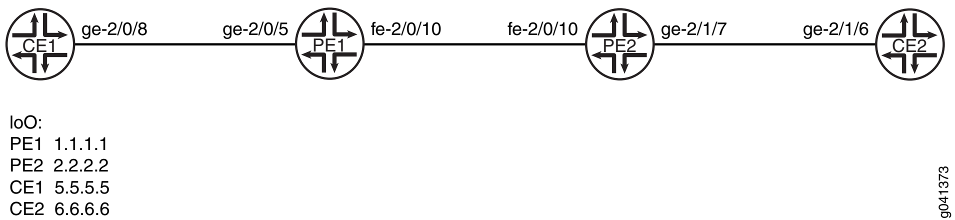
OAM options configured under **protocols l2vpn** apply to all sites and pseudowires in the routing instance. OAM options configured under a particular site apply to the pseudowires configured under that site. OAM options configured under a particular interface apply to the pseudowire configured under that interface.

Topology Diagram

Figure 7 on page 77 shows the topology used in this example.

This example uses a simple topology with two PE devices and two customer edge (CE) devices.

Figure 7: Simple VPWS Topology



“CLI Quick Configuration” on page 77 shows the configuration for all of the devices in Figure 7 on page 77. The section “Step-by-Step Procedure” on page 78 describes the steps on Device PE1.

Configuration

CLI Quick Configuration

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

Device CE1

```
set interfaces ge-2/0/8 unit 0 description CE1_to_PE1
set interfaces ge-2/0/8 unit 0 family inet address 172.16.0.1/24
set interfaces lo0 unit 0 family inet address 5.5.5.5/32
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface ge-2/0/8.0
```

Device CE2

```
set interfaces ge-2/1/6 unit 0 description CE2_to_PE2
set interfaces ge-2/1/6 unit 0 family inet address 172.16.0.4/24
set interfaces lo0 unit 0 family inet address 6.6.6.6/32
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface ge-2/1/6.0
```

Device PE1

```
set interfaces ge-2/0/5 encapsulation ethernet-ccc
set interfaces ge-2/0/5 unit 0 description PE1_to_CE1
set interfaces ge-2/0/5 unit 0 family ccc
set interfaces fe-2/0/10 unit 0 description to_PE2
set interfaces fe-2/0/10 unit 0 family inet address 10.0.0.1/30
set interfaces fe-2/0/10 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 1.1.1.1/32
set protocols mpls interface fe-2/0/10.0
```

```
set protocols bgp local-address 1.1.1.1
set protocols bgp group pe-pe type internal
set protocols bgp group pe-pe family l2vpn auto-discovery-only
set protocols bgp group pe-pe family l2vpn signaling
set protocols bgp group pe-pe neighbor 2.2.2.2
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface fe-2/0/10.0
set protocols ldp interface fe-2/0/10.0
set protocols ldp interface lo0.0
set routing-instances FEC129-VPWS instance-type l2vpn
set routing-instances FEC129-VPWS interface ge-2/0/5.0
set routing-instances FEC129-VPWS route-distinguisher 1.1.1.1:100
set routing-instances FEC129-VPWS l2vpn-id l2vpn-id:100:100
set routing-instances FEC129-VPWS vrf-target target:100:100
set routing-instances FEC129-VPWS protocols l2vpn site ONE
  source-attachment-identifier 1
set routing-instances FEC129-VPWS protocols l2vpn site ONE interface ge-2/0/5.0
  target-attachment-identifier 2
set routing-options autonomous-system 64510
```

Device PE2

```
set interfaces ge-2/1/7 encapsulation ethernet-ccc
set interfaces ge-2/1/7 unit 0 description PE2_to_CE2
set interfaces ge-2/1/7 unit 0 family ccc
set interfaces fe-2/0/10 unit 0 description to_PE1
set interfaces fe-2/0/10 unit 0 family inet address 10.0.0.2/30
set interfaces fe-2/0/10 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 2.2.2.2/32
set protocols mpls interface fe-2/0/10.0
set protocols bgp local-address 2.2.2.2
set protocols bgp group pe-pe type internal
set protocols bgp group pe-pe family l2vpn auto-discovery-only
set protocols bgp group pe-pe family l2vpn signaling
set protocols bgp group pe-pe neighbor 1.1.1.1
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface fe-2/0/10.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ldp interface fe-2/0/10.0
set protocols ldp interface lo0.0
set routing-instances FEC129-VPWS instance-type l2vpn
set routing-instances FEC129-VPWS interface ge-2/1/7.0
set routing-instances FEC129-VPWS route-distinguisher 2.2.2.2:100
set routing-instances FEC129-VPWS l2vpn-id l2vpn-id:100:100
set routing-instances FEC129-VPWS vrf-target target:100:100
set routing-instances FEC129-VPWS protocols l2vpn site TWO
  source-attachment-identifier 2
set routing-instances FEC129-VPWS protocols l2vpn site TWO interface ge-2/1/7.0
  target-attachment-identifier 1
set routing-options autonomous-system 64510
```

Step-by-Step Procedure

To configure a FEC 129 VPWS:

1. Configure the interfaces.

```
[edit interfaces]
user@PE1# set ge-2/0/5 encapsulation ethernet-ccc
```



```
user@PE1# set ge-2/0/5 unit 0 description PE1_to_CE1
user@PE1# set ge-2/0/5 unit 0 family ccc

user@PE1# set fe-2/0/10 unit 0 description to_PE2
user@PE1# set fe-2/0/10 unit 0 family inet address 10.0.0.1/30
user@PE1# set fe-2/0/10 unit 0 family mpls

user@PE1# set lo0 unit 0 family inet address 1.1.1.1/32
```

2. Configure MPLS on the core-facing interface.

```
[edit protocols mpls]
user@PE1# set interface fe-2/0/10.0
```

3. Configure BGP.

```
[edit protocols bgp]
user@PE1# set local-address 1.1.1.1
user@PE1# set group pe-pe type internal
user@PE1# set group pe-pe family l2vpn auto-discovery-only
user@PE1# set group pe-pe family l2vpn signaling
user@PE1# set group pe-pe neighbor 2.2.2.2
```

4. Configure an interior gateway protocol, such as IS-IS or OSPF.

   If you use OSPF, enable traffic engineering. Traffic engineering is supported by IS-IS by default.

```
[edit protocols ospf]
user@PE1# set traffic-engineering
user@PE1# set area 0.0.0.0 interface lo0.0 passive
user@PE1# set area 0.0.0.0 interface fe-2/0/10.0
```

5. Configure LDP on the core-facing interface and on the loopback interface.

```
[edit protocols ldp]
user@PE1# set interface fe-2/0/10.0
user@PE1# set interface lo0.0
```

6. Configure the VPWS routing instance.

   LDP listens for routes from instance.l2vpn.0 for any instance configured for FEC 129 VPWS. These routes are identified by the **instance-type l2vpn** statement in the routing instance and the presence of the **l2vpn-id** statement.

   Make sure that the **target-attachment-identifier** matches the **source-attachment-identifier** in the remote PE device's corresponding site. In this example, the pseudowire is established between Device PE1 and Device PE2. Device PE1 uses SAI 1 and TAI 2, while Device PE2 uses the opposite, SAI 2 and TAI 1.

```
[edit routing-instances FEC129-VPWS]
user@PE1# set instance-type l2vpn
user@PE1# set interface ge-2/0/5.0
user@PE1# set route-distinguisher 1.1.1.1:100
user@PE1# set l2vpn-id l2vpn-id:100:100
user@PE1# set vrf-target target:100:100
user@PE1# set protocols l2vpn site ONE source-attachment-identifier 1
```

```
user@PE1# set protocols l2vpn site ONE interface ge-2/0/5.0
  target-attachment-identifier 2
```

7. Configure the autonomous system (AS) number.

```
[edit routing-options]
user@PE1# set autonomous-system 64510
```

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

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

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

```
user@PE1# show interfaces
ge-2/0/5 {
  encapsulation ethernet-ccc;
  unit 0 {
    description PE1_to_CE1;
    family ccc;
  }
}
fe-2/0/10 {
  unit 1 {
    description to_PE2;
    family inet {
      address 10.0.0.1/30;
    }
    family mpls;
  }
}
lo0 {
  unit 0 {
    family inet {
      address 1.1.1.1/32;
    }
  }
}
```

```
user@PE1# show protocols
mpls {
  interface fe-2/0/10.0;
}
bgp {
  local-address 1.1.1.1;
  group pe-pe {
    type internal;
    family l2vpn {
      auto-discovery-only;
      inactive: signaling;
    }
    neighbor 2.2.2.2;
```



```
    }
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface lo0.0 {
        passive;
      }
      interface fe-2/0/10.0;
    }
  }
  ldp {
    interface fe-2/0/10.0;
    interface lo0.0;
  }
```

```
user@PE1# show routing-instances
FEC129-VPWS {
  instance-type l2vpn;
  interface ge-2/0/5.0;
  route-distinguisher 1.1.1.1:100;
  l2vpn-id l2vpn-id:100:100;
  vrf-target target:100:100;
  protocols {
    l2vpn {
      site ONE {
        source-attachment-identifier 1;
        interface ge-2/0/5.0 {
          target-attachment-identifier 2;
        }
      }
    }
  }
}
```

```
user@PE1# show routing-options
autonomous-system 64510;
```

Verification

Confirm that the configuration is working properly.

- Verifying the Routes on page 81
- Checking Connectivity Between the CE Devices on page 83
- Checking the VPWS Connections on page 83
- Checking Connectivity Between the PE Devices on page 84

Verifying the Routes

Purpose Verify that the expected routes are learned.

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

```
user@PE1> show route
```

```
inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1.1.1.1/32         *[Direct/0] 6d 21:16:32
                    > via lo0.0
2.2.2.2/32         *[OSPF/10] 6d 21:15:31, metric 1
                    > to 10.0.0.2 via fe-2/0/10.0
10.0.0.0/30        *[Direct/0] 6d 21:16:31
                    > via fe-2/0/10.0
10.0.0.1/32        *[Local/0] 6d 21:16:32
                      Local via fe-2/0/10.0
224.0.0.5/32       *[OSPF/10] 6d 21:16:34, metric 1
                      MultiRecv

inet.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2.2.2.2/32         *[LDP/9] 5d 22:25:19, metric 1
                    > to 10.0.0.2 via fe-2/0/10.0

mpls.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 6d 21:16:33, metric 1
                      Receive
1                  *[MPLS/0] 6d 21:16:33, metric 1
                      Receive
2                  *[MPLS/0] 6d 21:16:33, metric 1
                      Receive
13                 *[MPLS/0] 6d 21:16:33, metric 1
                      Receive
299808             *[LDP/9] 5d 22:25:19, metric 1
                    > to 10.0.0.2 via fe-2/0/10.0, Pop
299808(S=0)        *[LDP/9] 5d 22:25:19, metric 1
                    > to 10.0.0.2 via fe-2/0/10.0, Pop
299824             *[L2VPN/7] 5d 22:25:18
                    > via ge-2/0/5.0, Pop
ge-2/0/5.0         *[L2VPN/7] 5d 22:13:02, metric2 1
                    > to 10.0.0.2 via fe-2/0/10.0, Push 299872

bgp.l2vpn.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2.2.2.2:100:0.0.0.2/96 AD
                   *[BGP/170] 6d 20:51:23, localpref 100, from 2.2.2.2
                      AS path: I, validation-state: unverified
                    > to 10.0.0.2 via fe-2/0/10.0

ldp.l2vpn.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2.2.2.2:NoCtrlWord:5:100:100:0.0.0.2:0.0.0.1/176
                   *[LDP/9] 5d 22:13:02
                      Discard

FEC129-VPWS.l2vpn.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1.1.1.1:100:0.0.0.1/96 AD
                   *[L2VPN/170] 6d 20:53:26, metric2 1
                      Indirect
```



```
2.2.2.2:100:0.0.0.2/96 AD
                   *[BGP/170] 6d 20:51:23, localpref 100, from 2.2.2.2
                      AS path: I, validation-state: unverified
                    > to 10.0.0.2 via fe-2/0/10.0
2.2.2.2:NoCtrlWord:5:100:100:0.0.0.1:0.0.0.2/176
                   *[L2VPN/7] 6d 20:51:23, metric2 1
                    > to 10.0.0.2 via fe-2/0/10.0
2.2.2.2:NoCtrlWord:5:100:100:0.0.0.2:0.0.0.1/176
                   *[LDP/9] 5d 22:13:02
                      Discard
```

Meaning The output shows all the learned routes, including the autodiscovery (AD) routes.

Checking Connectivity Between the CE Devices

Purpose Verify that Device CE1 can ping Device CE2.

Action

```
user@CE1> ping 6.6.6.6
PING 6.6.6.6 (6.6.6.6): 56 data bytes
64 bytes from 6.6.6.6: icmp_seq=0 ttl=64 time=0.679 ms
64 bytes from 6.6.6.6: icmp_seq=1 ttl=64 time=0.524 ms
^C
--- 6.6.6.6 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.524/0.602/0.679/0.078 ms
```

Meaning The output shows that the VPWS is operational.

Checking the VPWS Connections

Purpose Make sure that all of the FEC 129 VPWS connections come up correctly.

Action

```
user@PE1> show l2vpn connections
Layer-2 VPN connections:

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

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

Instance: FEC129-VPWS
  L2vpn-id: 100:100
  Local source-attachment-id: 1 (ONE)
    Target-attachment-id      Type  St     Time last up          # Up trans
    2                         rmt   Up     Nov 28 16:16:14 2012           1
      Remote PE: 2.2.2.2, Negotiated control-word: No
      Incoming label: 299792, Outgoing label: 299792
      Local interface: ge-2/0/5.0, Status: Up, Encapsulation: ETHERNET
```

Meaning

As expected, the connection is up. The output includes the source attachment ID and the target attachment ID.

Checking Connectivity Between the PE Devices

Purpose

Verify that Device PE1 can ping Device PE2. The **ping mpls l2vpn fec129** command accepts SAIs and TAIs as integers or IP addresses and also allows you to use the CE-facing interface instead of the other parameters (**instance**, **local-id**, **remote-id**, **remote-pe-address**).

Action

```
user@PE1> ping mpls l2vpn fec129 instance FEC129-VPWS remote-id 2 remote-pe-address 2.2.2.2
local-id 1
!!!!!
--- lsping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss

user@PE1> ping mpls l2vpn fec129 interface ge-2/0/5.0
!!!!!
--- lsping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
```

Meaning The output shows that the VPWS is operational.

Related Documentation

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

Example: Configuring a Multisegment Pseudowire

- Understanding Multisegment Pseudowire for FEC 129 on page 85
- Example: Configuring a Multisegment Pseudowire on page 89

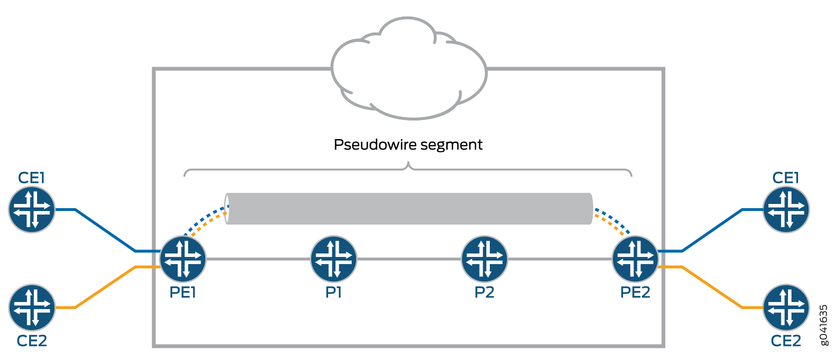
Understanding Multisegment Pseudowire for FEC 129

Understanding Multisegment Pseudowire

A pseudowire is a Layer 2 circuit or service that emulates the essential attributes of a telecommunications service, such as a T1 line, over an MPLS packet-switched network (PSN). The pseudowire is intended to provide only the minimum necessary functionality to emulate the wire with the required resiliency requirements for the given service definition.

When a pseudowire originates and terminates on the edge of the same PSN, the pseudowire label is unchanged between the originating and terminating provider edge (T-PE) devices. This is called a single-segment pseudowire (SS-PW). Figure 2 on page 8 illustrates an SS-PW established between two PE routers. The pseudowires between the PE1 and PE2 routers are located within the same autonomous system (AS).

Figure 8: L2VPN Pseudowire

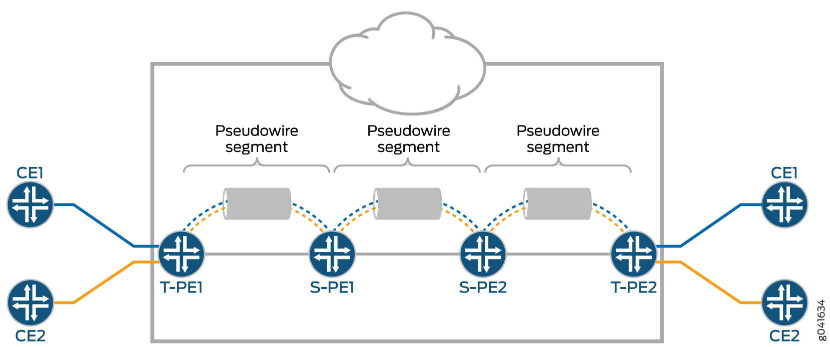


In cases where it is impossible to establish a single pseudowire from a local to a remote PE router, either because it is unfeasible or undesirable to establish a single control plane between the two PEs, a multisegment pseudowire (MS-PW) is used.

An MS-PW is a set of two or more contiguous SS-PWs that are made to function as a single point-to-point pseudowire. It is also known as switched pseudowire. MS-PWs can go across different regions or network domains. A region can be considered as an interior gateway protocol (IGP) area or a BGP autonomous system that belongs to the same or different administrative domain. An MS-PW spans multiple cores or ASs of the same or different carrier networks. A Layer 2 VPN MS-PW can include up to 254 pseudowire segments.

Figure 3 on page 9 illustrates a set of two or more pseudowire segments that function as a single pseudowire. The end routers are called terminating PE (T-PE) routers, and the switching routers are called switching PE (S-PE) routers. The S-PE router terminates the tunnels of the preceding and succeeding pseudowire segments in an MS-PW. The S-PE router can switch the control and data planes of the preceding and succeeding pseudowire segments of the MS-PW. An MS-PW is declared to be up when all the single-segment pseudowires are up.

Figure 9: Multisegment Pseudowire



Using FEC 129 for Multisegment Pseudowire

Currently, there are two types of attachment circuit identifiers (AIIs) defined under FEC 129:

- Type 1 AII
- Type 2 AII

The support of an MS-PW for FEC 129 uses type 2 AII. A type 2 AII is globally unique by definition of RFC 5003.

Single-segment pseudowires (SS-PWs) using FEC 129 on an MPLS PSN can use both type 1 and type 2 AIIs. For an MS-PW using FEC 129, a pseudowire itself is identified as a pair of endpoints. This requires that the pseudowire endpoints be uniquely identified.

In the case of a dynamically placed MS-PW, there is a requirement for the identifiers of attachment circuits to be globally unique, for the purposes of reachability and manageability of the pseudowire. Thus, individual globally unique addresses are allocated to all the attachment circuits and S-PEs that make up an MS-PW.

Type 2 AII is composed of three fields:

- Global_ID—Global identification, which is usually the AS number.
- Prefix—IPv4 address, which is usually the router ID.
- AC_ID—Local attachment circuit, which is a user-configurable value.

Since type 2 AII already contains the T-PE's IP address and it is globally unique from the FEC 129 pseudowire signaling point of view, the combination (AGI, SAII, TAII) uniquely identifies an MS-PW across all interconnected pseudowire domains.

Establishing a Multisegment Pseudowire Overview

An MS-PW is established by dynamically and automatically selecting the predefined S-PEs and placing the MS-PW between two T-PE devices.

When S-PEs are dynamically selected, each S-PE is automatically discovered and selected using the BGP autodiscovery feature, without the requirement of provisioning the FEC 129 pseudowire-related information on all the S-PEs. BGP is used to propagate pseudowire address information throughout the PSN.

Since there is no manual provisioning of FEC 129 pseudowire information on the S-PEs, the Attachment Group Identifier (AGI) and Attachment Individual Identifier (AII) are reused automatically, and choosing the same set of S-PEs for the pseudowire in both the forwarding and reverse direction is achieved through the active and passive role of each T-PE device.

- Active—The T-PE initiates an LDP label mapping message.
- Passive—The T-PE does not initiate an LDP label mapping message until it receives a label mapping message initiated by the active T-PE. The passive T-PE sends its label mapping message to the same S-PE from where it received the label mapping message originated from its active T-PE. This ensures that the same set of S-PEs are used in the reverse direction.

Pseudowire Status Support for Multisegment Pseudowire

- Pseudowire Status Behavior on T-PE on page 87
- Pseudowire Status Behavior on S-PE on page 88

Pseudowire Status Behavior on T-PE

The following pseudowire status messages are relevant on the T-PE:

- 0x00000010—Local PSN-facing pseudowire (egress) transmit fault.
- 0x00000001—Generic nonforwarding fault code. This is set as the local fault code. The local fault code is set at the local T-PE, and LDP sends a pseudowire status TLV message with the same fault code to the remote T-PE.
- Fault codes are bit-wise OR’ed and stored as remote pseudowire status codes.

Pseudowire Status Behavior on S-PE

The S-PE initiates the pseudowire status messages that indicate the pseudowire faults. The SP-PE in the pseudowire notification message hints where the fault was originated.

- When a local fault is detected by the S-PE, a pseudowire status message is sent in both directions along the pseudowire. Since there are no attachment circuits on an S-PE, only the following status messages are relevant:
  - 0x00000008—Local PSN-facing pseudowire (ingress) receive fault.
  - 0x00000010—Local PSN-facing pseudowire (egress) transmit fault.
- To indicate which SS-PW is at fault, an LDP SP-PE TLV is attached with the pseudowire status code in the LDP notification message. The pseudowire status is passed along from one pseudowire to another unchanged by the control plane switching function.
- If an S-PE initiates a pseudowire status notification message with one particular pseudowire status bit, then for the pseudowire status code an S-PE receives, the same bit is processed locally and not forwarded until the S-PE's original status error is cleared.
- An S-PE keeps only two pseudowire status codes for each SS-PW it is involved in – local pseudowire status code and remote pseudowire status code. The value of the remote pseudowire status code is the result of logic or operation of the pseudowire status codes in the chain of SS-PWs preceding this segment. This status code is incrementally updated by each S-PE upon receipt and communicated to the next S-PE. The local pseudowire status is generated locally based on its local pseudowire status.
- Only transmit fault is detected at the SP-PE. When there is no MPLS LSP to reach the next segment, a local transmit fault is detected. The transmit fault is sent to the next downstream segment, and the receive fault is sent to the upstream segment.
- Remote failures received on an S-PE are just passed along the MS-PW unchanged. Local failures are sent to both segments of the pseudowire that the S-PE is involved in.

Pseudowire TLV Support for MS-PW

MS-PW provides the following support for the LDP SP-PE TLV (RFC 6073):

- The LDP SP-PE TLVs for an MS-PW include:
  - Local IP address
  - Remote IP address
- An SP-PE adds the LDP SP-PE TLV to the label mapping message. Each SP-PE appends the local LDP SP-PE TLV to the SP-PE list it received from the other segment.
- The pseudowire status notification message includes the LDP SP-PE TLV when the notification is generated at the SP-PE.

Supported and Unsupported Features

Junos OS supports the following features with MS-PW:

- MPLS PSN for each SS-PW that builds up the MS-PW.
- The same pseudowire encapsulation for each SS-PW in an MS-PW – Ethernet or VLAN-CCC.
- The generalized PWid FEC with T-LDP as an end-to-end pseudowire signaling protocol to set up each SS-PW.
- MP-BGP to autodiscover the two endpoint PEs for each SS-PW associated with the MS-PW.
- Standard MPLS operation to stitch two side-by-side SS-PWs to form an MS-PW.
- Automatic discovery of S-PE so that the MS-PW can be dynamically placed.
- Minimum provisioning of S-PE.
- Operation, administration, and maintenance (OAM) mechanisms, including end-to-end MPLS ping or end-to-any-S-PE MPLS ping, MPLS path trace, end-to-end VCCV, and Bidirectional Forwarding Detection (BFD).
- Pseudowire switching point (SP) PE TLV for the MS-PW.
- Composite next hop on MS-PW.
- Pseudowire status TLV for MS-PW.

Junos OS does **not** support the following MS-PW functionality:

- Mix of LDP FEC 128 and LDP FEC 129.
- Static pseudowire where each label is provisioned staticially.
- Graceful Routing Engine switchover.
- Nonstop active routing.
- Multihoming.
- Partial connectivity verification (originating from an S-PE) in OAM.

Example: Configuring a Multisegment Pseudowire

This example shows how to configure a dynamic multisegment pseudowire (MS-PW), where the stitching provider edge (S-PE) devices are automatically and dynamically discovered by BGP, and pseudowires are signaled by LDP using FEC 129. This arrangement requires minimum provisioning on the S-PEs, thereby reducing the configuration burden that is associated with statically configured Layer 2 circuits while still using LDP as the underlying signaling protocol.

- Verification on page 114
- Troubleshooting on page 129

Requirements

This example uses the following hardware and software components:

- Six routers that can be a combination of M Series Multiservice Edge Routers, MX Series 3D Universal Edge Routers, T Series Core Routers, or PTX Series Packet Transport Routers.
    - Two remote PE devices configured as terminating PEs (T-PEs).
    - Two S-PEs configured as:
        - Route reflectors, in the case of interarea configuration.
        - AS boundary routers or route reflectors, in the case of inter-AS configuration.
- Junos OS Release 13.3 or later running on all the devices.

Before you begin:

1. Configure the device interfaces.
2. Configure OSPF or any other IGP protocol.
3. Configure BGP.
4. Configure LDP.
5. Configure MPLS.

Overview

Starting with Junos OS Release 13.3, you can configure an MS-PW using FEC 129 with LDP signaling and BGP autodiscovery in an MPLS packet-switched network (PSN). The MS-PW feature also provides operation, administration, and management (OAM) capabilities, such as ping, traceroute, and BFD, from the T-PE devices.

To enable autodiscovery of S-PEs in an MS-PW, include the **auto-discovery-mspw** statement at the **[edit protocols bgp group *group-name* family l2vpn]** hierarchy level.

```
family l2vpn {
  auto-discovery-mspw;
}
```

The automatic selection of S-PE and dynamic setting up of an MS-PW rely heavily on BGP. BGP network layer reachability information (NLRI) constructed for the FEC 129 pseudowire to autodiscover the S-PE is called an MS-PW NLRI [draft-ietf-pwe3-dynamic-ms-pw-15.txt]. The MS-PW NLRI is essentially a prefix consisting of a route distinguisher (RD) and FEC 129 source attachment identifier (SAII). It is referred to as a BGP autodiscovery (BGP-AD) route and is encoded as **RD:SAII**.

Only T-PEs that are provisioned with type 2 AIIs initiate their own MS-PW NLRI respectively. Since a type 2 AII is globally unique, an MS-PW NLRI is used to identify a PE device to which the type 2 AII is provisioned. The difference between a type 1 AII and a type 2 AII requires that a new address family indicator (AFI) and subsequent address family identifier (SAFI) be defined in BGP to support an MS-PW. The proposed AFI and SAFI value pair used to identify the MS-PW NLRI is 25 and 6, respectively (pending IANA allocation).

The AFI and SAFI values support autodiscovery of S-PEs and should be configured on both T-PEs that originate the routes, and the S-PEs that participate in the signaling.

Figure 10 on page 91 illustrates an inter-area MS-PW setup between two remote PE routers—T-PE1 and T-PE2. The Provider (P) routers are P1 and P2, and the S-PE routers are S-PE1 and S-PE2. The MS-PW is established between T-PE1 and T-PE2, and all the devices belong to the same AS—AS 100. Since S-PE1 and S-PE2 belong to the same AS, they act as route reflectors and are also known as RR 1 and RR 2, respectively.

Figure 11 on page 92 illustrates an inter-AS MS-PW setup. The MS-PW is established between T-PE1 and T-PE2, where T-PE1, P1, and S-PE1 belong to AS 1, and S-PE2, P2, and T-PE2 belong to AS 2. Since S-PE1 and S-PE2 belong to different ASs, they are configured as ASBR routers and are also known as ASBR 1 and ASBR 2, respectively.

Figure 10: Interarea Multisegment Pseudowire

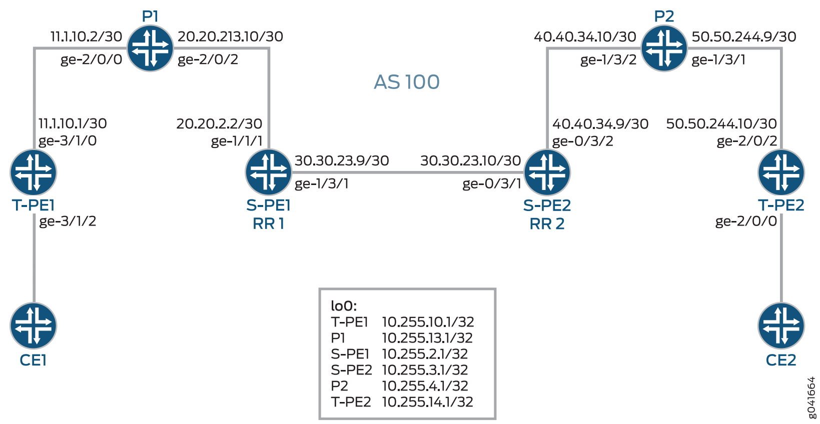
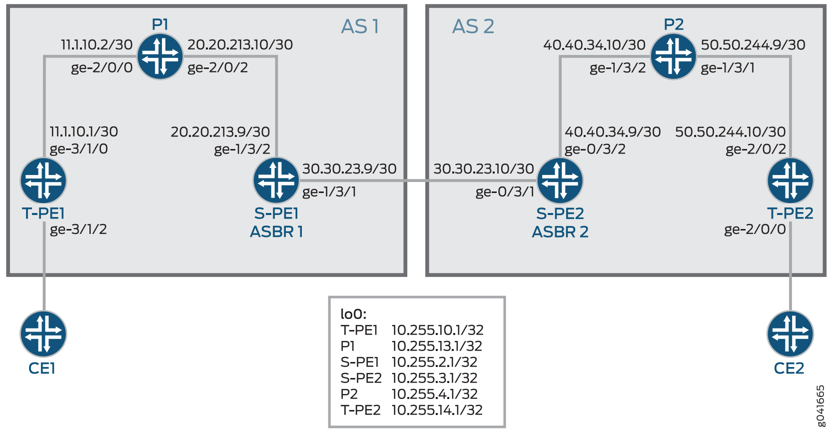


Figure 11: Inter-AS Multisegment Pseudowire



The following sections provide information about how an MS-PW is established in an interarea and inter-AS scenario.

Minimum Configuration Requirements on S-PE

In order to dynamically discover both ends of an SS-PW and set up a T-LDP session dynamically, the following is required:

- For interarea MS-PW, each S-PE plays both an ABR and BGP route reflector role.

  In the interarea case, as seen in Figure 10 on page 91, the S-PE plays a BGP route reflector role and reflects the BGP-AD route to its client. A BGP-AD route advertised by one T-PE eventually reaches its remote T-PE. Because of the next-hop-self set by each S-PE, the S-PE or T-PE that receives a BGP-AD route can always discover the S-PE that advertises the BGP-AD in its local AS or local area through the BGP next hop.

- For inter-AS MS-PW, each S-PE plays either an ASBR or a BGP route reflector role.

  In an MS-PW, the two T-PEs initiate a BGP-AD route respectively. When the S-PE receives the BGP-AD route through either the IBGP session with the T-PE or through a regular BGP-RR, it sets the next-hop-self before re-advertising the BGP-AD route to one or more of its EBGP peers in the inter-AS case, as seen in Figure 11 on page 92.

- Each S-PE must set next-hop-self when re-advertising or reflecting a BGP-AD route for the MS-PW.

Active and Passive Role of T-PE

To ensure that the same set of S-PEs are being used for a MS-PW in both directions, the two T-PEs play different roles in terms of FEC 129 signaling. This is to avoid different paths being chosen by T-PE1 and T-PE2 when each S-PE is dynamically selected for an MS-PW.

When an MS-PW is signaled using FEC 129, each T-PE might independently start signaling the MS-PW. The signaling procedure can result in an attempt to set up each direction of the MS-PW through different S-PEs.

To avoid this situation, one of the T-PEs must start the pseudowire signaling (active role), while the other waits to receive the LDP label mapping before sending the respective pseudowire LDP label mapping message (passive role). When the MS-PW path is dynamically placed, the active T-PE (the Source T-PE) and the passive T-PE (the Target T-PE) must be identified before signaling is initiated for a given MS-PW. The determination of which T-PE assumes the active role is done based on the SAII value, where the T-PE that has a larger SAII value plays the active role.

In this example, the SAII values of T-PE1 and T-PE 2 are **800:800:800** and **700:700:700**, respectively. Since T-PE1 has a higher SAII value, it assumes the active role and T-PE2 assumes the passive role.

Directions for Establishing an MS-PW

The directions used by the S-PE for setting up the MS-PW are:

- Forwarding direction—From an active T-PE to a passive T-PE.

  In this direction, the S-PEs perform a BGP-AD route lookup to determine the next-hop S-PE to send the label mapping message.
- Reverse direction—From a passive T-PE to an active T-PE.

  In this direction, the S-PEs do not perform a BGP-AD route lookup, because the label mapping messages are received from the T-PEs, and the stitching routes are installed in the S-PEs.

In this example, the MS-PW is established in the forwarding direction from T-PE1 to T-PE2. When the MS-PW is placed from T-PE2 to T-PE1, the MS-PW is established in the reverse direction.

Autodiscovery and Dynamic Selection of S-PE

A new AFI and SAFI value is defined in BGP to support the MS-PWs based on type 2 AII. This new address family supports autodiscovery of S-PEs. This address family must be configured on both the TPEs and SPEs.

It is the responsibility of the Layer 2 VPN component to dynamically select the next S-PE to use along the MS-PW in the forwarding direction.

- In the forwarding direction, the selection of the next S-PE is based on the BGP-AD route advertised by the BGP and pseudowire FEC information sent by the LDP. The BGP-AD route is initiated by the passive T-PE (T-PE2) in the reverse direction while the pseudowire FEC information is sent by LDP from the active T-PE (T-PE1) in the forwarding direction.
- In the reverse direction, the next S-PE (S-PE2) or the active T-PE (T-PE1) is obtained by looking up the S-PE (S-PE1) that it used to set up the pseudowire in the forwarding direction.

Provisioning a T-PE

To support FEC 129 type 2 AII, the T-PE needs to configure its remote T-PE's IP address, a global ID, and an attachment circuit ID. Explicit paths where a set of S-PEs to use is explicitly specified on a T-PE is not supported. This eliminates the need to provision each S-PE with a type 2 AII.

Stitching an MS-PW

An S-PE performs the following MPLS label operations before forwarding the received label mapping message to the next S-PE:

1. Pops the MPLS tunnel label.
2. Pops the VC label.
3. Pushes a new VC label.
4. Pushes an MPLS tunnel label used for the next segment.

Establishing an MS-PW

After completing the necessary configuration, an MS-PW is established in the following manner:

1. The SAII values are exchanged between T-PE1 and T-PE2 using BGP.

   T-PE1 assumes the active T-PE role, because it is configured with a higher SAII value. T-PE2 becomes the passive T-PE.
2. T-PE1 receives the BGP-AD route originated by T-PE2. It compares the AII values obtained from T-PE2 in the received BGP-AD route against the AII values provisioned locally.
3. If the AII values match, T-PE1 performs a BGP-AD route lookup to elect the first S-PE (S-PE1).
4. T-PE1 sends an LDP label mapping message to S-PE1.
5. Using the BGP-AD route originated from T-PE2, and the LDP label mapping message received from T-PE1, S-PE1 selects the next S-PE (S-PE2) in the forwarding direction.

   To do this, S-PE1 compares SAII obtained from the BGP-AD route against the TAI from the LDP label mapping message.
6. If the AII values match, S-PE1 finds S-PE2 through the BGP next hop associated with the BGP-AD route.
7. The process of selecting S-PE goes on until the last S-PE establishes a T-LDP session with T-PE2. When T-PE2 receives the LDP label mapping message from the last S-PE (S-PE2), it initiates its own label mapping message and sends it back to S-PE2.
8. When all the label mapping messages are received on S-PE1 and S-PE2, the S-PEs install the stitching routes. Thus, when the MS-PW is established in the reverse direction, the S-PEs need not perform BGP-AD route lookup to determine its next hop as it did in the forwarding direction.

OAM Support for an MS-PW

After the MS-PW is established, the following OAM capabilities can be executed from the T-PE devices:

- Ping
  - End-to-End Connectivity Verification Between T-PEs

    If T-PE1, S-PEs, and T-PE2 support Control Word (CW), the pseudowire control plane automatically negotiates the use of the CW. Virtual Circuit Connectivity Verification (VCCV) Control Channel (CC) Type 3 will function correctly whether or not the CW is enabled on the pseudowire. However, VCCV Type 1, which is used for end-to-end verification only, is only supported if the CW is enabled.

    The following is a sample:

    `user@T-PE1>` **ping mpls l2vpn fec129 instance *instance-name* local-id *SAII of T-PE1* remote-pe-address *address of T-PE2* remote-id *TAII of T-PE2***

    or

    `user@T-PE1>` **ping mpls l2vpn fec129 interface *CE1-facing interface***
  - Partial Connectivity Verification from T-PE to Any S-PE

    To trace part of an MS-PW, the TTL of the pseudowire label can be used to force the VCCV message to pop out at an intermediate node. When the TTL expires, the S-PE can determine that the packet is a VCCV packet either by checking the CW or by checking for a valid IP header with UDP destination port 3502 (if the CW is not in use). The packet should then be diverted to VCCV processing.

    If T-PE1 sends a VCCV message with the TTL of the pseudowire label equal to 1, the TTL expires at the S-PE. T-PE1 can thus verify the first segment of the pseudowire.

    The VCCV packet is built according to RFC 4379. All the information necessary to build the VCCV LSP ping packet is collected by inspecting the S-PE TLVs. This use of the TTL is subject to the caution expressed in RFC 5085. If a penultimate LSR between S-PEs or between an S-PE and a T-PE manipulates the pseudowire label TTL, the VCCV message might not emerge from the MS-PW at the correct S-PE.

    The following is a sample:

    `user@T-PE1>` **ping mpls l2vpn fec129 interface *CE1-facing interface* bottom-label-ttl *segment***

    The **bottom-label-ttl** value is 1 for S-PE1 and 2 for S-PE2.

    The **bottom-label-ttl** statement sets the correct VC label TTL, so the packets are popped to the correct SS-PW for VCCV processing.

    NOTE: **Junos OS supports VCCV Type 1 and Type 3 for the MS-PW OAM capability. VCCV Type 2 is not supported.**

- Traceroute

  Traceroute tests each S-PE along the path of the MS-PW in a single operation similar to LSP trace. This operation is able to determine the actual data path of the MS-PW, and is used for dynamically signaled MS-PWs.

user@T-PE1> **traceroute mpls l2vpn fec129 interface *CE1-facing interface***

- Bidirectional Forwarding Detection

  Bidirectional Forwarding Detection (BFD) is a detection protocol designed to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols. In addition to fast forwarding path failure detection, BFD provides a consistent failure detection method for network administrators. The router or switch can be configured to log a system log (syslog) message when BFD goes down.

  user@T-PE1> **show bfd session extensive**

Configuration

Configuring an Interarea MS-PW

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.

T-PE1

```
set interfaces ge-3/1/0 unit 0 family inet address 11.1.10.1/30
set interfaces ge-3/1/0 unit 0 family mpls
set interfaces ge-3/1/2 encapsulation ethernet-ccc
set interfaces ge-3/1/2 unit 0
set interfaces lo0 unit 0 family inet address 10.255.10.1/32 primary
set routing-options autonomous-system 100
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp family l2vpn auto-discovery-mspw
set protocols bgp group mspw type internal
set protocols bgp group mspw local-address 10.255.10.1
set protocols bgp group mspw neighbor 10.255.2.1
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols ldp interface lo0.0
set routing-instances ms-pw instance-type l2vpn
set routing-instances ms-pw interface ge-3/1/2.0
set routing-instances ms-pw route-distinguisher 10.10.10.10:15
set routing-instances ms-pw l2vpn-id l2vpn-id:100:15
set routing-instances ms-pw vrf-target target:100:115
set routing-instances ms-pw protocols l2vpn site CE1 source-attachment-identifier
  800:800:800
set routing-instances ms-pw protocols l2vpn site CE1 interface ge-3/1/2.0
  target-attachment-identifier 700:700:700
set routing-instances ms-pw protocols l2vpn pseudowire-status-tlv
set routing-instances ms-pw protocols l2vpn oam bfd-liveness-detection
  minimum-interval 300
```


P1

```
set interfaces ge-2/0/0 unit 0 family inet address 11.1.10.2/30
set interfaces ge-2/0/0 unit 0 family mpls
set interfaces ge-2/0/2 unit 0 family inet address 20.20.213.10/30
set interfaces ge-2/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.13.1/32 primary
set routing-options autonomous-system 100
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols ldp interface lo0.0
```

S-PE1 (RR 1)

```
set interfaces ge-1/3/1 unit 0 family inet address 30.30.23.9/30
set interfaces ge-1/3/1 unit 0 family mpls
set interfaces ge-1/3/2 unit 0 family inet address 20.20.213.9/30
set interfaces ge-1/3/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.2.1/32 primary
set routing-options autonomous-system 100
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp family l2vpn auto-discovery-mspw
set protocols bgp group mspw type internal
set protocols bgp group mspw local-address 10.255.2.1
set protocols bgp group mspw export next-hop-self
set protocols bgp group mspw cluster 1.1.1.1
set protocols bgp group mspw neighbor 10.255.10.1
set protocols bgp group mspw neighbor 10.255.3.1
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols ldp interface lo0.0
set policy-options policy-statement next-hop-self then next-hop self
set policy-options policy-statement send-inet0 from protocol bgp
set policy-options policy-statement send-inet0 then accept
```

S-PE2 (RR 2)

```
set interfaces ge-0/3/1 unit 0 family inet address 30.30.23.10/30
set interfaces ge-0/3/1 unit 0 family mpls
set interfaces ge-0/3/2 unit 0 family inet address 40.40.34.9/30
set interfaces ge-0/3/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.3.1/32 primary
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp family l2vpn auto-discovery-mspw
set protocols bgp group mspw type internal
set protocols bgp group mspw local-address 10.255.3.1
set protocols bgp group mspw export next-hop-self
set protocols bgp group mspw cluster 2.2.2.2
set protocols bgp group mspw neighbor 10.255.2.1
set protocols bgp group mspw neighbor 10.255.14.1
set protocols bgp group int type internal
```

```
set protocols bgp group int local-address 10.255.3.1
set protocols bgp group int neighbor 10.255.2.1
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols ldp interface lo0.0
set policy-options policy-statement next-hop-self then next-hop self
set policy-options policy-statement send-inet0 from protocol bgp
set policy-options policy-statement send-inet0 then accept
```

P2

```
set interfaces ge-1/3/1 unit 0 family inet address 50.50.244.9/30
set interfaces ge-1/3/1 unit 0 family mpls
set interfaces ge-1/3/2 unit 0 family inet address 40.40.34.10/30
set interfaces ge-1/3/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.4.1/32 primary
set routing-options autonomous-system 100
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols ldp interface lo0.0
```

T-PE2

```
set interfaces ge-2/0/0 encapsulation ethernet-ccc
set interfaces ge-2/0/0 unit 0
set interfaces ge-2/0/2 unit 0 family inet address 50.50.244.10/30
set interfaces ge-2/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.14.1/32 primary
set routing-options autonomous-system 100
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp family l2vpn auto-discovery-mspw
set protocols bgp group mspw type internal
set protocols bgp group mspw local-address 10.255.14.1
set protocols bgp group mspw neighbor 10.255.3.1
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols ldp interface lo0.0
set routing-instances ms-pw instance-type l2vpn
set routing-instances ms-pw interface ge-2/0/0.0
set routing-instances ms-pw route-distinguisher 10.10.10.10:15
set routing-instances ms-pw l2vpn-id l2vpn-id:100:15
set routing-instances ms-pw vrf-target target:100:115
set routing-instances ms-pw protocols l2vpn site CE2 source-attachment-identifier
  700:700:700
set routing-instances ms-pw protocols l2vpn site CE2 interface ge-2/0/0.0
  target-attachment-identifier 800:800:800
set routing-instances ms-pw protocols l2vpn pseudowire-status-tlv
```


set routing-instances ms-pw protocols l2vpn oam bfd-liveness-detection minimum-interval 300

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*.

To configure T-PE1 in the interarea scenario:

NOTE: Repeat this procedure for the T-PE2 device in the MPLS domain, after modifying the appropriate interface names, addresses, and other parameters.

1. Configure the T-PE1 interfaces.

```
[edit interfaces]
user@T-PE1# set ge-3/1/0 unit 0 family inet address 11.1.10.1/30
user@T-PE1# set ge-3/1/0 unit 0 family mpls

user@T-PE1# set ge-3/1/2 encapsulation ethernet-ccc
user@T-PE1# set ge-3/1/2 unit 0

user@T-PE1# set lo0 unit 0 family inet address 10.255.10.1/32 primary
```

2. Set the autonomous system number.

```
[edit routing-options]
user@T-PE1# set autonomous-system 100
```

3. Enable MPLS on all the interfaces of T-PE1, excluding the management interface.

```
[edit protocols]
user@T-PE1# set mpls interface all
user@T-PE1# set mpls interface fxp0.0 disable
```

4. Enable autodiscovery of intermediate S-PEs that make up the MS-PW using BGP.

```
[edit protocols]
user@T-PE1# set bgp family l2vpn auto-discovery-mspw
```

5. Configure the BGP group for T-PE1.

```
[edit protocols]
user@T-PE1# set bgp group mspw type internal
```

6. Assign local and neighbor addresses to the mspw group for T-PE1 to peer with S-PE1.

```
[edit protocols]
user@T-PE1# set bgp group mspw local-address 10.255.10.1
user@T-PE1# set bgp group mspw neighbor 10.255.2.1
```

7. Configure OSPF on all the interfaces of T-PE1, excluding the management interface.

```
[edit protocols]
user@T-PE1# set ospf area 0.0.0.0 interface lo0.0
user@T-PE1# set ospf area 0.0.0.0 interface all
```

user@T-PE1# **set ospf area 0.0.0.0 interface fxp0.0 disable**

8. Configure LDP on all the interfaces of T-PE1, excluding the management interface.

   [edit protocols]
   user@T-PE1# **set ldp interface all**
   user@T-PE1# **set ldp interface fxp0.0 disable**
   user@T-PE1# **set ldp interface lo0.0**

9. Configure the Layer 2 VPN routing instance on T-PE1.

   [edit routing-instances]
   user@T-PE1# **set ms-pw instance-type l2vpn**

10. Assign the interface name for the mspw routing instance.

    [edit routing-instances]
    user@T-PE1# **set ms-pw interface ge-3/1/2.0**

11. Configure the route distinguisher for the mspw routing instance.

    [edit routing-instances]
    user@T-PE1# **set ms-pw route-distinguisher 10.10.10.10:15**

12. Configure the Layer 2 VPN ID community for FEC 129 MS-PW.

    [edit routing-instances]
    user@T-PE1# **set ms-pw l2vpn-id l2vpn-id:100:15**

13. Configure a VPN routing and forwarding (VRF) target for the mspw routing instance.

    [edit routing-instances]
    user@T-PE1# **set ms-pw vrf-target target:100:115**

14. Configure the source attachment identifier (SAI) value using Layer 2 VPN as the routing protocol for the mspw routing instance.

    [edit routing-instances]
    user@T-PE1# **set ms-pw protocols l2vpn site CE1 source-attachment-identifier 800:800:800**

15. Assign the interface name that connects the CE1 site to the VPN, and configure the target attachment identifier (TAI) value using Layer 2 VPN as the routing protocol for the mspw routing instance.

    [edit routing-instances]
    user@T-PE1# **set ms-pw protocols l2vpn site CE1 interface ge-3/1/2.0 target-attachment-identifier 700:700:700**

16. (Optional) Configure T-PE1 to send MS-PW status TLVs.

    [edit routing-instances]
    user@T-PE1# **set ms-pw protocols l2vpn pseudowire-status-tlv**

17. (Optional) Configure OAM capabilities for the VPN.

    [edit routing-instances]
    user@T-PE1# **set ms-pw protocols l2vpn oam bfd-liveness-detection minimum-interval 300**

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*.

To configure S-PE1 (RR 1) in the interarea scenario:

NOTE: **Repeat this procedure for the S-PE2 (RR 2) device in the MPLS domain, after modifying the appropriate interface names, addresses, and other parameters.**

1. Configure the S-PE1 interfaces.

   ```
   [edit interfaces]
   user@S-PE1# set ge-1/3/1 unit 0 family inet address 30.30.23.9/30
   user@S-PE1# set ge-1/3/1 unit 0 family mpls

   user@S-PE1# set ge-1/3/2 unit 0 family inet address 20.20.213.9/30
   user@S-PE1# set ge-1/3/2 unit 0 family mpls

   user@S-PE1# set lo0 unit 0 family inet address 10.255.2.1/32 primary
   ```

2. Set the autonomous system number.

   ```
   [edit routing-options]
   user@S-PE1# set autonomous-system 100
   ```

3. Enable MPLS on all the interfaces of T-PE1, excluding the management interface.

   ```
   [edit protocols]
   user@S-PE1# set mpls interface all
   user@S-PE1# set mpls interface fxp0.0 disable
   ```

4. Enable autodiscovery of S-PE using BGP.

   ```
   [edit protocols]
   user@S-PE1# set bgp family l2vpn auto-discovery-mspw
   ```

5. Configure the BGP group for S-PE1.

   ```
   [edit protocols]
   user@S-PE1# set bgp group mspw type internal
   ```

6. Configure S-PE1 to act as a route reflector.

   ```
   [edit protocols]
   user@S-PE1# set bgp group mspw export next-hop-self
   user@S-PE1# set bgp group mspw cluster 1.1.1.1
   ```

7. Assign local and neighbor addresses to the mspw group for S-PE1 to peer with T-PE1 and S-PE2.

   ```
   [edit protocols]
   user@S-PE1# set bgp group mspw local-address 10.255.2.1
   user@S-PE1# set bgp group mspw neighbor 10.255.10.1 (to T-PE1)
   user@S-PE1# set bgp group mspw neighbor 10.255.3.1 (to S-PE2)
   ```

8. Configure OSPF on all the interfaces of S-PE1, excluding the management interface.

```
[edit protocols]
user@S-PE1# set ospf area 0.0.0.0 interface all
user@S-PE1# set ospf area 0.0.0.0 interface fxp0.0 disable
user@S-PE1# set ospf area 0.0.0.0 interface lo0.0
```

9. Configure LDP on all the interfaces of S-PE1, excluding the management interface.

```
[edit protocols]
user@S-PE1# set ldp interface all
user@S-PE1# set ldp interface fxp0.0 disable
user@S-PE1# set ldp interface lo0.0
```

10. Define the policy for enabling next-hop-self and accepting BGP traffic on S-PE1.

```
[edit policy-options]
user@S-PE1# set policy-statement next-hop-self then next-hop self
user@S-PE1# set policy-statement send-inet0 from protocol bgp
user@S-PE1# set policy-statement send-inet0 then accept
```

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

T-PE1

```
user@T-PE1# show interfaces
ge-3/1/0 {
  unit 0 {
    family inet {
      address 11.1.10.1/30;
    }
    family mpls;
  }
}
ge-3/1/2 {
  encapsulation ethernet-ccc;
  unit 0;
}
lo0 {
  unit 0 {
    family inet {
      address 10.255.10.1/32 {
        primary;
      }
    }
  }
}
```

```
user@T-PE1# show routing-options
autonomous-system 100;
```

```
user@T-PE1# show protocols
mpls {
  interface all;
  interface fxp0.0 {
    disable;
```

```
    }
}
bgp {
    family l2vpn {
        auto-discovery-mspw;
    }
    group mspw {
        type internal;
        local-address 10.255.10.1;
        neighbor 10.255.2.1;
    }
}
ospf {
    area 0.0.0.0 {
        interface all;
        interface fxp0.0 {
            disable;
        }
        interface lo0.0;
    }
}
ldp {
    interface all;
    interface fxp0.0 {
        disable;
    }
    interface lo0.0;
}
```

```
user@T-PE1# show routing-instances
ms-pw {
    instance-type l2vpn;
    interface ge-3/1/2.0;
    route-distinguisher 10.10.10.10:15;
    l2vpn-id l2vpn-id:100:15;
    vrf-target target:100:115;
    protocols {
        l2vpn {
            site CE1 {
                source-attachment-identifier 800:800:800;
                interface ge-3/1/2.0 {
                    target-attachment-identifier 700:700:700;
                }
            }
            pseudowire-status-tlv;
            oam {
                bfd-liveness-detection {
                    minimum-interval 300;
                }
            }
        }
    }
}
```

S-PE1 (RR 1)

```
user@S-PE1# show interfaces
ge-1/3/1 {
    unit 0 {
```

```
            family inet {
                address 30.30.23.9/30;
            }
            family mpls;
        }
    }
    ge-1/3/2 {
        unit 0 {
            family inet {
                address 20.20.213.9/30;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.2.1/32 {
                    primary;
                }
            }
        }
    }
```

```
user@S-PE1# show routing-options
autonomous-system 100;
```

```
user@S-PE1# show protocols
mpls {
    interface all;
    interface fxp0.0 {
        disable;
    }
}
bgp {
    family l2vpn {
        auto-discovery-mspw;
    }
    group mspw {
        type internal;
        local-address 10.255.2.1;
        export next-hop-self;
        cluster 1.1.1.1;
        neighbor 10.255.10.1;
        neighbor 10.255.3.1;
    }
}
ospf {
    area 0.0.0.0 {
        interface lo0.0;
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
}
```



```
ldp {
   interface all;
   interface fxp0.0 {
      disable;
   }
   interface lo0.0;
}
```

```
user@S-PE1# show policy-options
policy-statement next-hop-self {
   then {
      next-hop self;
   }
}
policy-statement send-inet0 {
   from protocol bgp;
   then accept;
}
```

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

Configuring an Inter-AS MS-PW

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.

T-PE1

```
set interfaces ge-3/1/0 unit 0 family inet address 11.1.10.1/30
set interfaces ge-3/1/0 unit 0 family mpls
set interfaces ge-3/1/2 encapsulation ethernet-ccc
set interfaces ge-3/1/2 unit 0
set interfaces lo0 unit 0 family inet address 10.255.10.1/32 primary
set routing-options autonomous-system 1
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp family l2vpn auto-discovery-mspw
set protocols bgp group mspw type internal
set protocols bgp group mspw local-address 10.255.10.1
set protocols bgp group mspw neighbor 10.255.2.1
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols ldp interface lo0.0
set routing-instances ms-pw instance-type l2vpn
set routing-instances ms-pw interface ge-3/1/2.0
set routing-instances ms-pw route-distinguisher 10.10.10.10:15
set routing-instances ms-pw l2vpn-id l2vpn-id:100:15
set routing-instances ms-pw vrf-target target:100:115
set routing-instances ms-pw protocols l2vpn site CE1 source-attachment-identifier
   800:800:800
set routing-instances ms-pw protocols l2vpn site CE1 interface ge-3/1/2.0
   target-attachment-identifier 700:700:700
set routing-instances ms-pw protocols l2vpn pseudowire-status-tlv
```

```
set routing-instances ms-pw protocols l2vpn oam bfd-liveness-detection
   minimum-interval 300
```

P1

```
set interfaces ge-2/0/0 unit 0 family inet address 11.1.10.2/30
set interfaces ge-2/0/0 unit 0 family mpls
set interfaces ge-2/0/2 unit 0 family inet address 20.20.213.10/30
set interfaces ge-2/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.13.1/32 primary
set routing-options autonomous-system 1
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols ldp interface lo0.0
```

S-PE1 (ASBR 1)

```
set interfaces ge-1/3/1 unit 0 family inet address 30.30.23.9/30
set interfaces ge-1/3/1 unit 0 family mpls
set interfaces ge-1/3/2 unit 0 family inet address 20.20.213.9/30
set interfaces ge-1/3/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.2.1/32 primary
set routing-options autonomous-system 1
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp family l2vpn auto-discovery-mspw
set protocols bgp group to_T-PE1 type internal
set protocols bgp group to_T-PE1 local-address 10.255.2.1
set protocols bgp group to_T-PE1 export next-hop-self
set protocols bgp group to_T-PE1 neighbor 10.255.10.1
set protocols bgp group to_S-PE2 type external
set protocols bgp group to_S-PE2 local-address 10.255.2.1
set protocols bgp group to_S-PE2 peer-as 2
set protocols bgp group to_S-PE2 neighbor 10.255.3.1 multihop ttl 1
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols ldp interface lo0.0
set policy-options policy-statement next-hop-self then next-hop self
```

S-PE2 (ASBR 2)

```
set interfaces ge-0/3/1 unit 0 family inet address 30.30.23.10/30
set interfaces ge-0/3/1 unit 0 family mpls
set interfaces ge-0/3/2 unit 0 family inet address 40.40.34.9/30
set interfaces ge-0/3/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.3.1/32 primary
set routing-options autonomous-system 2
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp family l2vpn auto-discovery-mspw
set protocols bgp group to_T-PE2 type internal
set protocols bgp group to_T-PE2 local-address 10.255.3.1
set protocols bgp group to_T-PE2 export next-hop-self
```



```
set protocols bgp group to_T-PE2 neighbor 10.255.14.1
set protocols bgp group to_S-PE1 type external
set protocols bgp group to_S-PE1 local-address 10.255.3.1
set protocols bgp group to_S-PE1 peer-as 1
set protocols bgp group to_S-PE1 neighbor 10.255.2.1 multihop ttl 1
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols ldp interface lo0.0
set policy-options policy-statement next-hop-self then next-hop self
```

P2

```
set interfaces ge-1/3/1 unit 0 family inet address 50.50.244.9/30
set interfaces ge-1/3/1 unit 0 family mpls
set interfaces ge-1/3/2 unit 0 family inet address 40.40.34.10/30
set interfaces ge-1/3/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.4.1/32 primary
set routing-options autonomous-system 2
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols ldp interface lo0.0
```

T-PE2

```
set interfaces ge-2/0/0 encapsulation ethernet-ccc
set interfaces ge-2/0/0 unit 0
set interfaces ge-2/0/2 unit 0 family inet address 50.50.244.10/30
set interfaces ge-2/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.14.1/32 primary
set routing-options autonomous-system 2
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp family l2vpn auto-discovery-mspw
set protocols bgp group mspw type internal
set protocols bgp group mspw local-address 10.255.14.1
set protocols bgp group mspw neighbor 10.255.3.1
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols ldp interface lo0.0
set routing-instances ms-pw instance-type l2vpn
set routing-instances ms-pw interface ge-2/0/0.0
set routing-instances ms-pw route-distinguisher 10.10.10.10:15
set routing-instances ms-pw l2vpn-id l2vpn-id:100:15
set routing-instances ms-pw vrf-target target:100:115
set routing-instances ms-pw protocols l2vpn site CE2 source-attachment-identifier
  700:700:700
set routing-instances ms-pw protocols l2vpn site CE2 interface ge-2/0/0.0
  target-attachment-identifier 800:800:800
```

```
set routing-instances ms-pw protocols l2vpn pseudowire-status-tlv
set routing-instances ms-pw protocols l2vpn oam bfd-liveness-detection
  minimum-interval 300
```

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*.

To configure the T-PE1 router in the inter-AS scenario:

NOTE: Repeat this procedure for the T-PE2 device in the MPLS domain, after modifying the appropriate interface names, addresses, and other parameters.

1. Configure the T-PE1 interfaces.

   ```
   [edit interfaces]
   user@T-PE1# set ge-3/1/0 unit 0 family inet address 11.1.10.1/30
   user@T-PE1# set ge-3/1/0 unit 0 family mpls

   user@T-PE1# set ge-3/1/2 encapsulation ethernet-ccc
   user@T-PE1# set ge-3/1/2 unit 0

   user@T-PE1# set lo0 unit 0 family inet address 10.255.10.1/32 primary
   ```

2. Set the autonomous system number.

   ```
   [edit routing-options]
   user@T-PE1# set autonomous-system 1
   ```

3. Enable MPLS on all the interfaces of T-PE1, excluding the management interface.

   ```
   [edit protocols]
   user@T-PE1# set mpls interface all
   user@T-PE1# set mpls interface fxp0.0 disable
   ```

4. Enable autodiscovery of intermediate S-PEs that make up the MS-PW using BGP.

   ```
   [edit protocols]
   user@T-PE1# set bgp family l2vpn auto-discovery-mspw
   ```

5. Configure the BGP group for T-PE1.

   ```
   [edit protocols]
   user@T-PE1# set bgp group mspw type internal
   ```

6. Assign local and neighbor addresses to the mspw group for T-PE1 to peer with S-PE1.

   ```
   [edit protocols]
   user@T-PE1# set bgp group mspw local-address 10.255.10.1
   user@T-PE1# set bgp group mspw neighbor 10.255.2.1
   ```

7. Configure OSPF on all the interfaces of T-PE1, excluding the management interface.

   ```
   [edit protocols]
   user@T-PE1# set ospf area 0.0.0.0 interface lo0.0
   ```

```
user@T-PE1# set ospf area 0.0.0.0 interface all
user@T-PE1# set ospf area 0.0.0.0 interface fxp0.0 disable
```

8. Configure LDP on all the interfaces of T-PE1, excluding the management interface.

```
[edit protocols]
user@T-PE1# set ldp interface all
user@T-PE1# set ldp interface fxp0.0 disable
user@T-PE1# set ldp interface lo0.0
```

9. Configure the Layer 2 VPN routing instance on T-PE1.

```
[edit routing-instances]
user@T-PE1# set ms-pw instance-type l2vpn
```

10. Assign the interface name for the mspw routing instance.

```
[edit routing-instances]
user@T-PE1# set ms-pw interface ge-3/1/2.0
```

11. Configure the route distinguisher for the mspw routing instance.

```
[edit routing-instances]
user@T-PE1# set ms-pw route-distinguisher 10.10.10.10:15
```

12. Configure the Layer 2 VPN ID community for FEC 129 MS-PW.

```
[edit routing-instances]
user@T-PE1# set ms-pw l2vpn-id l2vpn-id:100:15
```

13. Configure a VPN routing and forwarding (VRF) target for the mspw routing instance.

```
[edit routing-instances]
user@T-PE1# set ms-pw vrf-target target:100:115
```

14. Configure the source attachment identifier (SAI) value using Layer 2 VPN as the routing protocol for the mspw routing instance.

```
[edit routing-instances]
user@T-PE1# set ms-pw protocols l2vpn site CE1 source-attachment-identifier
  800:800:800
```

15. Assign the interface name that connects the CE1 site to the VPN, and configure the target attachment identifier (TAI) value using Layer 2 VPN as the routing protocol for the mspw routing instance.

```
[edit routing-instances]
user@T-PE1# set ms-pw protocols l2vpn site CE1 interface ge-3/1/2.0
  target-attachment-identifier 700:700:700
```

16. (Optional) Configure T-PE1 to send MS-PW status TLVs.

```
[edit routing-instances]
user@T-PE1# set ms-pw protocols l2vpn pseudowire-status-tlv
```

17. (Optional) Configure OAM capabilities for the VPN.

```
[edit routing-instances]
user@T-PE1# set ms-pw protocols l2vpn oam bfd-liveness-detection
  minimum-interval 300
```


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*.

To configure S-PE1 (ASBR 1) in the inter-AS scenario:

NOTE: Repeat this procedure for the S-PE2 (ASBR 2) device in the MPLS domain, after modifying the appropriate interface names, addresses, and other parameters.

1. Configure S-PE1 (ASBR 1) interfaces.

   ```
   [edit interfaces]
   user@S-PE1# set ge-1/3/1 unit 0 family inet address 30.30.23.9/30
   user@S-PE1# set ge-1/3/1 unit 0 family mpls

   user@S-PE1# set ge-1/3/2 unit 0 family inet address 20.20.213.9/30
   user@S-PE1# set ge-1/3/2 unit 0 family mpls

   user@S-PE1# set lo0 unit 0 family inet address 10.255.2.1/32 primary
   ```

2. Set the autonomous system number.

   ```
   [edit routing-options]
   user@S-PE1# set autonomous-system 1
   ```

3. Enable MPLS on all the interfaces of S-PE1 (ASBR 1), excluding the management interface.

   ```
   [edit protocols]
   user@S-PE1# set mpls interface all
   user@S-PE1# set mpls interface fxp0.0 disable
   ```

4. Enable autodiscovery of S-PE using BGP.

   ```
   [edit protocols]
   user@S-PE1# set bgp family l2vpn auto-discovery-mspw
   ```

5. Configure the IBGP group for S-PE1 (ASBR 1) to peer with T-PE1.

   ```
   [edit protocols]
   user@S-PE1# set bgp group to_T-PE1 type internal
   ```

6. Configure the IBGP group parameters.

   ```
   [edit protocols]
   user@S-PE1# set bgp group to_T-PE1 local-address 10.255.2.1
   user@S-PE1# set bgp group to_T-PE1 export next-hop-self
   user@S-PE1# set bgp group to_T-PE1 neighbor 10.255.10.1
   ```

7. Configure the EBGP group for S-PE1 (ASBR 1) to peer with S-PE2 (ASBR 2).

   ```
   [edit protocols]
   user@S-PE1# set bgp group to_S-PE2 type external
   ```

8. Configure the EBGP group parameters.

```
[edit protocols]
user@S-PE1# set bgp group to_S-PE2 local-address 10.255.2.1
user@S-PE1# set bgp group to_S-PE2 peer-as 2
user@S-PE1# set bgp group to_S-PE2 neighbor 10.255.3.1 multihop ttl 1
```

9. Configure OSPF on all the interfaces of S-PE1 (ASBR 1), excluding the management interface.

```
[edit protocols]
user@S-PE1# set ospf area 0.0.0.0 interface all
user@S-PE1# set ospf area 0.0.0.0 interface fxp0.0 disable
user@S-PE1# set ospf area 0.0.0.0 interface lo0.0 passive
```

10. Configure LDP on all the interfaces of S-PE1 (ASBR 1), excluding the management interface.

```
[edit protocols]
user@S-PE1# set ldp interface all
user@S-PE1# set ldp interface fxp0.0 disable
user@S-PE1# set ldp interface lo0.0
```

11. Define the policy for enabling next-hop-self on S-PE1 (ASBR 1).

```
[edit policy-options]
user@S-PE1# set policy-statement next-hop-self then next-hop self
```

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

T-PE1

```
user@T-PE1# show interfaces
ge-3/1/0 {
   unit 0 {
      family inet {
         address 11.1.10.1/30;
      }
      family mpls;
   }
}
ge-3/1/2 {
   encapsulation ethernet-ccc;
   unit 0;
}
lo0 {
   unit 0 {
      family inet {
         address 10.255.10.1/32 {
            primary;
         }
      }
   }
}

user@T-PE1# show routing-options
autonomous-system 1;
```

```
user@T-PE1# show protocols
mpls {
  interface all;
  interface fxp0.0 {
    disable;
  }
}
bgp {
  family l2vpn {
    auto-discovery-mspw;
  }
  group mspw {
    type internal;
    local-address 10.255.10.1;
    neighbor 10.255.2.1;
  }
}
ospf {
  area 0.0.0.0 {
    interface all;
    interface fxp0.0 {
      disable;
    }
    interface lo0.0;
  }
}
ldp {
  interface all;
  interface fxp0.0 {
    disable;
  }
  interface lo0.0;
}

user@T-PE1# show routing-instances
ms-pw {
  instance-type l2vpn;
  interface ge-3/1/2.0;
  route-distinguisher 10.10.10.10:15;
  l2vpn-id l2vpn-id:100:15;
  vrf-target target:100:115;
  protocols {
    l2vpn {
      site CE1 {
        source-attachment-identifier 800:800:800;
        interface ge-3/1/2.0 {
          target-attachment-identifier 700:700:700;
        }
      }
      pseudowire-status-tlv;
      oam {
        bfd-liveness-detection {
          minimum-interval 300;
        }
      }
    }
```



```
   }
}
```

S-PE1 (RR 1)

```
user@S-PE1# show interfaces
ge-1/3/1 {
   unit 0 {
      family inet {
         address 30.30.23.9/30;
      }
      family mpls;
   }
}
ge-1/3/2 {
   unit 0 {
      family inet {
         address 20.20.213.9/30;
      }
      family mpls;
   }
}
lo0 {
   unit 0 {
      family inet {
         address 10.255.2.1/32 {
            primary;
         }
      }
   }
}
```

```
user@T-PE1# show routing-options
autonomous-system 1;
```

```
user@S-PE1# show protocols
mpls {
   interface all;
   interface fxp0.0 {
      disable;
   }
}
bgp {
   family l2vpn {
      auto-discovery-mspw;
   }
   group to_T-PE1 {
      type internal;
      local-address 10.255.2.1;
      export next-hop-self;
      neighbor 10.255.10.1;
   }
   group to_S-PE2 {
      type external;
      local-address 10.255.2.1;
      peer-as 2;
      neighbor 10.255.3.1 {
         multihop {
```

```
            ttl 1;
          }
        }
      }
    }
    ospf {
      area 0.0.0.0 {
        interface lo0.0 {
          passive;
        }
        interface all;
        interface fxp0.0 {
          disable;
        }
      }
    }
    ldp {
      interface all;
      interface fxp0.0 {
        disable;
      }
      interface lo0.0;
    }
```

```
user@T-PE1# show policy-options
policy-statement next-hop-self {
  then {
    next-hop self;
  }
}
```

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

Verification

Confirm that the configuration is working properly.

- Verifying the Routes on page 114
- Verifying the LDP Database on page 117
- Checking the MS-PW Connections on T-PE1 on page 118
- Checking the MS-PW Connections on S-PE1 on page 121
- Checking the MS-PW Connections on S-PE2 on page 124
- Checking the MS-PW Connections on T-PE2 on page 127

Verifying the Routes

Purpose Verify that the expected routes are learned.

Action From operational mode, run the **show route** command for the **bgp.l2vpn.1**, **ldp.l2vpn.1**, **mpls.0**, and **ms-pw.l2vpn.1** routing tables.

From operational mode, run the **show route table bgp.l2vpn.1** command.

`user@T-PE1>` **show route table** ***bgp.l2vpn.1***


```
bgp.l2vpn.1: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.10.10:15:700:0.0.2.188:700/160 AD2
                   *[BGP/170] 16:13:11, localpref 100, from 10.255.2.1
                      AS path: 2 I, validation-state: unverified
                    > to 11.1.10.2 via ge-3/1/0.0, Push 300016
```

From operational mode, run the **show route table ldp.l2vpn.1** command.

```
user@T-PE1> show route table ldp.l2vpn.1
ldp.l2vpn.1: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.255.2.1:CtrlWord:5:100:15:700:0.0.2.188:700:800:0.0.3.32:800/304 PW2
                   *[LDP/9] 16:21:27
                      Discard
```

From operational mode, run the **show route table mpls.0** command.

```
user@T-PE1> show route table mpls.0
mpls.0: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 1w6d 00:28:26, metric 1
                      Receive
1                  *[MPLS/0] 1w6d 00:28:26, metric 1
                      Receive
2                  *[MPLS/0] 1w6d 00:28:26, metric 1
                      Receive
13                 *[MPLS/0] 1w6d 00:28:26, metric 1
                      Receive
299920             *[LDP/9] 1w5d 01:26:08, metric 1
                    > to 11.1.10.2 via ge-3/1/0.0, Pop
299920(S=0)        *[LDP/9] 1w5d 01:26:08, metric 1
                    > to 11.1.10.2 via ge-3/1/0.0, Pop
299936             *[LDP/9] 1w5d 01:26:08, metric 1
                    > to 11.1.10.2 via ge-3/1/0.0, Swap 300016
300096             *[LDP/9] 16:22:35, metric 1
                    > to 11.1.10.2 via ge-3/1/0.0, Swap 300128
300112             *[LDP/9] 16:22:35, metric 1
                    > to 11.1.10.2 via ge-3/1/0.0, Swap 300144
300128             *[LDP/9] 16:22:35, metric 1
                    > to 11.1.10.2 via ge-3/1/0.0, Swap 300160
300144             *[L2VPN/7] 16:22:33
                    > via ge-3/1/2.0, Pop       Offset: 4
ge-3/1/2.0         *[L2VPN/7] 16:22:33, metric2 1
                    > to 11.1.10.2 via ge-3/1/0.0, Push 300176, Push 300016(top)
 Offset: 252
```

From operational mode, run the **show route table ms-pw.l2vpn.1** command.

```
user@T-PE1> show route table ms-pw.l2vpn.1
ms-pw.l2vpn.1: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.10.10:15:700:0.0.2.188:700/160 AD2
                   *[BGP/170] 16:23:27, localpref 100, from 10.255.2.1
                      AS path: 2 I, validation-state: unverified
                    > to 11.1.10.2 via ge-3/1/0.0, Push 300016
```

```
10.10.10.10:15:800:0.0.3.32:800/160 AD2
                   *[L2VPN/170] 1w5d 23:25:19, metric2 1
                      Indirect
10.255.2.1:CtrlWord:5:100:15:700:0.0.2.188:700:800:0.0.3.32:800/304 PW2

                   *[LDP/9] 16:23:25
                      Discard
10.255.2.1:CtrlWord:5:100:15:800:0.0.3.32:800:700:0.0.2.188:700/304 PW2

                   *[L2VPN/7] 16:23:27, metric2 1
                    > to 11.1.10.2 via ge-3/1/0.0, Push 300016
```

Meaning The output shows all the learned routes, including the autodiscovery (AD) routes.

The AD2 prefix format is **RD:SAII-type2**, where:

- **RD** is the route distinguisher value.
- **SAII-type2** is the type 2 source attachment identifier value.

The PW2 prefix format is **Neighbor_Addr:C:PWtype:l2vpn-id:SAII-type2:TAII-type2**, where:

- **Neighbor_Addr** is the loopback address of neighboring S-PE device.
- **C** indicates if Control Word (CW) is enabled or not.
  - **C** is **CtrlWord** if CW is set.
  - **C** is **NoCtrlWord** if CW is not set.
- **PWtype** indicates the type of the pseudowire.
  - **PWtype** is **4** if it is in Ethernet tagged mode.
  - **PWtype** is **5** if it is Ethernet only.
- **l2vpn-id** is the Layer 2 VPN ID for the MS-PW routing instance.
- **SAII-type2** is the type 2 source attachment identifier value.
- **TAII-type2** is the type 2 target attachment identifier value.

Verifying the LDP Database

Purpose Verify the MS-PW labels received by T-PE1 from S-PE1 and sent from T-PE1 to S-PE1.

Action From operational mode, run the **show ldp database** command.

```
user@T-PE1> show ldp database
Input label database, 10.255.10.1:0--10.255.2.1:0
  Label     Prefix
      3      10.255.2.1/32
 300112      10.255.3.1/32
 300128      10.255.4.1/32
 299968      10.255.10.1/32
 299904      10.255.13.1/32
 300144      10.255.14.1/32
 300176   FEC129 CtrlWord ETHERNET 000a0064:0000000f 000002bc:000002bc:000002bc
 00000320:00000320:00000320

Output label database, 10.255.10.1:0--10.255.2.1:0
  Label     Prefix
 299936      10.255.2.1/32
 300096      10.255.3.1/32
 300112      10.255.4.1/32
      3      10.255.10.1/32
 299920      10.255.13.1/32
 300128      10.255.14.1/32
 300144    FEC129 CtrlWord ETHERNET 000a0064:0000000f
00000320:00000320:00000320 000002bc:000002bc:000002bc

Input label database, 10.255.10.1:0--10.255.13.1:0
  Label     Prefix
 300016      10.255.2.1/32
 300128      10.255.3.1/32
 300144      10.255.4.1/32
 300080      10.255.10.1/32
      3      10.255.13.1/32
 300160      10.255.14.1/32

Output label database, 10.255.10.1:0--10.255.13.1:0
  Label     Prefix
 299936      10.255.2.1/32
 300096      10.255.3.1/32
 300112      10.255.4.1/32
      3      10.255.10.1/32
 299920      10.255.13.1/32
 300128      10.255.14.1/32
```

Meaning The labels with **FEC129** prefix are related to the MS-PW.

Checking the MS-PW Connections on T-PE1

Purpose Make sure that all of the FEC 129 MS-PW connections come up correctly.

Action From operational mode, run the **show l2vpn connections extensive** command.

```
user@T-PE1> show l2vpn connections extensive
Layer-2 VPN connections:

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

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

Instance: ms-pw
  L2vpn-id: 100:15
    Number of local interfaces: 1
    Number of local interfaces up: 1
    ge-3/1/2.0
  Local source-attachment-id: 800:0.0.3.32:800 (CE1)
    Target-attachment-id      Type  St     Time last up          # Up trans
    700:0.0.2.188:700         rmt   Up     Sep 18 01:10:55 2013           1
      Remote PE: 10.255.2.1, Negotiated control-word: Yes (Null)
      Incoming label: 300048, Outgoing label: 300016
      Negotiated PW status TLV: Yes
      local PW status code: 0x00000000, Neighbor PW status code: 0x00000000
      Local interface: ge-3/1/2.0, Status: Up, Encapsulation: ETHERNET
      Pseudowire Switching Points :
        Local address          Remote address        Status
        10.255.2.1             10.255.3.1            forwarding
        10.255.3.1             10.255.14.1           forwarding
    Connection History:
        Sep 18 01:10:55 2013  status update timer
        Sep 18 01:10:55 2013  PE route changed
        Sep 18 01:10:55 2013  Out lbl Update                    300016
        Sep 18 01:10:55 2013  In lbl Update                     300048
        Sep 18 01:10:55 2013  loc intf up                   ge-3/1/2.0
```

Check the following fields in the output to verify that MS-PW is established between the T-PE devices:

- **Target-attachment-id**—Check if the TAI value is the SAI value of T-PE2.
- **Remote PE**—Check if the T-PE2 loopback address is listed.

- **Negotiated PW status TLV**—Ensure that the value is **Yes**.
- **Pseudowire Switching Points**—Check if the switching points are listed from S-PE1 to S-PE2 and from S-PE2 to T-PE2.

Meaning MS-PW is established between T-PE1 and T-PE2 in the forwarding direction.

Checking the MS-PW Connections on S-PE1

Purpose Make sure that all of the FEC 129 MS-PW connections come up correctly for the mspw routing instance.

Action From operational mode, run the **show l2vpn connections instance __*MSPW*__ extensive** command.

```
user@S-PE1> show l2vpn connections instance __MSPW__ extensive
Layer-2 VPN connections:

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

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

Instance: __MSPW__
  L2vpn-id: 100:15
  Local source-attachment-id: 700:0.0.2.188:700
    Target-attachment-id      Type  St     Time last up          # Up trans
    800:0.0.3.32:800          rmt   Up     Sep 18 01:17:38 2013           1
      Remote PE: 10.255.10.1, Negotiated control-word: Yes (Null), Encapsulation:
 ETHERNET
      Incoming label: 300016, Outgoing label: 300048
      Negotiated PW status TLV: Yes
      local PW status code: 0x00000000, Neighbor PW status code: 0x00000000
  Local source-attachment-id: 800:0.0.3.32:800
    Target-attachment-id      Type  St     Time last up          # Up trans
    700:0.0.2.188:700         rmt   Up     Sep 18 01:17:38 2013           1
      Remote PE: 10.255.3.1, Negotiated control-word: Yes (Null), Encapsulation:
 ETHERNET
      Incoming label: 300000, Outgoing label: 300064
      Negotiated PW status TLV: Yes
      local PW status code: 0x00000000, Neighbor PW status code: 0x00000000
      Pseudowire Switching Points :
        Local address          Remote address        Status
        10.255.3.1             10.255.14.1           forwarding
```

Check the following fields in the output to verify that MS-PW is established between the T-PE devices:

- **Target-attachment-id**—Check if the TAI value is the SAI value of T-PE2.
- **Remote PE**—Check if the T-PE1 and S-PE2 loopback addresses are listed.
- **Negotiated PW status TLV**—Ensure that the value is **Yes**.

- **Pseudowire Switching Points**—Check if the switching points are listed from S-PE2 to T-PE2.

Meaning MS-PW is established between T-PE1 and T-PE2 in the forwarding direction.

Checking the MS-PW Connections on S-PE2

Purpose Make sure that all of the FEC 129 MS-PW connections come up correctly for the mspw routing instance.

Action From operational mode, run the **show l2vpn connections instance __*MSPW*__ extensive** command.

```
user@S-PE2> show l2vpn connections instance __MSPW__ extensive
Layer-2 VPN connections:

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

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

Instance: __MSPW__
  L2vpn-id: 100:15
  Local source-attachment-id: 700:0.0.2.188:700
    Target-attachment-id      Type  St     Time last up          # Up trans
    800:0.0.3.32:800          rmt   Up     Sep 18 00:58:55 2013           1
      Remote PE: 10.255.2.1, Negotiated control-word: Yes (Null), Encapsulation:
 ETHERNET
      Incoming label: 300064, Outgoing label: 300000
      Negotiated PW status TLV: Yes
      local PW status code: 0x00000000, Neighbor PW status code: 0x00000000
      Pseudowire Switching Points :
        Local address          Remote address        Status
        10.255.2.1             10.255.10.1           forwarding
  Local source-attachment-id: 800:0.0.3.32:800
    Target-attachment-id      Type  St     Time last up          # Up trans
    700:0.0.2.188:700         rmt   Up     Sep 18 00:58:55 2013           1
     Remote PE: 10.255.14.1, Negotiated control-word: Yes (Null), Encapsulation:
 ETHERNET
      Incoming label: 300048, Outgoing label: 300112
      Negotiated PW status TLV: Yes
      local PW status code: 0x00000000, Neighbor PW status code: 0x00000000
```

Check the following fields in the output to verify that MS-PW is established between the T-PE devices:

- **Target-attachment-id**—Check if the TAI value is the SAI value of T-PE1.
- **Remote PE**—Check if the S-PE1 and T-PE2 loopback addresses are listed.
- **Negotiated PW status TLV**—Ensure that the value is **Yes**.

- **Pseudowire Switching Points**—Check if the switching points are listed from S-PE1 to T-PE1.

Meaning MS-PW is established between T-PE1 and T-PE2 in the reverse direction.

Checking the MS-PW Connections on T-PE2

Purpose Make sure that all of the FEC 129 MS-PW connections come up correctly.

Action From operational mode, run the **show l2vpn connections extensive** command.

```
user@T-PE2> show l2vpn connections extensive
Layer-2 VPN connections:

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

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

Instance: ms-pw
  L2vpn-id: 100:15
    Number of local interfaces: 1
    Number of local interfaces up: 1
    ge-2/0/0.0
  Local source-attachment-id: 700:0.0.2.188:700 (CE2)
    Target-attachment-id      Type  St     Time last up          # Up trans
    800:0.0.3.32:800          rmt   Up     Sep 18 01:35:21 2013           1
      Remote PE: 10.255.3.1, Negotiated control-word: Yes (Null)
      Incoming label: 300112, Outgoing label: 300048
      Negotiated PW status TLV: Yes
      local PW status code: 0x00000000, Neighbor PW status code: 0x00000000
      Local interface: ge-2/0/0.0, Status: Up, Encapsulation: ETHERNET
      Pseudowire Switching Points :
        Local address          Remote address        Status
        10.255.3.1             10.255.2.1            forwarding
        10.255.2.1             10.255.10.1           forwarding
    Connection History:
        Sep 18 01:35:21 2013  status update timer
        Sep 18 01:35:21 2013  PE route changed
        Sep 18 01:35:21 2013  Out lbl Update                    300048
        Sep 18 01:35:21 2013  In lbl Update                     300112
        Sep 18 01:35:21 2013  loc intf up                   ge-2/0/0.0
```

Check the following fields in the output to verify that MS-PW is established between the T-PE devices:

- **Target-attachment-id**—Check if the TAI value is the SAI value of T-PE1.
- **Remote PE**—Check if the T-PE1 loopback address is listed.

- **Negotiated PW status TLV**—Ensure that the value is **Yes**.
- **Pseudowire Switching Points**—Check if the switching points are listed from S-PE2 to S-PE1 and from S-PE1 to T-PE1.

Meaning MS-PW is established between T-PE1 and T-PE2 in the reverse direction.

Troubleshooting

To troubleshoot the MS-PW connection, see:

- Ping on page 129
- Bidirectional Forwarding Detection on page 129
- Traceroute on page 130

Ping

Problem How to check the connectivity between the T-PE devices and between a T-PE device and an intermediary device.

Solution Verify that T-PE1 can ping T-PE2. The **ping mpls l2vpn fec129** command accepts SAIs and TAIs as integers or IP addresses and also allows you to use the CE-facing interface instead of the other parameters (**instance**, **local-id**, **remote-id**, **remote-pe-address**).

Checking Connectivity Between T-PE1 and T-PE2

```
user@T-PE1> ping mpls l2vpn fec129 instance FEC129-VPWS local-id 800:800:800
remote-pe-address 10.255.14.1 remote-id 700:700:700
!!!!!
--- lsping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
```

```
user@T-PE1> ping mpls l2vpn fec129 interface ge-3/1/2
!!!!!
--- lsping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
```

Checking Connectivity Between T-PE1 and S-PE2

```
user@T-PE1> ping mpls l2vpn fec129 interface ge-3/1/2 bottom-label-ttl 2
!!!!!
--- lsping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
```

Bidirectional Forwarding Detection

Problem How to use BFD to troubleshoot the MS-PW connection from the T-PE device.

Solution From operational mode, verify the **show bfd session extensive** command output.

```
user@T-PE1> show bfd session extensive

                                                  Detect   Transmit
Address                  State     Interface      Time     Interval  Multiplier
127.0.0.1                Up        ge-3/1/0.0     0.900     0.300       3
 Client FEC129-OAM, TX interval 0.300, RX interval 0.300
```

```
 Session up time 03:12:42
 Local diagnostic None, remote diagnostic None
 Remote state Up, version 1
 Replicated
 Session type: VCCV BFD
 Min async interval 0.300, min slow interval 1.000
 Adaptive async TX interval 0.300, RX interval 0.300
 Local min TX interval 0.300, minimum RX interval 0.300, multiplier 3
 Remote min TX interval 0.300, min RX interval 0.300, multiplier 3
 Local discriminator 19, remote discriminator 19
 Echo mode disabled/inactive
 Remote is control-plane independent
 L2vpn-id 100:15, Local-id 800:0.0.3.32:800, Remote-id 700:0.0.2.188:700
  Session ID: 0x103

1 sessions, 1 clients
Cumulative transmit rate 3.3 pps, cumulative receive rate 3.3 pps
```

Traceroute

Problem How to verify that MS-PW was established.

Solution From operational mode, verify **traceroute** output.

```
user@T-PE1> traceroute mpls l2vpn fec129 interface interface
Probe options: ttl 64, retries 3, exp 7

  ttl    Label  Protocol    Address          Previous Hop     Probe Status
    1           FEC129      10.255.10.1         (null)           Success

    2           FEC129      10.255.2.1       10.255.10.1         Success

    3           FEC129      10.255.3.1       10.255.2.1          Success

    4           FEC129      10.255.14.1      10.255.2.1          Egress

    Path 1 via ge-3/1/2 destination 127.0.0.0
```


CHAPTER 5

Layer 2 Circuits Examples

Example: Configuring BFD for VCCV for Layer 2 Circuits

This example shows how to configure BFD for VCCV for Layer 2 circuits which enables faster detection of failure in the data path.

Requirements

This example uses the following hardware and software components:

- Two MX Series 3D Universal Edge Routers
- Junos OS Release 12.1 or later running on all devices

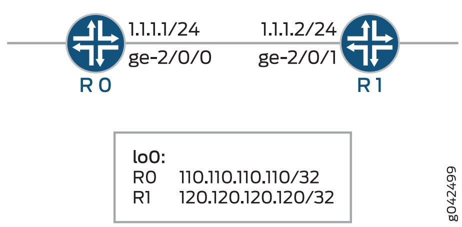
Overview

Starting with Junos OS Release 12.1, Bidirectional Forwarding Detection (BFD) support for virtual circuit connection verification (VCCV) allows you to configure a control channel for a pseudowire, in addition to the corresponding operations and management functions to be used over that control channel. BFD provides a low resource mechanism for the continuous monitoring of the pseudowire data path and for detecting data plane failures. This feature provides support for asynchronous mode BFD for VCCV as described in RFC 5885, *Bidirectional Forwarding Detection (BFD) for the Pseudowire Virtual Circuit Connectivity Verification (VCCV)*. You can also use a ping to detect pseudowire failures. However, the processing resources required for a ping are greater than what is needed for BFD. In addition, BFD is capable of detecting data plane failure faster than VCCV ping. BFD for pseudowires is supported for Layer 2 circuits (LDP-based).

To configure BFD for VCCV for Layer 2 circuits, configure the **oam** configuration statement at the **[edit protocols l2circuit neighbor *address* interface *interface-name*]** hierarchy level. The **control-channel** configuration statement at the **[edit routing-instances *routing-instance-name* protocols l2vpn oam]** hierarchy level does not apply to Layer 2 circuit configurations.

Topology

In the topology, BFD for VCCV for Layer 2 circuits is configured on Device R0.



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, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

R0

```
set chassis redundancy graceful-switchover
set interfaces ge-1/1/9 vlan-tagging
set interfaces ge-1/1/9 encapsulation vlan-ccc
set interfaces ge-1/1/9 unit 0 encapsulation vlan-ccc
set interfaces ge-1/1/9 unit 0 vlan-id 512
set interfaces ge-2/0/0 unit 0 family inet address 1.1.1.1/24
set interfaces ge-2/0/0 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 110.110.110.110/32
set routing-options nonstop-routing
set routing-options static route 120.120.120.120/32 next-hop 1.1.1.2
set routing-options router-id 110.110.110.110
set protocols rsvp interface ge-2/0/0.0
```

```
set protocols mpls label-switched-path lsp1 to 120.120.120.120
set protocols mpls interface ge-2/0/0.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-2/0/0.0
set protocols ldp interface all
set protocols l2circuit neighbor 120.120.120.120 interface ge-1/1/9.0 virtual-circuit-id 1
set protocols l2circuit neighbor 120.120.120.120 interface ge-1/1/9.0 oam
  bfd-liveness-detection minimum-interval 300
set protocols l2circuit neighbor 120.120.120.120 interface ge-1/1/9.0 oam
  bfd-liveness-detection minimum-receive-interval 10
set protocols l2circuit neighbor 120.120.120.120 interface ge-1/1/9.0 oam
  bfd-liveness-detection multiplier 3
set protocols l2circuit neighbor 120.120.120.120 interface ge-1/1/9.0 oam
  bfd-liveness-detection transmit-interval minimum-interval 5
set protocols l2circuit neighbor 120.120.120.120 interface ge-1/1/9.0 oam
  bfd-liveness-detection transmit-interval threshold 30
set protocols l2circuit neighbor 120.120.120.120 interface ge-1/1/9.0 oam
  bfd-liveness-detection detection-time threshold 40
```

R1

```
set interfaces ge-1/1/9 vlan-tagging
set interfaces ge-1/1/9 encapsulation vlan-ccc
set interfaces ge-1/1/9 unit 0 encapsulation vlan-ccc
set interfaces ge-1/1/9 unit 0 vlan-id 512
set interfaces ge-2/0/1 unit 0 family inet address 1.1.1.2/24
set interfaces ge-2/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 120.120.120.120/32
set routing-options static route 110.110.110.110/32 next-hop 1.1.1.1
set routing-options router-id 120.120.120.120
set protocols rsvp interface ge-2/0/1.0
set protocols mpls label-switched-path lsp2 to 110.110.110.110
set protocols mpls interface ge-2/0/1.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-2/0/1.0
set protocols ldp interface all
set protocols l2circuit neighbor 110.110.110.110 interface ge-1/1/9.0 virtual-circuit-id 1
set protocols l2circuit neighbor 110.110.110.110 interface ge-1/1/9.0 oam
  bfd-liveness-detection minimum-interval 300
set protocols l2circuit neighbor 110.110.110.110 interface ge-1/1/9.0 oam
  bfd-liveness-detection minimum-receive-interval 10
set protocols l2circuit neighbor 110.110.110.110 interface ge-1/1/9.0 oam
  bfd-liveness-detection multiplier 3
set protocols l2circuit neighbor 110.110.110.110 interface ge-1/1/9.0 oam
  bfd-liveness-detection transmit-interval minimum-interval 5
set protocols l2circuit neighbor 110.110.110.110 interface ge-1/1/9.0 oam
  bfd-liveness-detection transmit-interval threshold 30
set protocols l2circuit neighbor 110.110.110.110 interface ge-1/1/9.0 oam
  bfd-liveness-detection detection-time threshold 40
```

Configuring Device R0

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 Device R0:

NOTE: Repeat this procedure for Device R1 after modifying the appropriate interface names, addresses, and any other parameters for the device.

1. Configure graceful switchover redundancy.

   ```
   [edit chassis]
   user@R0# set redundancy graceful-switchover
   ```

2. Configure the interfaces.

   ```
   [edit interfaces]
   user@R0# set ge-1/1/9 vlan-tagging
   user@R0# set ge-1/1/9 encapsulation vlan-ccc
   user@R0# set ge-1/1/9 unit 0 encapsulation vlan-ccc
   user@R0# set ge-1/1/9 unit 0 vlan-id 512

   user@R0# set ge-2/0/0 unit 0 family inet address 1.1.1.1/24
   user@R0# set ge-2/0/0 unit 0 family mpls

   user@R0# set lo0 unit 0 family inet address 110.110.110.110/32
   ```

3. Configure the nonstop routing option, the static route, and the router ID routing options.

   ```
   [edit routing-options]
   user@R0# set nonstop-routing
   user@R0# set static route 120.120.120.120/32 next-hop 1.1.1.2
   user@R0# set router-id 110.110.110.110
   ```

4. Configure the RSVP protocol.

   ```
   [edit protocols rsvp]
   user@R0# set interface ge-2/0/0.0
   ```

5. Configure the MPLS protocol.

   ```
   [edit protocols mpls]
   user@R0# set label-switched-path lsp1 to 120.120.120.120
   user@R0# set interface ge-2/0/0.0
   ```

6. Configure the OSPF protocol.

   ```
   [edit protocols ospf]
   user@R0# set traffic-engineering
   user@R0# set area 0.0.0.0 interface ge-2/0/0.0
   ```

7. Configure the LDP protocol.

```
[edit protocols ldp]
user@R0# set interface all
```

8. Configure the virtual circuit ID for the neighbor of Layer 2 circuit protocols.

```
[edit protocols l2circuit]
user@R0# set neighbor 120.120.120.120 interface ge-1/1/9.0 virtual-circuit-id 1
```

9. Configure the oam attributes of the Layer 2 circuit protocol.

```
[edit protocols l2circuit]
user@R0# set neighbor 120.120.120.120 interface ge-1/1/9.0 oam
  bfd-liveness-detection minimum-interval 300
user@R0# set neighbor 120.120.120.120 interface ge-1/1/9.0 oam
  bfd-liveness-detection minimum-receive-interval 10
user@R0# set neighbor 120.120.120.120 interface ge-1/1/9.0 oam
  bfd-liveness-detection multiplier 3
user@R0# set neighbor 120.120.120.120 interface ge-1/1/9.0 oam
  bfd-liveness-detection transmit-interval minimum-interval 5
user@R0# set neighbor 120.120.120.120 interface ge-1/1/9.0 oam
  bfd-liveness-detection transmit-interval threshold 30
user@R0# set neighbor 120.120.120.120 interface ge-1/1/9.0 oam
  bfd-liveness-detection detection-time threshold 40
```

Results

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

```
user@R0# show chassis
redundancy {
  graceful-switchover;
}
```

```
user@R0# show interfaces
ge-1/1/9 {
  vlan-tagging;
  encapsulation vlan-ccc;
  unit 0 {
    encapsulation vlan-ccc;
    vlan-id 512;
  }
}
ge-2/0/0 {
  unit 0 {
    family inet {
      address 1.1.1.1/24;
    }
    family mpls;
  }
}
lo0 {
  unit 0 {
    family inet {
      address 110.110.110.110/32;
```

```
        }
    }
}
user@R0# show protocols
rsvp {
    interface ge-2/0/0.0;
}
mpls {
    label-switched-path lsp1 {
        to 120.120.120.120;
    }
    interface ge-2/0/0.0;
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface ge-2/0/0.0;
    }
}
ldp {
    interface all;
}
l2circuit {
    neighbor 120.120.120.120 {
        interface ge-1/1/9.0 {
            virtual-circuit-id 1;
            oam {
                bfd-liveness-detection {
                    minimum-interval 300;
                    minimum-receive-interval 10;
                    multiplier 3;
                    transmit-interval {
                        minimum-interval 5;
                        threshold 30;
                    }
                    detection-time {
                        threshold 40;
                    }
                }
            }
        }
    }
}
user@R0# show routing-options
nonstop-routing;
static {
    route 120.120.120.120/32 next-hop 1.1.1.2;
}
router-id 110.110.110.110;
```

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

Verification

Verify that the configuration is working properly.

- Verifying the Layer 2 Circuit Connections on page 137
- Verifying the BFD Session on page 137
- Verifying Detailed BFD Session Information on page 138

Verifying the Layer 2 Circuit Connections

Purpose Verify the connections in a Layer 2 Circuit.

Action From operational mode, run the **show l2circuit connections** command for Device R0.

```
user@R0> show l2circuit connections

Layer-2 Circuit Connections:

Legend for connection status (St)
EI -- encapsulation invalid      NP -- interface h/w not present
MM -- mtu mismatch               Dn -- down
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
VM -- vlan id mismatch           CF -- Call admission control failure
OL -- no outgoing label          IB -- TDM incompatible bitrate
NC -- intf encaps not CCC/TCC    TM -- TDM misconfiguration
BK -- Backup Connection          ST -- Standby Connection
CB -- rcvd cell-bundle size bad  SP -- Static Pseudowire
LD -- local site signaled down   RS -- remote site standby
RD -- remote site signaled down  HS -- Hot-standby Connection
XX -- unknown

Legend for interface status
Up -- operational
Dn -- down
Neighbor: 120.120.120.120
    Interface                 Type  St     Time last up          # Up trans
    ge-1/1/9.0(vc 1)          rmt   Up     Jun  2 03:19:44 2014           1
      Remote PE: 120.120.120.120, Negotiated control-word: Yes (Null)
      Incoming label: 299792, Outgoing label: 299792
      Negotiated PW status TLV: No
      Local interface: ge-1/1/9.0, Status: Up, Encapsulation: VLAN
      Flow Label Transmit: No, Flow Label Receive: No
      Flow Label Transmit: No, Flow Label Receive: No
```

Meaning The output shows the Layer 2 virtual circuit information from Device R0 to its neighbor.

Verifying the BFD Session

Purpose Verify the BFD session.

Action From operational mode, run the **show bfd session** command for Device R0.

```
user@R0> show bfd session

                                                  Detect   Transmit
Address                  State     Interface      Time     Interval  Multiplier

127.0.0.1                Up        ge-2/0/0.0     0.030     0.010       3

1 sessions, 1 clients
Cumulative transmit rate 100.0 pps, cumulative receive rate 100.0 pps
```

Meaning The output shows the address, and the interface on which the BFD session is active. The state *Up* indicates that the BFD session is up. The BFD session has a time interval of 30 milliseconds to detect BFD control packets , the transmitting system has a time interval of 10 milliseconds to send BFD control packets, and the transmitting system determines the detection time by multiplying *3* with the time interval. Total number of active BFD sessions and total number of clients that are hosting active BFD sessions. Cumulative transmit rate indicates the total number of BFD control packets transmitted, per second, on all active sessions and cumulative receive rate indicates the total number of BFD control packets received, per second, on all active sessions.

Verifying Detailed BFD Session Information

Purpose Verify detailed BFD session information.

Action From operational mode, run the **show bfd session extensive** command for Device R0.

```
user@R0> show bfd session extensive

                                                      Detect   Transmit
Address                  State     Interface      Time     Interval  Multiplier

127.0.0.1                Up        ge-2/0/0.0     0.030     0.010       3
 Client L2CKT-OAM, TX interval 0.005, RX interval 0.010
 Session up time 03:47:14
 Local diagnostic None, remote diagnostic None
 Remote state Up, version 1
 Replicated
 Session type: VCCV BFD
 Min async interval 0.005, min slow interval 1.000
 Adaptive async TX interval 0.005, RX interval 0.010
 Local min TX interval 0.005, minimum RX interval 0.010, multiplier 3
 Remote min TX interval 0.005, min RX interval 0.010, multiplier 3
 Threshold transmission interval 0.030, Threshold for detection time 0.040
 Local discriminator 20, remote discriminator 13004
 Echo mode disabled/inactive
 Remote is control-plane independent
 Neighbor address 120.120.120.120, Virtual circuit id 1
  Session ID: 0x0

1 sessions, 1 clients
Cumulative transmit rate 100.0 pps, cumulative receive rate 100.0 pps
```

Meaning The output shows detailed information for the BFD session.

Related Documentation

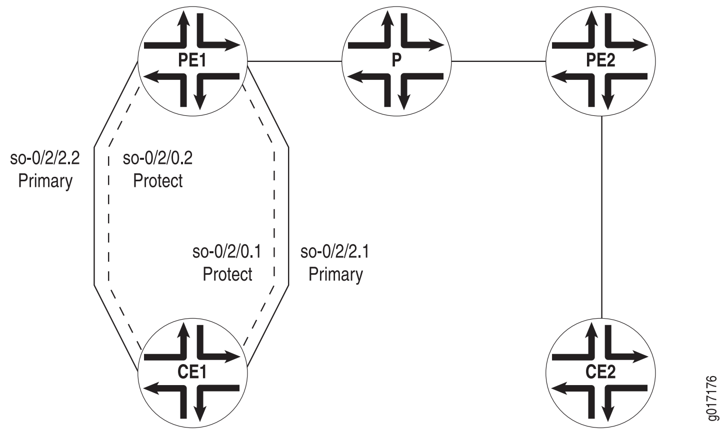
- Configuring BFD for VCCV for Layer 2 Circuits on page 26

Example: Configuring Layer 2 Circuit Protect Interfaces

This example illustrates how you might configure a Layer 2 circuit with protect interfaces. Protect interfaces act as backups for their associated interfaces. The primary interface has priority over the protect interface and carries network traffic as long as it is functional. If the primary interface fails, the protect interface is activated. These interfaces can also share the same virtual path identifier (VPI) or virtual circuit identifier (VCI).

Figure 12 on page 140 shows the network topology used in this example.

Figure 12: Layer 2 Circuits Using Protect Interfaces



The following sections describe how to configure a Layer 2 circuit to use a protect interface:

- Configuring Router PE1 on page 140
- Configuring Router PE2 on page 142
- Configuring Router CE1 on page 143
- Configuring Router CE2 on page 144

Configuring Router PE1

Configure an interface for traffic to Router CE1 from Router PE1 at the **[edit interfaces]** hierarchy level:

```
[edit interfaces]
so-0/2/2 {
   description "Router CE1 so-0/2/2";
   no-keepalives;
   encapsulation frame-relay-ccc;
   unit 1 {
      encapsulation frame-relay-ccc;
      point-to-point;
      dlci 600;
   }
   unit 2 {
      encapsulation frame-relay-ccc;
      point-to-point;
      dlci 602;
   }
}
```

Configure an interface for traffic to Router CE1 from Router PE1 at the **[edit interfaces]** hierarchy level. Logical interface **so-0/2/0.2** acts as the protect interface for **so-0/2/2.2**, and logical interface **so-0/2/0.1** acts as the protect interface for **so-0/2/2.1**:

```
[edit interfaces]
so-0/2/0 {
   description "to Router CE1 so-0/3/0";
```



```
    no-keepalives;
    encapsulation frame-relay-ccc;
    unit 1 {
      encapsulation frame-relay-ccc;
      dlci 600;
    }
    unit 2 {
      encapsulation frame-relay-ccc;
      dlci 602;
    }
  }
```

Configure an interface for traffic to Router PE2 from Router PE1 at the **[edit interfaces]** hierarchy level:

```
[edit interfaces]
so-0/2/1 {
  description "to Router PE2 so-1/0/1";
  unit 0 {
    family inet {
      address 100.100.40.22/32 {
        destination 100.100.40.23;
      }
    }
    family iso;
    family mpls;
  }
}
```

Configure an interface for traffic to Router PE2 from Router PE1 at the **[edit interfaces]** hierarchy level:

```
[edit interfaces]
so-0/2/3 {
  description "Router PE2 so-1/0/3";
  unit 0 {
    family inet;
    family iso;
    family mpls;
  }
  lo0 {
    unit 0 {
      family inet {
        address 127.0.0.1/32;
        address 10.100.40.200/32;
      }
      family iso {
        address 47.0005.80ff.f800.0000.0108.0001.1921.6800.4213.00;
      }
    }
  }
}
```

Configure the Layer 2 circuit by including the **l2circuit** statement at the **[edit protocols]** hierarchy level. The logical interfaces for the Layer 2 circuits and their corresponding protect interfaces are included here:

```
[edit protocols]
l2circuit {
   neighbor 10.100.40.210 {
      interface so-0/2/2.2 {
         protect-interface so-0/2/0.2;
         virtual-circuit-id 2;
         no-control-word;
      }
      interface so-0/2/2.1 {
         protect-interface so-0/2/0.1;
         virtual-circuit-id 1;
         no-control-word;
      }
   }
}
```

Configuring Router PE2

Configure an interface for traffic to Router CE2 from Router PE2:

```
[edit interfaces]
so-1/0/0 {
   description "to Router CE2 so-0/2/0";
   no-keepalives;
   encapsulation frame-relay-ccc;
   unit 1 {
      encapsulation frame-relay-ccc;
      point-to-point;
      dlci 700;
   }
   unit 2 {
      encapsulation frame-relay-ccc;
      point-to-point;
      dlci 702;
   }
}
```

Configure an interface for traffic to Router PE1 from Router PE2:

```
[edit interfaces]
so-1/0/1 {
   description "to Router PE1 so-0/2/1";
   unit 0 {
      family inet {
         address 100.100.40.23/32 {
            destination 100.100.40.22;
         }
      }
      family iso;
      family mpls;
   }
}
```

Configure an interface for traffic to Router PE1 from Router PE2:

```
[edit interfaces]
so-1/0/3 {
```



```
    description "to Router PE1 so-0/2/3";
    unit 0 {
      family inet;
      family iso;
      family mpls;
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 127.0.0.1/32;
        address 10.100.40.210/32;
      }
      family iso {
        address 47.0005.80ff.f800.0000.0108.0001.1921.6800.4216.00;
      }
    }
  }
```

Configure the Layer 2 circuit at the **[edit protocols]** hierarchy level:

```
[edit protocols]
l2circuit {
  neighbor 10.100.40.200 {
    interface so-1/0/0.1 {
      virtual-circuit-id 1;
      no-control-word;
    }
    interface so-1/0/0.2 {
      virtual-circuit-id 2;
      no-control-word;
    }
  }
}
```

Configuring Router CE1

Configure an interface for traffic to Router PE1 from Router CE1:

```
[edit interfaces]
so-0/3/0 {
  description "to Router PE1 so-0/2/0";
  no-keepalives;
  encapsulation frame-relay;
  unit 1 {
    dlci 601;
    family inet {
      address 12.12.12.1/24;
    }
  }
}
```

Configure an interface for traffic to Router PE1 from Router CE1:

```
[edit interfaces]
so-0/3/1 {
  description "Router PE1 so-0/2/2";
```



```
    no-keepalives;
    encapsulation frame-relay;
    unit 0 {
      dlci 600;
      family inet {
        address 10.10.10.1/24;
        address 11.1.1.1/24;
      }
      family iso;
      family mpls;
    }
    unit 2 {
      dlci 602;
      family inet {
        address 13.13.13.1/24;
      }
    }
  }
```

Configuring Router CE2

Configure an interface for traffic to Router PE2 from Router CE2:

```
[edit interfaces]
so-0/2/0 {
  description "to Router PE2 so-1/0/0";
  no-keepalives;
  encapsulation frame-relay;
  unit 1 {
    dlci 700;
    family inet {
      address 10.10.10.2/24;
      address 11.1.1.2/24;
      address 12.12.12.2/24;
    }
  }
  unit 2 {
    dlci 702;
    family inet {
      address 13.13.13.2/24;
    }
  }
}
```

Example: Configuring Layer 2 Circuit Switching Protection

Unlike Layer 2 circuit protect interfaces (see “Example: Configuring Layer 2 Circuit Protect Interfaces” on page 139), which provide traffic protection for paths configured between the PE routers and CE routers, Layer 2 circuit switching protection provides traffic protection for the paths configured between the PE routers. In the event the path used by a Layer 2 circuit fails, traffic can be switched to an alternate path (or protection path). Switching protection is supported for locally switched Layer 2 circuits and provides 1 to 1 protection for each Layer 2 circuit interface.

When you enable Layer 2 circuit switching protection, each Layer 2 circuit interface requires the following paths:

- Working path—Used by the Layer 2 circuit when working normally.
- Protection path—Used by the Layer 2 circuit when the working path fails.

Requirements

This example uses the following hardware and software components:

- MX Series 3D Universal Edge Routers
- Junos OS Release 12.3

Overview

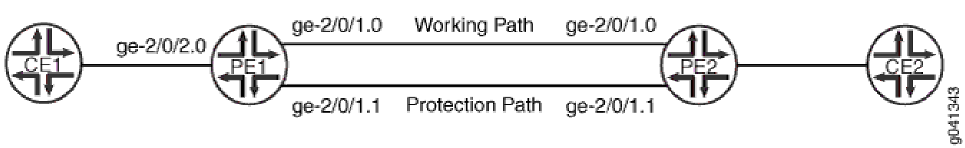
Each working path can be configured to have a either a protection path routed directly to the neighboring PE router (as shown in Figure 13 on page 145) or indirectly using a pseudowire configured through an intermediate PE router (as shown in Figure 14 on page 146 and Figure 15 on page 146). The protection path provides failure protection for the traffic flowing between the PE routers. Ethernet OAM monitors the status of these paths. When OAM detects a failure, it reroutes the traffic from the failed working path to the protection path. You can configure OAM to revert the traffic automatically to the working path when it is restored. You can also manually switch traffic between the working path, the protection path, and back.

NOTE: **Non-stop routing (NSR) and graceful routing engine switchover (GRES) do not support Layer 2 circuit switching protection.**

Topology

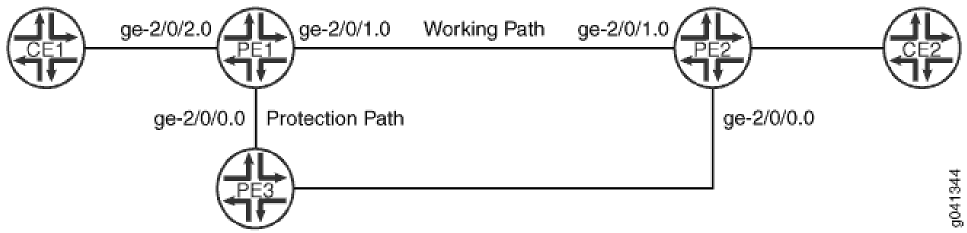
Figure 13 on page 145 illustrates Layer 2 circuit local switching. There are two OAM sessions running between router PE1 and router PE2. One OAM session is configured over the working path and the other is configured over the protection path.

Figure 13: Connection Protection Enabled Between Router PE1 and Router PE2



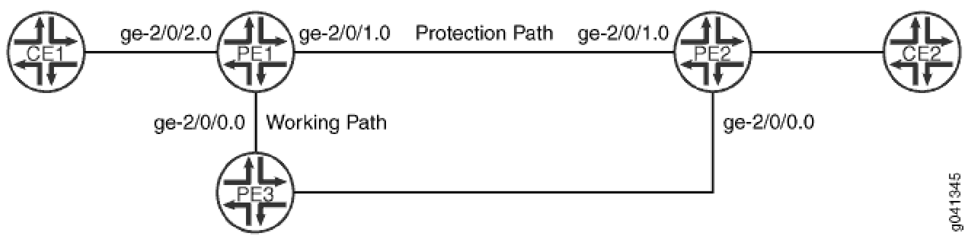
In Figure 14 on page 146 and Figure 15 on page 146, there are two OAM sessions running between router PE1 and router PE2. For Figure 2, one OAM session is configured over the working path between router PE1 and router PE2. The other OAM session is configured over the protection path between router PE1 and router PE3 to router PE2.

Figure 14: Connection Protection Using a Pseudowire Configured through Router PE3 as the Protection Path



For Figure 15 on page 146, one OAM session is configured over the working path, the pseudowire between router PE1 and router PE3, then to router PE2. The other OAM session is configured on the protect path between router PE1 and router PE2.

Figure 15: Connection Protection Using a Pseudowire Configured through Router PE3 as the Working Path



Configuration

The following sections describe how to configure each of the variations of Layer 2 circuit connection protection:

- Configuring Connection Protection Between Two PE Routers on page 146
- Configuring Connection Protection Using Another PE Router for the Protection Path on page 150
- Configuring Connection Protection Using an Another PE Router for the Working Path on page 153

Configuring Connection Protection Between Two PE Routers

Step-by-Step Procedure

To configure Layer 2 Circuit switching protection as shown in Figure 13 on page 145 on router PE1:

1. Configure the Layer 2 circuit on router PE1.

```
[edit protocols l2circuit]
set local-switching interface ge-2/0/2.0 connection-protection
set local-switching interface ge-2/0/2.0 end-interface interface ge-2/0/1.0
set local-switching interface ge-2/0/2.0 end-interface backup-interface ge-2/0/1.1
```

2. Configure the routing policy on router PE1.

```
[edit policy-options]
```

```
set policy-statement protection-policy then load-balance per-packet
```

3. Enable the routing policy on router PE1.

```
[edit routing-options]
set routing-options forwarding-table export protection-policy
```

4. Configure OAM on Router PE1. OAM is used to monitor the working path between router PE1 and router PE2. In the event of a failure on the working path, traffic is switched automatically to the protection path. A connectivity fault management (CFM) session is configured on the working path and on the protection path. Begin by configuring the OAM maintenance domain.

```
[edit protocols oam ethernet]
set protocols oam ethernet connectivity-fault-management maintenance-domain
  l2circuit-example-md level 5
```

5. Configure OAM on Router PE1 for the working path.

```
[edit protocols oam ethernet]
set protocols oam ethernet connectivity-fault-management maintenance-domain
  l2circuit-example-md maintenance-association working continuity-check interval
  100ms
set protocols oam ethernet connectivity-fault-management maintenance-domain
  l2circuit-example-md maintenance-association working mep 1000 interface
  ge-2/0/1.0
set protocols oam ethernet connectivity-fault-management maintenance-domain
  l2circuit-example-md maintenance-association working mep 1000 interface
  working
set protocols oam ethernet connectivity-fault-management maintenance-domain
  l2circuit-example-md maintenance-association working mep 1000 direction down
set protocols oam ethernet connectivity-fault-management maintenance-domain
  l2circuit-example-md maintenance-association working mep 1000 remote-mep
  103
```

6. Configure OAM on Router PE1 for the protection path.

```
[edit protocols oam ethernet]
set protocols oam ethernet connectivity-fault-management maintenance-domain
  l2circuit-example-md maintenance-association protection continuity-check
  interval 100ms
set protocols oam ethernet connectivity-fault-management maintenance-domain
  l2circuit-example-md maintenance-association protection mep 1001 interface
  ge-2/0/1.1
set protocols oam ethernet connectivity-fault-management maintenance-domain
  l2circuit-example-md maintenance-association protection mep 1001 interface
  protect
set protocols oam ethernet connectivity-fault-management maintenance-domain
  l2circuit-example-md maintenance-association protection mep 1001 direction
  down
set protocols oam ethernet connectivity-fault-management maintenance-domain
  l2circuit-example-md maintenance-association protection mep 1001 remote-mep
  104
```

7. Configure the OAM maintenance domain on Router PE2.

```
set protocols oam ethernet connectivity-fault-management maintenance-domain
  l2circuit-example-md level 5
```


8. Configure OAM on Router PE2 for the working path.

```
set protocols oam ethernet connectivity-fault-management maintenance-domain
  l2circuit-example-md maintenance-association working continuity-check interval
  100ms
set protocols oam ethernet connectivity-fault-management maintenance-domain
  l2circuit-example-md maintenance-association working mep 103 interface
  ge-2/0/1.0
set protocols oam ethernet connectivity-fault-management maintenance-domain
  l2circuit-example-md maintenance-association working mep 103 interface working
set protocols oam ethernet connectivity-fault-management maintenance-domain
  l2circuit-example-md maintenance-association working mep 103 direction down
set protocols oam ethernet connectivity-fault-management maintenance-domain
  l2circuit-example-md maintenance-association working mep 103 remote-mep
  1000
```

9. Configure OAM on Router PE2 for the protection path.

```
set protocols oam ethernet connectivity-fault-management maintenance-domain
  l2circuit-example-md maintenance-association protection continuity-check
  interval 100ms
set protocols oam ethernet connectivity-fault-management maintenance-domain
  l2circuit-example-md maintenance-association protection mep 104 interface
  ge-2/0/1.1
set protocols oam ethernet connectivity-fault-management maintenance-domain
  l2circuit-example-md maintenance-association protection mep 104 interface
  protect
set protocols oam ethernet connectivity-fault-management maintenance-domain
  l2circuit-example-md maintenance-association protection mep 104 direction
  down
set protocols oam ethernet connectivity-fault-management maintenance-domain
  l2circuit-example-md maintenance-association protection mep 104 remote-mep
  1001
```

Results From configuration mode on Router PE1, confirm your configuration by entering the **show protocols l2circuit**, **show policy-options**, **show routing-options**, and **show protocols oam ethernet** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host> show protocols l2circuit
local-switching {
  interface ge-2/0/2.0 {
    connection-protection;
    end-interface {
      interface ge-2/0/1.0;
      backup-interface ge-2/0/1.1;
    }
  }
}

user@host> show policy-options
policy-statement protection-policy {
  then {
    load-balance per-packet;
  }
}
```



```
user@host> show routing-options
forwarding-table {
  export protection-policy;
}
```

```
user@host> show protocols oam ethernet
connectivity-fault-management {
  maintenance-domain l2circuit-example-md {
    level 5;
    maintenance-association working {
      continuity-check {
        interval 100ms;
      }
      mep 1000 {
        interface ge-2/0/1.0 working;
        direction down;
        remote-mep 103;
      }
    }
    maintenance-association protection {
      continuity-check {
        interval 100ms;
      }
      mep 1001 {
        interface ge-2/0/1.1 protect;
        direction down;
        remote-mep 104;
      }
    }
  }
}
```

From configuration mode on Router PE2, confirm your configuration by entering the **show protocols oam ethernet** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
connectivity-fault-management {
  maintenance-domain l2circuit-example-md {
    level 5;
    maintenance-association working {
      continuity-check {
        interval 100ms;
      }
      mep 103 {
        interface ge-2/0/1.0 working;
        direction down;
        remote-mep 1000;
      }
    }
    maintenance-association protection {
      continuity-check {
        interval 100ms;
      }
      mep 104 {
        interface ge-2/0/1.1 protect;
        direction down;
```

```
            remote-mep 1001;
          }
        }
      }
    }
```

Configuring Connection Protection Using Another PE Router for the Protection Path

Step-by-Step Procedure

To configure Layer 2 Circuit switching protection as shown in Figure 14 on page 146 on router PE1:

1. Configure the Layer 2 circuit on router PE1.

   ```
   [edit protocols l2circuit]
   set protocols l2circuit local-switching interface ge-2/0/2.0 connection-protection
   set protocols l2circuit local-switching interface ge-2/0/2.0 backup-neighbor 2.2.2.2
     virtual-circuit-id 2
   set protocols l2circuit local-switching interface ge-2/0/2.0 backup-neighbor 2.2.2.2
     community example
   set protocols l2circuit local-switching interface ge-2/0/2.0 end-interface interface
     ge-2/0/1.0
   ```

2. Configure the routing policy on router PE1.

   ```
   [edit policy-options]
   set policy-statement load-balance then load-balance per-packet
   set policy-statement protection-policy term protect from community example
   set policy-statement protection-policy term protect then install-nexthop lsp-regex
     lsp-protect-*
   ```

3. Configure the routing options on router PE1.

   ```
   [edit routing-options]
   set routing-options forwarding-table export load-balance
   ```

4. Configure OAM on Router PE1 to setup the maintenance domain. OAM is used to monitor the working path between router PE1 and router PE2. In the event of a failure on the working path, traffic is switched automatically to the protection path.

   ```
   [edit protocols oam ethernet]
   set connectivity-fault-management maintenance-domain l2circuit-example-md
     level 5
   ```

5. Configure OAM on Router PE1 for the working path.

   ```
   [edit protocols oam ethernet]
   set connectivity-fault-management maintenance-domain l2circuit-example-md
     maintenance-association working mep 1000 interface ge-2/0/1.0
   set connectivity-fault-management maintenance-domain l2circuit-example-md
     maintenance-association working mep 1000 direction down
   set connectivity-fault-management maintenance-domain l2circuit-example-md
     maintenance-association working mep 1000 remote-mep 103
   ```

6. Configure OAM on Router PE1 for the protection path.

   ```
   [edit protocols oam ethernet]
   set connectivity-fault-management maintenance-domain l2circuit-example-md
     maintenance-association protection mep 1001 interface ge-2/0/0.0
   ```



```
set connectivity-fault-management maintenance-domain l2circuit-example-md
   maintenance-association protection mep 1001 direction down
set connectivity-fault-management maintenance-domain l2circuit-example-md
   maintenance-association protection mep 1001 remote-mep 104
```

7. Configure OAM on Router PE2 to setup the maintenance domain.

```
[edit protocols oam ethernet]
set connectivity-fault-management maintenance-domain l2circuit-example-md
   level 5
```

8. Configure OAM on Router PE2 for the working path.

```
[edit protocols oam ethernet]
set connectivity-fault-management maintenance-domain l2circuit-example-md
   maintenance-association working mep 103 interface ge-2/0/1.0
set connectivity-fault-management maintenance-domain l2circuit-example-md
   maintenance-association working mep 103 direction down
set connectivity-fault-management maintenance-domain l2circuit-example-md
   maintenance-association working mep 103 remote-mep 1000
```

9. Configure OAM on Router PE2 for the protection path.

```
[edit protocols oam ethernet]
set connectivity-fault-management maintenance-domain l2circuit-example-md
   maintenance-association protection mep 104 interface ge-2/0/0.0
set connectivity-fault-management maintenance-domain l2circuit-example-md
   maintenance-association protection mep 104 direction down
set connectivity-fault-management maintenance-domain l2circuit-example-md
   maintenance-association protection mep 104 remote-mep 1001
```

Results From configuration mode on Router PE1, confirm your configuration by entering the **show protocols l2circuit**, **show policy-options**, **show routing-options**, and **show protocols oam ethernet** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host> show protocols l2circuit
local-switching {
   interface ge-2/0/2.0 {
      connection-protection;
      backup-neighbor 2.2.2.2 {
         virtual-circuit-id 2;
         community example;
      }
      end-interface {
         interface ge-2/0/1.0;
      }
   }
}
```

```
user@host> show policy-options
policy-statement load-balance {
   then {
      load-balance per-packet;
   }
}
policy-statement protection-policy {
   term protect {
```

```
      from community example;
      then {
        install-nexthop lsp-regex lsp-protect-*;
      }
    }
  }
```

```
user@host> show routing-options
forwarding-table {
  export load-balance;
}
```

```
user@host> show protocols oam ethernet
connectivity-fault-management {
  maintenance-domain l2circuit-example-md {
    level 5;
    maintenance-association working {
      mep 1000 {
        interface ge-2/0/1.0;
        direction down;
        remote-mep 103;
      }
    }
    maintenance-association protection {
      mep 1001 {
        interface ge-2/0/0.0;
        direction down;
        remote-mep 104;
      }
    }
  }
}
```

From configuration mode on Router PE2, confirm your configuration by entering the **show protocols oam ethernet** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
connectivity-fault-management {
  maintenance-domain l2circuit-example-md {
    level 5;
    maintenance-association working {
      mep 103 {
        interface ge-2/0/1.0;
        direction down;
        remote-mep 1000;
      }
    }
    maintenance-association protection {
      mep 104 {
        interface ge-2/0/0.0;
        direction down;
        remote-mep 1001;
      }
    }
  }
}
```


Configuring Connection Protection Using an Another PE Router for the Working Path

Step-by-Step Procedure

To configure Layer 2 Circuit switching protection as shown in Figure 15 on page 146 on router PE1:

1. Configure the Layer 2 circuit on router PE1.

   [edit protocols l2circuit]
   set protocols l2circuit neighbor 2.2.2.2 interface ge-2/0/2.0 virtual-circuit-id 2
   set protocols l2circuit neighbor 2.2.2.2 interface ge-2/0/2.0 community example
   set protocols l2circuit neighbor 2.2.2.2 interface ge-2/0/2.0 connection-protection
   set protocols l2circuit neighbor 2.2.2.2 interface ge-2/0/2.0 backup-neighbor 3.3.3.3 virtual-circuit-id 3
   set protocols l2circuit neighbor 2.2.2.2 interface ge-2/0/2.0 backup-neighbor 3.3.3.3 standby

2. Configure the policies on router PE1.

   [edit policy-options]
   set policy-options policy-statement load-balance then load-balance per-packet
   set policy-options policy-statement protection-policy term protect from community example
   set policy-options policy-statement protection-policy term protect then install-nexthop lsp-regex lsp-primary

3. [edit routing-options]
   set routing-options forwarding-table export load-balance

4. Configure OAM on Router PE1 to setup the maintenance domain. OAM is used to monitor the working path between router PE1 and router PE2. In the event of a failure on the working path, traffic is switched automatically to the protection path.

   [edit protocols oam ethernet]
   set connectivity-fault-management maintenance-domain l2circuit-example-md level 5

5. Configure OAM on Router PE1 for the working path.

   [edit protocols oam ethernet]
   set connectivity-fault-management maintenance-domain l2circuit-example-md maintenance-association working mep 1000 interface ge-2/0/0.0
   set connectivity-fault-management maintenance-domain l2circuit-example-md maintenance-association working mep 1000 direction down
   set connectivity-fault-management maintenance-domain l2circuit-example-md maintenance-association working mep 1000 remote-mep 103

6. Configure OAM on Router PE1 for the protection path.

   [edit protocols oam ethernet]
   set connectivity-fault-management maintenance-domain l2circuit-example-md maintenance-association protection mep 1001 interface ge-2/0/1.0
   set connectivity-fault-management maintenance-domain l2circuit-example-md maintenance-association protection mep 1001 direction down
   set connectivity-fault-management maintenance-domain l2circuit-example-md maintenance-association protection mep 1001 remote-mep 104

7. Configure OAM on Router PE2 to setup the maintenance domain.

```
[edit protocols oam ethernet]
set connectivity-fault-management maintenance-domain l2circuit-example-md
  level 5
```

8. Configure OAM on Router PE2 for the working path.

```
[edit protocols oam ethernet]
set connectivity-fault-management maintenance-domain l2circuit-example-md
  maintenance-association working mep 103 interface ge-2/0/0.0
set connectivity-fault-management maintenance-domain l2circuit-example-md
  maintenance-association working mep 103 direction down
set connectivity-fault-management maintenance-domain l2circuit-example-md
  maintenance-association working mep 103 remote-mep 1000
```

9. Configure OAM on Router PE2 for the protection path.

```
[edit protocols oam ethernet]
set connectivity-fault-management maintenance-domain l2circuit-example-md
  maintenance-association protection mep 104 interface ge-2/0/1.0
set connectivity-fault-management maintenance-domain l2circuit-example-md
  maintenance-association protection mep 104 direction down
set connectivity-fault-management maintenance-domain l2circuit-example-md
  maintenance-association protection mep 104 remote-mep 1001
```

Results From configuration mode on Router PE1, confirm your configuration by entering the **show protocols l2circuit**, **show policy-options**, **show routing-options**, and **show protocols oam ethernet** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host> show protocols l2circuit
neighbor 2.2.2.2 {
  interface ge-2/0/2.0 {
    virtual-circuit-id 2;
    community example;
    connection-protection;
    backup-neighbor 3.3.3.3 {
      virtual-circuit-id 3;
      standby;
    }
  }
}
```

```
user@host> show policy-options
policy-statement load-balance {
  then {
    load-balance per-packet;
  }
}
  policy-statement protection-policy {
    term protect {
      from community example;
      then {
        install-nexthop lsp-regex lsp-primary;
      }
    }
  }
```



```
user@host> show routing-options
forwarding-table {
  export load-balance;
}
```

```
user@host> show protocols oam ethernet
connectivity-fault-management {
  maintenance-domain l2circuit-example-md {
    level 5;
    maintenance-association working {
      mep 1000 {
        interface ge-2/0/0.0;
        direction down;
        remote-mep 103;
      }
    }
    maintenance-association protection {
      mep 1001 {
        interface ge-2/0/1.0;
        direction down;
        remote-mep 104;
      }
    }
  }
}
```

From configuration mode on Router PE2, confirm your configuration by entering the **show protocols oam ethernet** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
connectivity-fault-management {
  maintenance-domain l2circuit-example-md {
    level 5;
    maintenance-association working {
      mep 103 {
        interface ge-2/0/0.0;
        direction down;
        remote-mep 1000;
      }
    }
    maintenance-association protection {
      mep 104 {
        interface ge-2/0/1.0;
        direction down;
        remote-mep 1001;
      }
    }
  }
}
```

Related Documentation

- Example: Configuring Layer 2 Circuit Protect Interfaces on page 139

Example: Configuring an Egress Protection LSP for a Layer 2 Circuit

This example shows how to configure an egress protection LSP.

- Requirements on page 156
- Egress Protection LSP Overview on page 156
- Egress Protection LSP Configuration on page 157

Requirements

Egress protection LSPs are supported on Juniper Networks MX Series routers only. This requirement applies to the PE routers facilitating the egress protection LSP.

Egress Protection LSP Overview

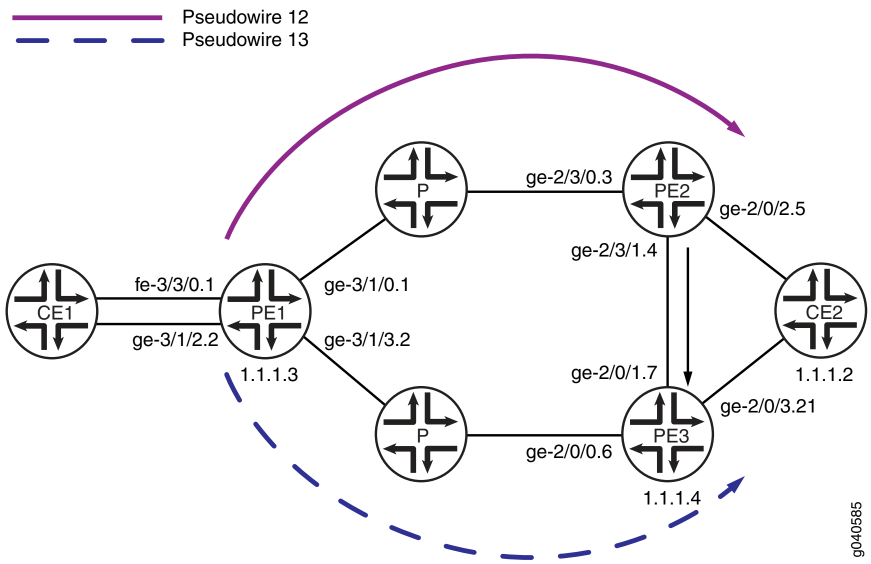
If there is a link or node failure in the core network, a protection mechanism such as MPLS fast reroute can be triggered on the transport LSPs between the PE routers to repair the connection within tens of milliseconds. An egress protection LSP addresses the problem of when a link failure occurs at the edge of the network (for example, a link failure between a PE router and a CE device). Egress protection LSPs do not address the problem of a node failure at the edge of the network (for example, a failure of a PE router). An egress protection LSP is an RSVP-signaled ultimate hop popping LSP.

This example includes the following configuration concepts and statements that are unique to the configuration of an egress protection LSP:

- **context-identifier**—Specifies an IPV4 address used to define the pair of PE routers participating in the egress protection LSP. The context identifier is used to assign an identifier to the protector PE router. The identifier is propagated to the other PE routers participating in the network, making it possible for the protected egress PE router to signal the egress protection LSP to the protector PE router.
- **egress-protection**—Configures the protector information for the protected Layer 2 circuit and configures the protector Layer 2 circuit at the **[edit protocols l2circuit]** hierarchy level. Configures an LSP as an egress protection LSP at the **[edit protocols mpls label-switched-path *lsp-name*]** hierarchy level. It also configures the context identifier at the **[edit protocols mpls]** hierarchy level.
- **protected-l2circuit**—Specifies which Layer 2 circuit is to be protected by the egress protect LSP. This statement includes the following sub-statements: **ingress-pe**, **egress-pe**, and **virtual-circuit-id**. These sub-statements specify the address of the PE router at the ingress of the Layer 2 circuit, the address of the PE router at the egress of the Layer 2 circuit, and the Layer 2 circuit's identifier respectively.
- **protector-interface**—Specify the interface used by the egress protection LSP. In the event of a local link failure to a CE device, the egress protect LSP uses the interface specified to communicate with the protector PE router.
- **protector-pe**—Specify the IPv4 address of the protector PE router. The protector PE router must have a connection to the same CE device as the protected PE router for the egress protect LSP to function. This statement includes the following

sub-statements: **context-identifier** and **lsp**. The **lsp** statement specifies the LSP to be used as the actual egress protection LSP.

Figure 16: Egress Protection LSP Configured from Router PE2 to Router PE3



Pseudowires are configured along two paths, one from router PE1 to router PE2 (pseudowire 12) and one from router PE1 to router PE3 (pseudowire 13). In the event of a failure on the link between router PE2 and device CE2, traffic is switched to the egress protection LSP configured between router PE2 and router PE3 (the protector PE router):

- Device CE1—Traffic origin
- Router PE1—Ingress PE router
- Router PE2—Egress PE router
- Router PE3—Protector PE router
- Device CE2—Traffic destintation

This example shows how to configure routers PE1, PE2, and PE3.

Egress Protection LSP Configuration

- Step-by-Step Procedure on page 159
- Results on page 163

CLI Quick Configuration

To quickly configure an egress protection LSP, copy the following commands into a text file, modify the interface configurations to match your equipment, remove any line breaks, and then paste the commands into the CLI. This group of set commands is for router PE1.

```
set protocols rsvp interface ge-3/1/0.1
set protocols rsvp interface ge-3/1/3.2
set protocols mpls interface ge-3/1/0.1
set protocols mpls interface ge-3/1/3.2
```

```
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-3/1/0.1
set protocols ospf area 0.0.0.0 interface ge-3/1/3.2
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ldp interface ge-3/1/0.1
set protocols ldp interface ge-3/1/3.2
set protocols ldp interface lo0.0
set protocols l2circuit neighbor 1.1.1.3 interface fe-3/3/0.1 virtual-circuit-id 32
set protocols l2circuit neighbor 1.1.1.3 interface fe-3/3/0.1 egress-protection
  protector-interface ge-3/1/2.2
set protocols l2circuit neighbor 1.1.1.4 interface ge-3/1/2.2 virtual-circuit-id 33
set policy-options policy-statement load-balance-example then load-balance per-packet
set routing-options router-id 1.1.1.2
set routing-options forwarding-table export load-balance-example
```

To quickly configure an egress protection LSP, copy the following commands into a text file, modify the interface configurations to match your equipment, remove any line breaks, and then paste the commands into the CLI. This group of set commands is for router PE2.

```
[edit]
set protocols rsvp tunnel-services
set protocols rsvp interface ge-2/3/0.3
set protocols rsvp interface ge-2/3/1.4 link-protection
set protocols ldp interface ge-2/3/0.3
set protocols ldp interface ge-2/3/1.4
set protocols ldp interface lo0.0
set protocols ldp upstream-label-assignment
set protocols mpls label-switched-path protected-lsp to 2.2.3.4
set protocols mpls label-switched-path protected-lsp egress-protection
set protocols mpls interface ge-2/3/0.3
set protocols mpls interface ge-2/3/1.4
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-2/3/0.3
set protocols ospf area 0.0.0.0 interface ge-2/3/1.4
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols l2circuit neighbor 1.1.1.2 interface ge-2/0/2.5 virtual-circuit-id 23
set protocols l2circuit neighbor 1.1.1.2 interface ge-2/0/2.5 egress-protection protector-pe
  1.1.1.4
set protocols l2circuit neighbor 1.1.1.2 interface ge-2/0/2.5 egress-protection protector-pe
  context-identifier 2.2.3.4
set policy-options policy-statement load-balance-example then load-balance per-packet
set routing-options router-id 1.1.1.3
set routing-options forwarding-table export load-balance-example
```

To quickly configure an egress protection LSP, copy the following commands into a text file, modify the interface configurations to match your equipment, remove any line breaks, and then paste the commands into the CLI. This group of set commands is for router PE3.

```
set protocols rsvp tunnel-services
set protocols rsvp interface ge-2/0/0.6
set protocols rsvp interface ge-2/0/1.7
set protocols mpls interface ge-2/0/0.6
set protocols mpls interface ge-2/0/1.7
set protocols mpls egress-protection context-identifier 2.2.3.4 protector
```



```
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-2/0/0.6
set protocols ospf area 0.0.0.0 interface ge-2/0/1.7
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ldp interface ge-2/0/0.6
set protocols ldp interface ge-2/0/1.7
set protocols ldp interface lo0.0
set protocols ldp upstream-label-assignment
set protocols l2circuit neighbor 1.1.1.2 interface ge-2/0/3.21 virtual-circuit-id 42
set protocols l2circuit neighbor 1.1.1.2 interface ge-2/0/3.21 egress-protection
  protected-l2circuit PW1
set protocols l2circuit neighbor 1.1.1.2 interface ge-2/0/3.21 egress-protection
  protected-l2circuit ingress-pe 1.1.1.2
set protocols l2circuit neighbor 1.1.1.2 interface ge-2/0/3.21 egress-protection
  protected-l2circuit egress-pe 1.1.1.3
set protocols l2circuit neighbor 1.1.1.2 interface ge-2/0/3.21 egress-protection
  protected-l2circuit virtual-circuit-id 31
```

Step-by-Step Procedure

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*.

To configure an egress protection LSP, complete the following steps for router PE1:

1. Configure RSVP. Include the interface linked to router PE2 and the interface linked to router PE3.

   ```
   [edit]
   user@PE1# edit protocols rsvp
   [edit protocols rsvp]
   user@PE1# set interface ge-3/1/0.1
   [edit protocols rsvp]
   user@PE1# set interface ge-3/1/3.2
   ```

2. Configure LDP. Include the interface linked to router PE2, the interface linked to router PE3, and the loopback interface.

   ```
   [edit]
   user@PE1# edit protocols ldp
   [edit protocols ldp]
   user@PE1# set interface ge-3/1/0.1
   [edit protocols ldp]
   user@PE1# set interface ge-3/1/3.2
   [edit protocols ldp]
   user@PE1# set interface lo0.0
   ```

3. Configure MPLS. Include the interface linked to router PE2 and the interface linked to router PE3.

   ```
   [edit]
   user@PE1# edit protocols mpls
   [edit protocols mpls]
   user@PE1# set interface ge-3/1/0.1
   [edit protocols mpls]
   user@PE1# set interface ge-3/1/3.2
   ```

4. Configure OSPF. Include the interface linked to router PE2, the interface linked to router PE3, and the loopback interface in the configuration for the OSPF area.

```
[edit]
user@PE1# edit protocols ospf
[edit protocols ospf]
user@PE1# set interface traffic-engineering
[edit protocols ospf]
user@PE1# set area 0.0.0.0 interface ge-3/1/0.1
[edit protocols ospf]
user@PE1# set area 0.0.0.0 interface ge-3/1/3.2
[edit protocols ospf]
user@PE1# set area 0.0.0.0 interface lo0.0 passive
```

5. Configure Layer 2 circuits to use the egress protection LSP to protect against a link failure to device CE1.

```
[edit]
user@PE1# edit protocols l2circuit
[edit protocols l2circuit]
user@PE1# set neighbor 1.1.1.3 interface fe-3/3/0.1 virtual-circuit-id 32
[edit protocols l2circuit]
user@PE1# edit neighbor 1.1.1.3
[edit protocols l2circuit neighbor 1.1.1.3]
user@PE1# set interface fe-3/3/0.1 egress-protection protector-interface ge-3/1/2.2
[edit protocols l2circuit]
user@PE1# set neighbor 1.1.1.4 interface ge-3/1/2.2 virtual-circuit-id 33
```

6. Configure a load balancing policy.

```
[edit]
user@PE1# set policy-options policy-statement load-balance-example then
  load-balance per-packet
```

7. Configure the routing options to export routes based on the load balancing policy.

```
[edit]
user@PE1# set routing-options router-id 1.1.1.2
[edit]
user@PE1# set routing-options forwarding-table export load-balance-example
```

8. If you are done configuring the device, commit the 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*.

To configure an egress protection LSP, complete the following steps for router PE2:

1. Configure RSVP. Include the interface linked to the ingress PE router and the interface linked to the CE device.

```
[edit]
user@PE2# edit protocols rsvp
[edit protocols rsvp]
user@PE2# set tunnel-services
[edit protocols rsvp]
user@PE2# set interface ge-2/3/0.3
```



```
[edit protocols rsvp]
user@PE2# set interface ge-2/3/1.4 link-protection
```

2. Configure LDP. Include the interface linked to the ingress PE router and the interface linked to the CE device.

```
[edit]
user@PE2# edit protocols ldp
[edit protocols ldp]
user@PE2# set interface ge-2/3/0.3
[edit protocols ldp]
user@PE2# set interface ge-2/3/1.4
[edit protocols ldp]
user@PE2# set interface lo0.0
[edit protocols ldp]
user@PE2# set upstream-label-assignment
```

3. Configure MPLS and the LSP which acts as the egress protection LSP.

```
[edit]
user@PE2# edit protocols mpls
[edit protocols mpls]
user@PE2# set interface ge-2/3/0.3
[edit protocols mpls]
user@PE2# set interface ge-2/3/1.4
[edit protocols mpls]
user@PE2# set label-switched-path protected-lsp to 2.2.3.4
[edit protocols mpls]
user@PE2# set label-switched-path protected-lsp egress-protection
```

4. Configure OSPF.

```
[edit]
user@PE2# edit protocols ospf
[edit protocols ospf]
user@PE2# set interface traffic-engineering
[edit protocols ospf]
user@PE2# set interface area 0.0.0.0 interface ge-2/3/0.3
[edit protocols ospf]
user@PE2# set interface area 0.0.0.0 interface ge-2/3/1.4
[edit protocols ospf]
user@PE2# set interface area 0.0.0.0 interface lo0.0 passive
```

5. Configure the Layer 2 circuit to use the egress protection LSP.

```
[edit]
user@PE2# edit protocols l2circuit
[edit protocols l2circuit]
user@PE2# set neighbor 1.1.1.2 interface ge-2/0/2.5 virtual-circuit-id 23
[edit protocols l2circuit]
user@PE2# edit neighbor 1.1.1.2
[edit protocols l2circuit neighbor 1.1.1.2]
user@PE2# set interface ge-2/0/2.5 egress-protectionprotector-pe 1.1.1.4
[edit protocols l2circuit neighbor 1.1.1.2]
user@PE2# set interface ge-2/0/2.5 egress-protection protector-pe
  context-identifier 2.2.3.4
```

6. Configure a load balancing policy.

```
[edit]
user@PE1# set policy-options policy-statement load-balance-example then
  load-balance per-packet
```

7. Configure the routing options to export routes based on the load balancing policy.

```
[edit]
user@PE2# set routing-options router-id 1.1.1.3
[edit]
user@PE2# set routing-options forwarding-table export load-balance-example
```

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

Step-by-Step Procedure

To configure an egress protection LSP, complete the following steps for router PE3:

1. Configure RSVP. Include the interface linked to the ingress PE router and the interface linked to the CE device.

```
[edit]
user@PE3# edit protocols rsvp
[edit protocols rsvp]
user@PE3# set tunnel-services
[edit protocols rsvp]
user@PE3# set interface ge-2/0/0.6
[edit protocols rsvp]
user@PE3# set interface ge-2/0/1.7
```

2. Configure LDP. Include the interface linked to the ingress PE router and the interface linked to the CE device.

```
[edit]
user@PE3# edit protocols ldp
[edit protocols ldp]
user@PE3# set interface ge-2/0/0.6
[edit protocols ldp]
user@PE3# set interface ge-2/0/1.7
[edit protocols ldp]
user@PE3# set interface lo0.0
[edit protocols ldp]
user@PE3# set upstream-label-assignment
```

3. Configure MPLS and the LSP which acts as the egress protection LSP.

```
[edit]
user@PE3# edit protocols mpls
[edit protocols mpls]
user@PE3# set interface ge-2/0/0.6
[edit protocols mpls]
user@PE3# set interface ge-2/0/1.7
[edit protocols mpls]
user@PE3# set egress-protection context-identifier 2.2.3.4 protector
```

4. Configure OSPF.

```
[edit]
user@PE3# edit protocols ospf
[edit protocols ospf]
user@PE3# set interface traffic-engineering
[edit protocols ospf]
```



```
user@PE3# set area 0.0.0.0 interface ge-2/0/0.6
[edit protocols ospf]
user@PE3# set area 0.0.0.0 interface ge-2/0/1.7
[edit protocols ospf]
user@PE3# set area 0.0.0.0 interface lo0.0 passive
```

5. Configure the Layer 2 circuit to use the egress protection LSP.

```
[edit]
user@PE3# edit protocols l2circuit
[edit protocols l2circuit]
user@PE3# set neighbor 1.1.1.2 interface ge-2/0/3.21 virtual-circuit-id 42
[edit protocols l2circuit]
user@PE3# edit neighbor 1.1.1.2
[edit protocols l2circuit neighbor 1.1.1.2]
user@PE3# set interface ge-2/0/3.21 egress-protection protected-l2circuit ingress-pe
  1.1.1.2
[edit protocols l2circuit neighbor 1.1.1.2]
user@PE3# set interface ge-2/0/3.21 egress-protection protected-l2circuit egress-pe
  1.1.1.3
[edit protocols l2circuit neighbor 1.1.1.2]
user@PE3# set interface ge-2/0/3.21 egress-protection
  protected-l2circuitvirtual-circuit-id 31
```

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

Results

From configuration mode, confirm your configuration on router PE1 by entering the **show protocols**, **show policy-options**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@PE1# show protocols
rsvp {
  interface ge-3/1/0.1;
  interface ge-3/1/3.2;
}
mpls {
  interface ge-3/1/0.1;
  interface ge-3/1/3.2;
}
ospf {
  traffic-engineering;
  area 0.0.0.0 {
    interface ge-3/1/0.1;
    interface ge-3/1/3.2;
    interface lo0.0 {
      passive;
    }
  }
}
ldp {
  interface ge-3/1/0.1;
  interface ge-3/1/3.2;
```

```
        interface lo0.0;
    }
    l2circuit {
        neighbor 1.1.1.3 {
            interface fe-3/3/0.1 {
                virtual-circuit-id 32;
                egress-protection {
                    protector-interface ge-3/1/2.2;
                }
            }
        }
        neighbor 1.1.1.4 {
            interface ge-3/1/2.2 {
                virtual-circuit-id 33;
            }
        }
    }
    [edit]
    user@PE1# show policy-options
    policy-statement load-balance-example {
        then {
            load-balance per-packet;
        }
    }
    [edit]
    user@PE1# show routing-options
    router-id 1.1.1.2;
    forwarding-table {
        export load-balance-example;
    }
```

From configuration mode, confirm your configuration on router PE2 by entering the **show protocols**, **show policy-options**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@PE2# show protocols
rsvp {
    tunnel-services;
    interface ge-2/3/0.3;
    interface ge-2/3/1.4 {
        link-protection;
    }
}
mpls {
    label-switched-path protected-lsp {
        to 2.2.3.4;
        egress-protection;
    }
    interface ge-2/3/0.3;
    interface ge-2/3/1.4;
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
```



```
            interface ge-2/3/0.3;
            interface ge-2/3/1.4;
            interface lo0.0 {
                passive;
            }
        }
    }
    ldp {
        interface ge-2/3/0.3;
        interface ge-2/3/1.4;
        interface lo0.0;
        upstream-label-assignment;
    }
    l2circuit {
        neighbor 1.1.1.2 {
            interface ge-2/0/2.5 {
                virtual-circuit-id 23;
                egress-protection {
                    protector-pe 1.1.1.4 context-identifier 2.2.3.4;
                }
            }
        }
    }

[edit]
user@PE2# show policy-options
policy-options {
    policy-statement load-balance-example {
        then {
            load-balance per-packet;
        }
    }
}

[edit]
user@PE2# show routing-options
routing-options {
    router-id 1.1.1.3;
    forwarding-table {
        export load-balance-example;
    }
}
```

From configuration mode, confirm your configuration on router PE3 by entering the **show protocols** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@PE3# show protocols
rsvp {
    tunnel-services;
    interface ge-2/0/0.6;
    interface ge-2/0/1.7;
}
mpls {
    interface ge-2/0/0.6;
    interface ge-2/0/1.7;
```

```
    egress-protection {
      context-identifier 2.2.3.4 {
        protector;
      }
    }
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface ge-2/0/0.6;
      interface ge-2/0/1.7;
      interface lo0.0 {
        passive;
      }
    }
  }
  ldp {
    interface ge-2/0/0.6;
    interface ge-2/0/1.7;
    interface lo0.0;
    upstream-label-assignment;
  }
  l2circuit {
    neighbor 1.1.1.2 {
      interface ge-2/0/3.21 {
        virtual-circuit-id 42;
        egress-protection {
          protected-l2circuit PW1 ingress-pe 1.1.1.2 egress-pe 1.1.1.3 virtual-circuit-id 31;
        }
      }
    }
  }
```

Using the Layer 2 Interworking Interface to Interconnect a Layer 2 Circuit to a Layer 2 VPN

Instead of using a physical Tunnel PIC for looping the packet received from the Layer 2 circuit, the Layer 2 interworking interface uses Junos OS to stitch together both Layer 2 VPN routes.

To configure the interworking interface, include the **iw0** statement. The **iw0** statement is configured at the **[edit interfaces]** hierarchy level. This specifies the peering between two logical interfaces. This configuration is similar to the configuration for a logical tunnel interface. The logical Interfaces must be associated with the endpoints of a Layer 2 circuit and Layer 2 VPN connections.

```
[edit interfaces]
iw0 {
  unit 0 {
    peer-unit 1;
  }
  unit 1 {
    peer-unit 0;
  }
}
```

Configure the Layer 2 circuit protocol by including the **l2circuit** statement at the **[edit protocols]** hierarchy level and specifying the **neighbor** and **iw0** interface.

```
[edit protocols]
l2circuit {
  neighbor 1.2.3.4 {
    interface iw0.0;
  }
}
```

Configure the Layer 2 VPN connection, by including the ***routing-instance-name*** statement at the **[edit routing-instances]** hierarchy level and specifying the **instance-type l2vpn** option.

```
[edit routing-instances]
routing-instance-name {
  instance-type l2vpn;
  interface iw0.1;
  ...
  protocols {
    l2vpn {
      <l2vpn configuration>;
    }
  }
}
```

In addition to the **iw0** interface configuration, Layer 2 interworking **l2iw** protocols must be enabled. Without the **l2iw** configuration, the **l2iw** routes will not be formed, regardless of whether any **iw** interfaces are present. Within the l2iw protocols, only trace options can be configured in the standard fashion. The minimum configuration necessary for the feature to work is shown below:

```
[edit]
protocols {
  l2iw;
}
```

Related Documentation

- *Layer 2 VPN Overview*
- Example: Interconnecting a Layer 2 Circuit with a Layer 2 VPN on page 167

Example: Interconnecting a Layer 2 Circuit with a Layer 2 VPN

This example provides a step-by-step procedure and commands for configuring and verifying a Layer 2 circuit to a Layer 2 VPN. It contains the following sections:

Requirements

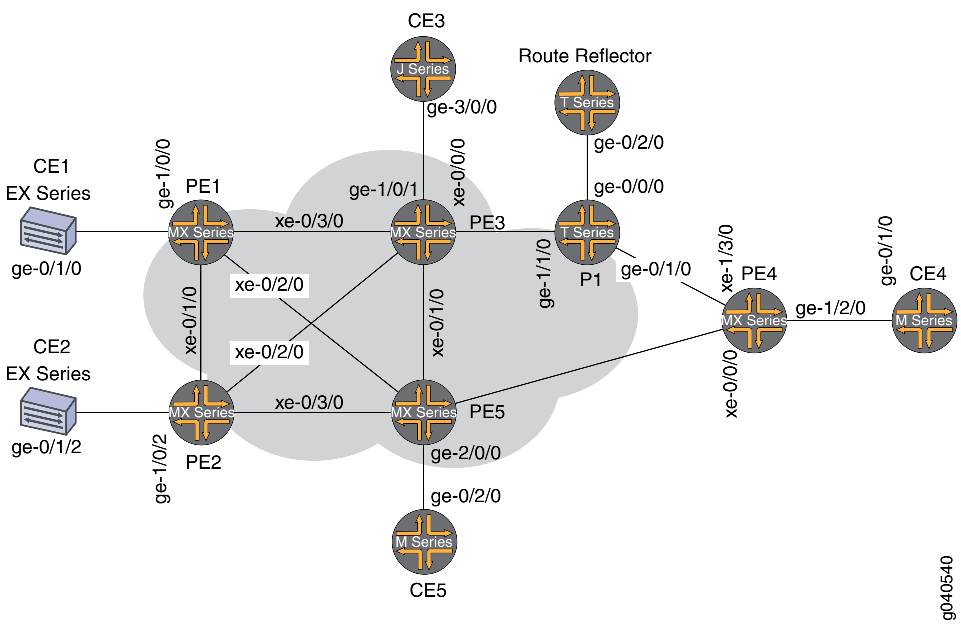
This example uses the following hardware and software components:

- Junos OS Release 9.3 or later
- 2 MX Series 3D Universal Edge Routers
- 2 M Series Multiservice Edge Router
- 1 T Series Core Router
- 1 EX Series Ethernet Switch
- 1 J Series Services Routers

Overview and Topology

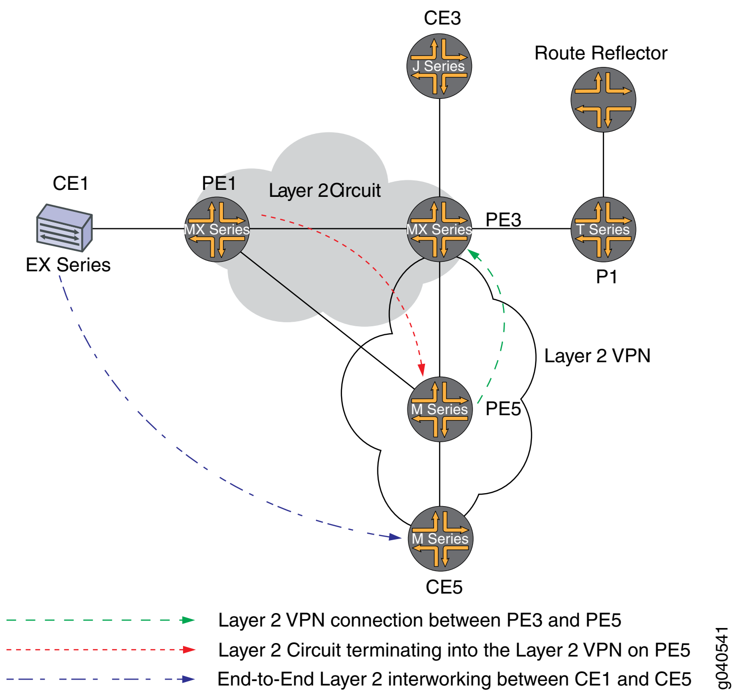
The physical topology of a Layer 2 circuit to a Layer 2 VPN connection is shown in Figure 17 on page 168.

Figure 17: Physical Topology of a Layer 2 Circuit to a Layer 2 VPN Connection



The logical topology of a Layer 2 circuit to a Layer 2 VPN connection is shown in Figure 18 on page 169.

Figure 18: Logical Topology of a Layer 2 Circuit to a Layer 2 VPN Connection



Configuration

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

In this example, the router being configured is identified using the following command prompts:

- **CE1** identifies the customer edge 1 (CE1) router
- **PE1** identifies the provider edge 1 (PE1) router
- **CE3** identifies the customer edge 3 (CE3) router
- **PE3** identifies the provider edge 3 (PE3) router
- **CE5** identifies the customer edge 5 (CE5) router
- **PE5** identifies the provider edge 5 (PE5) router

This example is organized in the following sections:

- Configuring Protocols on the PE and P Routers on page 170
- Verification on page 174

Configuring Protocols on the PE and P Routers

Step-by-Step Procedure

In this example, all of the PE routers and P routers are configured with OSPF as the IGP protocol. The MPLS, LDP, and BGP protocols are enabled on all of the interfaces except **fxp.0**. Core-facing interfaces are enabled with the MPLS address and inet address.

1. Configure all the PE and P routers with OSPF as the IGP. Enable the MPLS, LDP, and BGP protocols on all interfaces except **fxp.0**. LDP is used as the signaling protocol on Router PE1 for the Layer 2 circuit. The following configuration snippet shows the protocol configuration for Router PE1:

```
[edit]
protocols {
  mpls {
    interface all;
    interface fxp0.0 {
      disable;
    }
  }
  bgp {
    group RR {
      type internal;
      local-address 1.1.1.1;
      family l2vpn {
        signaling;
      }
      neighbor 7.7.7.7;
    }
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface all;
      interface fxp0.0 {
        disable;
      }
    }
  }
  ldp {
    interface all;
    interface fxp0.0 {
      disable;
    }
  }
}
```

2. Configure the PE and P routers with OSPF as the IGP. Enable the MPLS, LDP, and BGP protocols on all interfaces except **fxp.0**. BGP is used as the signaling protocol on Router PE3 for the Layer 2 VPN. The following configuration snippet shows the protocol configuration for Router PE3:

```
[edit]
protocols {
  mpls {
    interface all;
```



```
            interface fxp0.0 {
                disable;
            }
        }
        bgp {
            group RR {
                type internal;
                local-address 3.3.3.3;
                family l2vpn {
                    signaling;
                }
                neighbor 7.7.7.7;
            }
        }
        ospf {
            traffic-engineering;
            area 0.0.0.0 {
                interface all;
                interface fxp0.0 {
                    disable;
                }
            }
        }
        ldp {
            interface all;
            interface fxp0.0 {
                disable;
            }
        }
    }
```

Step-by-Step Procedure

Configuring Interfaces

1. On Router PE1, configure the **ge-1/0/0** interface encapsulation. To configure the interface encapsulation, include the **encapsulation** statement and specify the **ethernet-ccc** option (vlan-ccc encapsulation is also supported). Configure the **ge-1/0/0.0** logical interface family for circuit cross-connect functionality. To configure the logical interface family, include the **family** statement and specify the **ccc** option. The encapsulation should be configured the same way for all routers in the Layer 2 circuit domain.

```
[edit interfaces]
ge-1/0/0 {
    encapsulation ethernet-ccc;
    unit 0 {
        family ccc;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 1.1.1.1/32;
        }
    }
}
```

2. Router PE5 is the router that is *stitching* the Layer 2 circuit to the Layer 2 VPN using the interworking interface. The configuration of the peer unit interfaces is what makes the interconnection.

   On Router PE5, configure the **iw0** interface with two logical interfaces. To configure the **iw0** interface, include the **interfaces** statement and specify **iw0** as the interface name. For the unit 0 logical interface, include the **peer-unit** statement and specify the logical interface **unit 1** as the peer interface. For the unit 1 logical interface, include the **peer-unit** statement and specify the logical interface **unit 0** as the peer interface.

   ```
   [edit interfaces]
   iw0 {
      unit 0 {
         encapsulation ethernet-ccc;
         peer-unit 1;
      }
      unit 1 {
         encapsulation ethernet-ccc;
         peer-unit 0;
      }
   }
   ```

3. On Router PE5, configure the logical loopback interface. The loopback interface is used to establish the targeted LDP sessions to Routers PE1 and PE5.

   ```
   [edit interfaces]
   lo0 {
      unit 0 {
         family inet {
            address 5.5.5.5/32;
         }
      }
   }
   ```

Step-by-Step Procedure

Configuring the Layer 2 circuit protocol

1. On Router PE1, configure the IP address of the remote PE router with the **neighbor** statement. The loopback address and router ID of the PE neighbor is commonly the neighbor's IP address. To allow a Layer 2 circuit to be established even though the maximum transmission unit (MTU) configured on the PE router does not match the MTU configured on the remote PE router, include the **ignore-mtu-mismatch** statement.

   ```
   [edit]
   protocols {
      l2circuit {
         neighbor 5.5.5.5 {
            interface ge-1/0/0.0 {
               virtual-circuit-id 100;
               no-control-word;
               ignore-mtu-mismatch;
            }
         }
      }
   }
   ```

2. On Router PE5, configure the IP address of the remote PE router. To configure the IP address of the remote PE router, include the **neighbor** statement and specify the IP address of the loopback interface on Router PE1. Configure the virtual circuit ID to be the same as the virtual circuit ID on the neighbor router. To allow a Layer 2 circuit to be established even though the MTU configured on the local PE router does not match the MTU configured on the remote PE router, include the **ignore-mtu-mismatch** statement. Also disable the use of the control word for demultiplexing by including the **no-control-word** statement.

```
[edit protocols]
l2circuit {
  neighbor 1.1.1.1 {
    interface iw0.0 {
      virtual-circuit-id 100;
      no-control-word;
      ignore-mtu-mismatch;
    }
  }
}
```

3. On Router PE5, configure the Layer 2 VPN protocols by including the **l2vpn** statement at the **[edit routing-instances *routing-instances-name* protocols]** hierarchy level. To configure the **iw0** interface, include the **interfaces** statement and specify **iw0** as the interface name. The **iw0** interface is configured under the Layer 2 VPN protocols to receive the looped packet from the **iw0.1** logical interface. The **l2vpn** protocol is configured on Router PE5 with site CE5, which is configured in the BGP L2VPN routing instance. Router CE1 has communication to Router CE5, through the Layer 2 interworking configuration on Router PE5.

```
[edit]
routing-instances {
  L2VPN {
    instance-type l2vpn;
    interface ge-2/0/0.0;
    interface iw0.1;
    route-distinguisher 65000:5;
    vrf-target target:65000:2;
    protocols {
      l2vpn {
        encapsulation-type ethernet;
        site CE5 {
          site-identifier 5;
          interface ge-2/0/0.0 {
            remote-site-id 3;
          }
        }
        site l2-circuit {
          site-identifier 6;
          interface iw0.1 {
            remote-site-id 3;
          }
        }
      }
    }
  }
```

```
}
```

4. In addition to the **iw0** interface configuration, the Layer 2 interworking **l2iw** protocol must be configured. Without the **l2iw** protocol configuration, the Layer 2 interworking routes are not formed, regardless of whether any **iw** interfaces are present.

   On Router PE5, configure the **l2iw** protocol. To configure the protocol, include the **l2iw** statement at the **[edit protocols]** hierarchy level.

```
[edit]
protocols {
  l2iw;
}
```

Verification

Step-by-Step Procedure

Verifying the Layer 2 Circuit Connection on Router PE1.

1. On Router PE1, use the **show l2circuit connections** command to verify that the Layer 2 Circuit from Router PE1 to Router PE5 is **Up**.

```
user@PE1> show l2circuit connections
Layer-2 Circuit Connections:
Legend for connection status (St)
EI -- encapsulation invalid      NP -- interface h/w not present
MM -- mtu mismatch               Dn -- down
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
VM -- vlan id mismatch           CF -- Call admission control failure
OL -- no outgoing label          IB -- TDM incompatible bitrate
NC -- intf encaps not CCC/TCC    TM -- TDM misconfiguration
BK -- Backup Connection          ST -- Standby Connection
CB -- rcvd cell-bundle size bad  XX -- unknown
SP -- Static Pseudowire

Legend for interface status
Up -- operational
Dn -- down
Neighbor: 5.5.5.5
    Interface                 Type  St     Time last up   # Up trans
    ge-1/0/0.0(vc 100)        rmt   Up     Jan  3 22:00:49 2010    1
      Remote PE: 5.5.5.5, Negotiated control-word: No
      Incoming label: 301328, Outgoing label: 300192
      Local interface: ge-1/0/0.0, Status: Up, Encapsulation: ETHERNET
```

2. On Router PE5, use the **show l2vpn connections** command to verify that the Layer 2 VPN connection is **Up** using the **iw0** peer interface of the Layer 2 circuit.

```
user@PE5> show l2vpn connections
Instance: L2VPN
  Local site: CE5 (5)
    connection-site           Type  St     Time last up     # Up trans
    l2-circuit (6)            loc   OR
    3                         rmt   Up
     Jan  3 22:51:12 2010           1
      Remote PE: 3.3.3.3, Negotiated control-word: Yes (Null)
      Incoming label: 800258, Outgoing label: 800000
      Local interface: ge-2/0/0.0, Status: Up, Encapsulation: ETHERNET
  Local site: l2-circuit (6)
    connection-site           Type  St     Time last up    # Up trans
```



```
    CE5 (5)                    loc    OR
    3                          rmt    Up     Jan  3 22:56:38 2010    1
      Remote PE: 3.3.3.3, Negotiated control-word: Yes (Null)
      Incoming label: 800262, Outgoing label: 800001
      Local interface: iw0.1, Status: Up, Encapsulation: ETHERNET
```

Step-by-Step Procedure

Verifying that the Layer 2 Circuit is terminating into the Layer 2 VPN connection.

1. On Router PE5, use the **show l2circuit connections** command to verify that the Layer 2 circuit is **Up** using the **iw0** interface. This will be looped through the **iwo.1** interface to the Layer 2 VPN.

```
user@PE5> show l2circuit connections

Layer-2 Circuit Connections:
Neighbor: 1.1.1.1
    Interface           Type  St     Time last up  # Up trans
    iw0.0(vc 100)  rmt   Up Jan  3 21:59:07 2010   1
      Remote PE: 1.1.1.1, Negotiated control-word: No
      Incoming label: 300192, Outgoing label: 301328
```

2. On Router PE 5, use the **show route table mpls.0** command to verify the Layer 2 circuit and Layer 2 VPN routes. In the example below, the Layer 2 circuit is associated with LDP label **301328** and the Layer 2 VPN is associated with LDP label **800001**. Notice the two **iw0** interfaces that are used for the Layer 2 interworking route.

```
user@PE5>show route table mpls.0

mpls.0: 18 destinations, 20 routes (18 active, 2 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 5d 20:07:31, metric 1
                      Receive
1                  *[MPLS/0] 5d 20:07:31, metric 1
                      Receive
2                  *[MPLS/0] 5d 20:07:31, metric 1
                      Receive
299776             *[LDP/9] 2d 03:00:51, metric 1
300048             *[LDP/9] 2d 03:00:49, metric 1
                    > to 10.10.6.1 via xe-0/1/0.0, Pop
300048(S=0)        *[LDP/9] 2d 03:00:49, metric 1
                    > to 10.10.6.1 via xe-0/1/0.0, Pop
300192             *[L2IW/6] 19:11:05, metric2 1
                    > to 10.10.6.1 via xe-0/1/0.0, Swap 800001
                    [L2CKT/7] 20:08:36
                    > via iw0.0, Pop
800258             *[L2VPN/7] 19:16:31
                    > via ge-2/0/0.0, Pop       Offset: 4
800262        *[L2IW/6] 19:11:05, metric2 1              > to 10.10.3.1 via xe-1/1/0.0, Swap 301328             [L2VPN/7]
19:11:05           > via iw0.1, Pop      Offset: 4ge-2/0/0.0          *[L2VPN/7] 19:16:31, metric2 1
                    > to 10.10.6.1 via xe-0/1/0.0, Push 800000 Offset: -4
iw0.0              *[L2CKT/7] 20:08:36, metric2 1
                    > to 10.10.3.1 via xe-1/1/0.0, Push 301328
iw0.1              *[L2VPN/7] 19:11:05, metric2 1
                    > to 10.10.6.1 via xe-0/1/0.0, Push 800001 Offset: -4
```


Related Documentation

- *Layer 2 VPN Overview*
- *Layer 2 VPN Applications*
- Using the Layer 2 Interworking Interface to Interconnect a Layer 2 Circuit to a Layer 2 VPN on page 166

Applications for Interconnecting a Layer 2 Circuit with a Layer 2 Circuit

MPLS-based Layer 2 services are growing in demand among enterprise and service providers. This creates new challenges for service providers who want to provide end-to-end value-added services. There are various reasons to stitch different Layer 2 services to one another and to Layer 3 services, for example, to expand the service offerings and to expand geographically. The Junos OS has various features to address the needs of the service provider.

Interconnecting a Layer 2 circuit with a Layer 2 circuit includes the following benefits:

- Interconnecting a Layer 2 circuit with a Layer 2 circuit enables the sharing of a service provider's core network infrastructure between Layer 2 circuit services, reducing the cost of providing those services. A Layer 2 MPLS circuit allows service providers to create a Layer 2 circuit service over an existing IP and MPLS backbone.
- Service providers do not have to invest in separate Layer 2 equipment to provide Layer 2 circuit service. A service provider can configure a provider edge router to run any Layer 2 protocol. Customers who prefer to maintain control over most of the administration of their own networks want Layer 2 circuit connections with their service provider instead of a Layer 3 VPN connection.

Related Documentation

- Example: Interconnecting a Layer 2 Circuit with a Layer 2 Circuit on page 176

Example: Interconnecting a Layer 2 Circuit with a Layer 2 Circuit

This example provides a step-by-step procedure and commands for configuring and verifying a Layer 2 circuit to a Layer 2 circuit interconnection. It contains the following sections:

Requirements

This example uses the following hardware and software components:

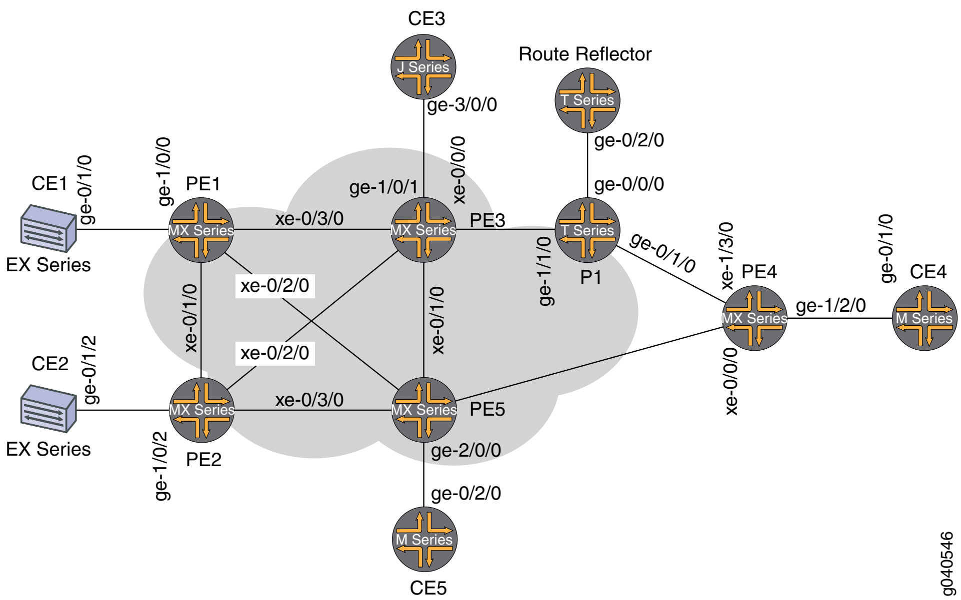
- Junos OS Release 9.3 or later
- 2 MX Series routers
- 2 M Series routers

- 1 T Series router
- 1 EX Series router
- 1 J Series router

Overview and Topology

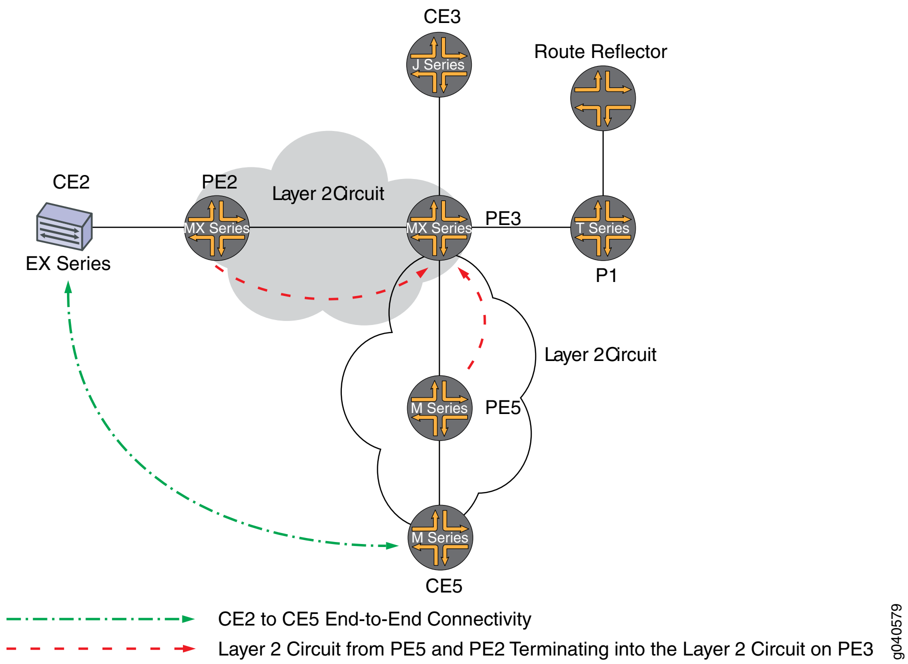
The physical topology of a Layer 2 circuit to Layer 2 circuit interconnection is shown in Figure 19 on page 177

Figure 19: Physical Topology of a Layer 2 Circuit Terminating into a Layer 2 Circuit



The logical topology of a Layer 2 circuit to Layer 2 circuit interconnection is shown in Figure 20 on page 178

Figure 20: Logical Topology of a Layer 2 Circuit Terminating into a Layer 2 Circuit



Configuration

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

In this example, the router being configured is identified using the following command prompts:

- **CE2** identifies the customer edge 2 (CE2) router
- **PE1** identifies the provider edge 1 (PE1) router
- **CE3** identifies the customer edge 3 (CE3) router
- **PE3** identifies the provider edge 3 (PE3) router
- **CE5** identifies the customer edge 5 (CE5) router
- **PE5** identifies the provider edge 5 (PE5) router

This example contains the following procedures:

- Configuring PE Router Customer-facing and Loopback Interfaces on page 179
- Configuring Core-facing Interfaces on page 180
- Configuring Protocols on page 181
- Configuring the Layer 2 Circuits on page 182
- Interconnecting the Layer 2 Circuits on page 184

Configuring PE Router Customer-facing and Loopback Interfaces

Step-by-Step Procedure

To begin building the interconnection, configure the interfaces on the PE routers. If your network contains provider (P) routers, configure the interfaces on the P routers also. This example shows the configuration for Router PE1 and Router PE5.

1. On Router PE1, configure the **ge-1/0/0** interface encapsulation. To configure the interface encapsulation, include the **encapsulation** statement and specify the **ethernet-ccc** option (vlan-ccc encapsulation is also supported). Configure the **ge-1/0/0.0** logical interface family for circuit cross-connect functionality. To configure the logical interface family, include the **family** statement and specify the **ccc** option.

```
[edit interfaces]
ge-1/0/0 {
   encapsulation ethernet-ccc;
   unit 0 {
      family ccc;
   }
}
lo0 {
   unit 0 {
      family inet {
         address 1.1.1.1/32;
      }
   }
}
```

2. On Router PE5, configure the **ge-2/0/0** interface encapsulation. To configure the interface encapsulation, include the **encapsulation** statement and specify the **ethernet-ccc** option. Configure the **ge-2/0/0.0** logical interface family for circuit cross-connect functionality. To configure the logical interface family, include the **family** statement and specify the **ccc** option

```
[edit interfaces]
ge-2/0/0 {
   encapsulation ethernet-ccc;
   unit 0 {
      family ccc;
   }
}
lo0 {
   unit 0 {
      family inet {
         address 5.5.5.5/32;
      }
   }
}
```

3. On Router PE3, configure the logical loopback interface. The loopback interface is used to establish the targeted LDP sessions to Routers PE1 and PE5.

```
[edit interfaces]
```

```
lo0 {
   unit 0 {
      family inet {
         address 3.3.3.3/32;
      }
   }
}
```

Configuring Core-facing Interfaces

Step-by-Step Procedure

This procedure describes how to configure the core-facing interfaces on the PE routers. This example does not include all the core-facing interfaces shown in the physical topology illustration. Enable the **mpls** and **inet** address families on the core-facing interfaces.

1. On Router PE1, configure the **xe-0/3/0** interface. Include the **family** statement and specify the **inet** address family. Include the **address** statement and specify **10.10.1.1/30** as the interface address. Include the **family** statement and specify the **mpls** address family.

   ```
   [edit interfaces]
   xe-0/3/0 {
      unit 0 {
         family inet {
            address 10.10.1.1/30;
         }
         family mpls;
      }
   }
   ```

2. On Router PE3, configure the core-facing interfaces. Include the **family** statement and specify the **inet** address family. Include the **address** statement and specify the IPv4 addresses shown in the example as the interface addresses. Include the **family** statement and specify the **mpls** address family. In the example, the **xe-0/0/0** interface is connected to the route reflector, the **xe-0/1/0** interface is connected to Router PE5, the **xe-0/2/0** interface is connected to Router PE2, and the **xe-0/3/0** interface is connected to Router PE1.

   ```
   [edit interfaces]
   xe-0/0/0 {
      unit 0 {
         family inet {
            address 10.10.20.2/30;
         }
         family mpls;
      }
   }
   xe-0/1/0 {
      unit 0 {
         family inet {
            address 10.10.6.1/30;
         }
         family mpls;
      }
   ```



```
}
xe-0/2/0 {
   unit 0 {
      family inet {
         address 10.10.5.2/30;
      }
      family mpls;
   }
}
xe-0/3/0 {
   unit 0 {
      family inet {
         address 10.10.1.2/30;
      }
      family mpls;
   }
}
```

3. On Router PE5, configure the **xe-0/1/0** interface. Include the **family** statement and specify the **inet** address family. Include the **address** statement and specify **10.10.6.2/30** as the interface address. Include the **family** statement and specify the **mpls** address family.

```
[edit interfaces]
xe-0/1/0 {
   unit 0 {
      family inet {
         address 10.10.6.2/30;
      }
      family mpls;
   }
}
```

Configuring Protocols

Step-by-Step Procedure

This procedure describes how to configure the protocols used in this example. If your network contains P routers, configure the protocols on the P routers also.

Configure all of the PE routers and P routers with OSPF as the IGP protocol. Enable MPLS and LDP protocols on all of the interfaces except **fxp.0**.

1. On Router PE1, enable OSPF as the IGP. Enable the MPLS and LDP protocols on all interfaces except **fxp.0**. LDP is used as the signaling protocol on Router PE1 for the Layer 2 circuit. The following configuration snippet shows the protocol configuration for Router PE1:

```
[edit]
protocols {
   mpls {
      interface all;
      interface fxp0.0 {
         disable;
      }
   }
   ospf {
      traffic-engineering;
```

```
      area 0.0.0.0 {
        interface all;
        interface fxp0.0 {
          disable;
        }
      }
    }
    ldp {
      interface all;
      interface fxp0.0 {
        disable;
      }
    }
  }
```

2. Configure the PE and P routers with OSPF as the IGP. Enable the MPLS and LDP protocols on all interfaces except **fxp.0**. The following configuration snippet shows the protocol configuration for Router PE3:

```
[edit]
protocols {
  mpls {
    interface all;
    interface fxp0.0 {
      disable;
    }
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface all;
      interface fxp0.0 {
        disable;
      }
    }
  }
  ldp {
    interface all;
    interface fxp0.0 {
      disable;
    }
  }
}
```

Configuring the Layer 2 Circuits

Step-by-Step Procedure

This procedure describes how to configure the Layer 2 circuits.

NOTE: In this example the **ignore-mtu-mismatch** statement is required for the circuit to come up.

1. On Router PE1, configure the Layer 2 circuit. Include the **l2circuit** statement. Include the **neighbor** statement and specify the loopback IPv4 address of Router PE3 as

the neighbor. Include the interface statement and specify **ge-1/0/0.0** as the logical interface that is participating in the Layer 2 circuit. Include the **virtual-circuit-id** statement and specify **100** as the identifier. Include the **ignore-mtu-mismatch** statement to allow a Layer 2 circuit to be established even though the maximum transmission unit (MTU) configured on the local PE router does not match the MTU configured on the remote PE router.

```
[edit]
protocols {
  l2circuit {
    neighbor 3.3.3.3 {
      interface ge-1/0/0.0 {
        virtual-circuit-id 100;
        ignore-mtu-mismatch;
      }
    }
  }
}
```

2. On Router PE5, configure the Layer 2 circuit. Include the **l2circuit** statement. Include the **neighbor** statement and specify the loopback IPv4 address of Router PE3 as the neighbor. Include the interface statement and specify **ge-2/0/0.0** as the logical interface that is participating in the Layer 2 circuit. Include the **virtual-circuit-id** statement and specify **200** as the identifier. Include the **ignore-mtu-mismatch** statement to allow a Layer 2 circuit to be established even though the MTU configured on the local PE router does not match the MTU configured on the remote PE router.

```
[edit]
protocols {
  l2circuit {
    neighbor 3.3.3.3 {
      interface ge-2/0/0.0 {
        virtual-circuit-id 200;
        ignore-mtu-mismatch;
      }
    }
  }
}
```

3. On Router PE3, configure the Layer 2 circuit to Router PE1. Include the **l2circuit** statement. Include the **neighbor** statement and specify the loopback IPv4 address of Router PE1 as the neighbor. Include the interface statement and specify **iw0.0** as the logical interworking interface that is participating in the Layer 2 circuit. Include the **virtual-circuit-id** statement and specify **100** as the identifier. Include the **ignore-mtu-mismatch** statement to allow a Layer 2 circuit to be established even though the MTU configured on the local PE router does not match the MTU configured on the remote PE router.

   On Router PE3, configure the Layer 2 circuit to Router PE5. Include the **l2circuit** statement. Include the **neighbor** statement and specify the loopback IPv4 address of Router PE5 as the neighbor. Include the interface statement and specify **iw0.1** as the logical interworking interface that is participating in the Layer 2 circuit. Include the **virtual-circuit-id** statement and specify **200** as the identifier. Include the **ignore-mtu-mismatch** statement.

```
[edit protocols]
l2circuit {
   neighbor 1.1.1.1 {
      interface iw0.0 {
         virtual-circuit-id 100;
         ignore-mtu-mismatch;
      }
   }
   neighbor 5.5.5.5 {
      interface iw0.1 {
         virtual-circuit-id 200;
         ignore-mtu-mismatch;
      }
   }
}
```

Interconnecting the Layer 2 Circuits

Step-by-Step Procedure

Router PE3 is the router that is *stitching* the Layer 2 circuits together using the interworking interface. The configuration of the peer unit interfaces is what makes the interconnection.

1. On Router PE3, configure the **iw0.0** interface. Include the **encapsulation** statement and specify the **ethernet-ccc** option. Include the **peer-unit** statement and specify the logical interface unit **1** as the peer tunnel interface.

   On Router PE3, configure the **iw0.1** interface. Include the **encapsulation** statement and specify the **ethernet-ccc** option. Include the **peer-unit** statement and specify the logical interface unit **0** as the peer tunnel interface.

   ```
   [edit interfaces]
   iw0 {
      unit 0 {
         encapsulation ethernet-ccc;
         peer-unit 1;
      }
      unit 1 {
         encapsulation ethernet-ccc;
         peer-unit 0;
      }
   }
   ```

2. On Router PE3, configure the Layer 2 interworking **l2iw** protocol. To configure the Layer 2 interworking protocol, include the **l2iw** statement at the **[edit protocols]** hierarchy level.

   ```
   [edit]
   protocols {
      l2iw;
   }
   ```

3. On each router, commit the configuration.

   ```
   user@host> commit check
   ```


configuration check succeeds
user@host> **commit**

Verifying the Layer 2 Circuit to Layer 2 Circuit Interconnection

Step-by-Step Procedure

Verify that the Layer 2 circuit connection on Router PE1 is up, the LDP neighbors are correct, and the MPLS label operations are correct.

1. On Router PE1, use the **show l2circuit connections** command to verify that the Layer 2 circuit from Router PE1 to Router PE3 is **Up**.

```
user@PE1> show l2circuit connections
Layer-2 Circuit Connections:
Legend for connection status (St)
EI -- encapsulation invalid      NP -- interface h/w not present
MM -- mtu mismatch               Dn -- down
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
VM -- vlan id mismatch           CF -- Call admission control failure
OL -- no outgoing label          IB -- TDM incompatible bitrate
NC -- intf encaps not CCC/TCC    TM -- TDM misconfiguration
BK -- Backup Connection          ST -- Standby Connection
CB -- rcvd cell-bundle size bad  XX -- unknown
SP -- Static Pseudowire

Legend for interface status
Up -- operational
Dn -- down
Neighbor: 3.3.3.3
    Interface                 Type  St     Time last up     # Up trans
    ge-1/0/0.0(vc 100)        rmt   Up     Jan  5 22:00:49 2010    1
      Remote PE: 3.3.3.3, Negotiated control-word: Yes (Null)
      Incoming label: 301328, Outgoing label: 314736
      Local interface: ge-1/0/0.0, Status: Up, Encapsulation: ETHERNET
```

2. On Router PE1, use the **show ldp neighbor** command to verify that the IPv4 address of Router PE3 is shown as the LDP neighbor.

```
user@PE1> show ldp neighbor

Address            Interface          Label space ID         Hold time
3.3.3.3            lo0.0              3.3.3.3:0                41
```

3. On Router PE 1, use the **show route table mpls.0** command to verify the Layer 2 circuit is using the LDP label to Router PE3 in both directions (Push and Pop). In the example below, the Layer 2 circuit is associated with LDP label **301328**.

```
user@PE1> show route table mpls.0

mpls.0: 13 destinations, 13 routes (13 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 1w1d 08:25:39, metric 1
                      Receive
1                  *[MPLS/0] 1w1d 08:25:39, metric 1
                      Receive
2                  *[MPLS/0] 1w1d 08:25:39, metric 1
                      Receive
300432             *[LDP/9] 3d 01:13:57, metric 1
                    > to 10.10.2.2 via xe-0/1/0.0, Pop
```

```
300432(S=0)        *[LDP/9] 3d 01:13:57, metric 1
                    > to 10.10.2.2 via xe-0/1/0.0, Pop
300768             *[LDP/9] 3d 01:13:57, metric 1
                    > to 10.10.3.2 via xe-0/2/0.0, Pop
300768(S=0)        *[LDP/9] 3d 01:13:57, metric 1
                    > to 10.10.3.2 via xe-0/2/0.0, Pop
300912             *[LDP/9] 3d 01:13:57, metric 1
                    > to 10.10.3.2 via xe-0/2/0.0, Swap 299856
301264             *[LDP/9] 3d 01:13:53, metric 1
                    > to 10.10.1.2 via xe-0/3/0.0, Swap 308224
301312             *[LDP/9] 3d 01:13:56, metric 1
                    > to 10.10.1.2 via xe-0/3/0.0, Pop
301312(S=0)        *[LDP/9] 3d 01:13:56, metric 1
                    > to 10.10.1.2 via xe-0/3/0.0, Pop
301328         *[L2CKT/7] 02:33:26         > via ge-1/0/0.0, Pop      Offset: 4
ge-1/0/0.0       *[L2CKT/7] 02:33:26, metric2 1         > to 10.10.1.2 via xe-0/3/0.0,
 Push 314736 Offset: -4
```

4. On Router PE3, use the **show l2circuit connections** command to verify that the Layer 2 circuit from Router PE3 to Router PE5 is **Up**, that the Layer 2 circuit from Router PE3 to Router PE1 is **Up**, that the connections to Router PE1 and Router PE5 use the iw0 interface, and that the status for both local iw0 interfaces is **Up**.

```
user@PE3>  show l2circuit connections
Layer-2 Circuit Connections:
Legend for connection status (St)
EI -- encapsulation invalid      NP -- interface h/w not present
MM -- mtu mismatch               Dn -- down
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
VM -- vlan id mismatch           CF -- Call admission control failure
OL -- no outgoing label          IB -- TDM incompatible bitrate
NC -- intf encaps not CCC/TCC    TM -- TDM misconfiguration
BK -- Backup Connection          ST -- Standby Connection
CB -- rcvd cell-bundle size bad  XX -- unknown
SP -- Static Pseudowire

Legend for interface status
Up -- operational
Dn -- down
Neighbor: 1.1.1.1
    Interface                 Type  St     Time last up          # Up trans
    iw0.0(vc 100)             rmt   Up     Jan  5 13:50:14 2010           1
      Remote PE: 1.1.1.1, Negotiated control-word: Yes (Null)
      Incoming label: 314736, Outgoing label: 301328
      Local interface: iw0.0, Status: Up, Encapsulation: ETHERNET
Neighbor: 5.5.5.5
    Interface                 Type  St     Time last up          # Up trans
    iw0.1(vc 200)             rmt   Up     Jan  5 13:49:58 2010           1
      Remote PE: 5.5.5.5, Negotiated control-word: Yes (Null)
      Incoming label: 314752, Outgoing label: 300208
      Local interface: iw0.1, Status: Up, Encapsulation: ETHERNET
```

5. On Router PE3, use the **show ldp neighbor** command to verify that the correct IPv4 addresses are shown as the LDP neighbor.

```
user@PE3>  show ldp neighbor
Address            Interface          Label space ID         Hold time
1.1.1.1            lo0.0              1.1.1.1:0                44
2.2.2.2            lo0.0              2.2.2.2:0                42
```



```
4.4.4.4            lo0.0              4.4.4.4:0                31
5.5.5.5            lo0.0              5.5.5.5:0                44
```

6. On Router PE3, use the **show route table mpls.0** command to verify that the **mpls.0** routing table is populated with the Layer 2 interworking routes. Notice that in this example, the router is swapping label **314736** received from Router PE1 on the **iw0.0** to label **301328**.

```
user@PE3> show route table mpls.0
mpls.0: 16 destinations, 18 routes (16 active, 2 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 1w1d 08:28:24, metric 1
                      Receive
1                  *[MPLS/0] 1w1d 08:28:24, metric 1
                      Receive
2                  *[MPLS/0] 1w1d 08:28:24, metric 1
                      Receive
308160             *[LDP/9] 3d 01:16:55, metric 1
                    > to 10.10.1.1 via xe-0/3/0.0, Pop
308160(S=0)        *[LDP/9] 3d 01:16:55, metric 1
                    > to 10.10.1.1 via xe-0/3/0.0, Pop
308176             *[LDP/9] 3d 01:16:54, metric 1
                    > to 10.10.6.2 via xe-0/1/0.0, Pop
308176(S=0)        *[LDP/9] 3d 01:16:54, metric 1
                    > to 10.10.6.2 via xe-0/1/0.0, Pop
308192             *[LDP/9] 00:21:40, metric 1
                    > to 10.10.20.1 via xe-0/0/0.0, Swap 601649
                      to 10.10.6.2 via xe-0/1/0.0, Swap 299856
308208             *[LDP/9] 3d 01:16:54, metric 1
                    > to 10.10.5.1 via xe-0/2/0.0, Pop
308208(S=0)        *[LDP/9] 3d 01:16:54, metric 1
                    > to 10.10.5.1 via xe-0/2/0.0, Pop
308224             *[LDP/9] 3d 01:16:52, metric 1
                    > to 10.10.20.1 via xe-0/0/0.0, Pop
308224(S=0)        *[LDP/9] 3d 01:16:52, metric 1
                    > to 10.10.20.1 via xe-0/0/0.0, Pop
314736             *[L2IW/6] 02:35:31, metric2 1
                    > to 10.10.6.2 via xe-0/1/0.0, Swap 300208
                    [L2CKT/7] 02:35:31
                    > via iw0.0, Pop      Offset: 4
314752             *[L2IW/6] 02:35:31, metric2 1
                    > to 10.10.1.1 via xe-0/3/0.0, Swap 301328
                    [L2CKT/7] 02:35:47
                    > via iw0.1, Pop      Offset: 4
iw0.0              *[L2CKT/7] 02:35:31, metric2 1
                    > to 10.10.1.1 via xe-0/3/0.0, Push 301328 Offset: -4
iw0.1              *[L2CKT/7] 02:35:47, metric2 1
                    > to 10.10.6.2 via xe-0/1/0.0, Push 300208 Offset: -4
```

7. Verify that Router CE1 can send traffic to and receive traffic from Router CE5 across the interconnection, using the **ping** command.

```
user@CE1>ping 40.40.40.11
PING 40.40.40.11 (40.40.40.11): 56 data bytes
64 bytes from 40.40.40.11: icmp_seq=1 ttl=64 time=22.425 ms
64 bytes from 40.40.40.11: icmp_seq=2 ttl=64 time=1.299 ms
```

```
64 bytes from 40.40.40.11: icmp_seq=3 ttl=64 time=1.032 ms
64 bytes from 40.40.40.11: icmp_seq=4 ttl=64 time=1.029 ms
```

8. Verify that Router CE5 can send traffic to and receive traffic from Router CE1 across the interconnection, using the **ping** command.

```
user@CE5>ping 40.40.40.1
PING 40.40.40.1 (40.40.40.1): 56 data bytes
64 bytes from 40.40.40.1: icmp_seq=0 ttl=64 time=1.077 ms
64 bytes from 40.40.40.1: icmp_seq=1 ttl=64 time=0.957 ms
64 bytes from 40.40.40.1: icmp_seq=2 ttl=64 time=1.057 ms 1.017 ms
```

Results

The configuration and verification of this example has been completed. The following section is for your reference.

The relevant sample configuration for Router PE1 follows.

Router PE1

```
[edit]
interfaces {
  xe-0/1/0 {
    unit 0 {
      family inet {
        address 10.10.2.1/30;
      }
      family mpls;
    }
  }
  xe-0/2/0 {
    unit 0 {
      family inet {
        address 10.10.3..1/30;
      }
      family mpls;
    }
  }
  xe-0/3/0 {
    unit 0 {
      family inet {
        address 10.10.1.1/30;
      }
      family mpls;
    }
  }
  ge-1/0/0 {
    encapsulation ethernet-ccc;
    unit 0 {
      family ccc;
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 1.1.1.1/32;
      }
    }
```



```
    }
}
forwarding-options {
    hash-key {
        family inet {
            layer-3;
            layer-4;
        }
        family mpls {
            label-1;
            label-2;
        }
    }
}
routing-options {
    static {
        route 172.0.0.0/8 next-hop 172.19.59.1;
    }
    autonomous-system 65000;
}
protocols {
    mpls {
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface all;
            interface fxp0.0 {
                disable;
            }
        }
    }
    ldp {
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
    l2circuit {
        neighbor 3.3.3.3 {
            interface ge-1/0/0.0 {
                virtual-circuit-id 100;
                ignore-mtu-mismatch;
            }
        }
    }
}
```

The relevant sample configuration for Router PE3 follows.

Router PE3

```
[edit]
interfaces {
```

```
    xe-0/0/0 {
        unit 0 {
            family inet {
                address 10.10.20.2/30;
            }
            family mpls;
        }
    }
    xe-0/1/0 {
        unit 0 {
            family inet {
                address 10.10.6.1/30;
            }
            family mpls;
        }
    }
    xe-0/2/0 {
        unit 0 {
            family inet {
                address 10.10.5.2/30;
            }
            family mpls;
        }
    }
    xe-0/3/0 {
        unit 0 {
            family inet {
                address 10.10.1.2/30;
            }
            family mpls;
        }
    }
    ge-1/0/1 {
        encapsulation ethernet-ccc;
        unit 0 {
            family ccc;
        }
    }
    iw0 {
        unit 0 {
            encapsulation ethernet-ccc;
            peer-unit 1;
        }
        unit 1 {
            encapsulation ethernet-ccc;
            peer-unit 0;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 3.3.3.3/32;
            }
        }
    }
}
```



```
routing-options {
  static {
    route 172.0.0.0/8 next-hop 172.19.59.1;
  }
  autonomous-system 65000;
}
protocols {
  l2iw;
  mpls {
    interface all;
    interface fxp0.0 {
      disable;
    }
  }
  ospf {
    area 0.0.0.0 {
      interface all;
      interface fxp0.0 {
        disable;
      }
    }
  }
  ldp {
    interface all;
    interface fxp0.0 {
      disable;
    }
  }
  l2circuit {
    neighbor 1.1.1.1 {
      interface iw0.0 {
        virtual-circuit-id 100;
        ignore-mtu-mismatch;
      }
    }
    neighbor 5.5.5.5 {
      interface iw0.1 {
        virtual-circuit-id 200;
        ignore-mtu-mismatch;
      }
    }
  }
}
```

Related Documentation

- Applications for Interconnecting a Layer 2 Circuit with a Layer 2 Circuit on page 176

Applications for Interconnecting a Layer 2 Circuit with a Layer 3 VPN

MPLS-based Layer 2 services are growing in demand among enterprise and service providers. This creates new challenges related to interoperability between Layer 2 and Layer 3 services for service providers who want to provide end-to-end value-added services. There are various reasons to stitch different Layer 2 services to one another and to Layer 3 services. For example, to expand the service offerings and to expand geographically. The Junos OS has various features to address the needs of the service provider.

Interconnecting a Layer 2 Circuit with a Layer 3 VPN provides the following benefits:

- Interconnecting a Layer 2 Circuit with a Layer 3 VPN enables the sharing of a service provider's core network infrastructure between IP and Layer 2 circuit services, reducing the cost of providing those services. A Layer 2 MPLS circuit allows service providers to create a Layer 2 circuit service over an existing IP and MPLS backbone.
- Service providers do not have to invest in separate Layer 2 equipment to provide Layer 2 circuit service. A service provider can configure a provider edge router to run any Layer 3 protocol in addition to the Layer 2 protocols. Customers who prefer to maintain control over most of the administration of their own networks want Layer 2 circuit connections with their service provider instead of a Layer 3 VPN connection.

Related Documentation

- *Layer 3 VPN Overview*
- Example: Interconnecting a Layer 2 Circuit with a Layer 3 VPN on page 192

Example: Interconnecting a Layer 2 Circuit with a Layer 3 VPN

This example provides a step-by-step procedure and commands for configuring and verifying a Layer 2 circuit to Layer 3 VPN interconnection. It contains the following sections:

Requirements

This example uses the following hardware and software components:

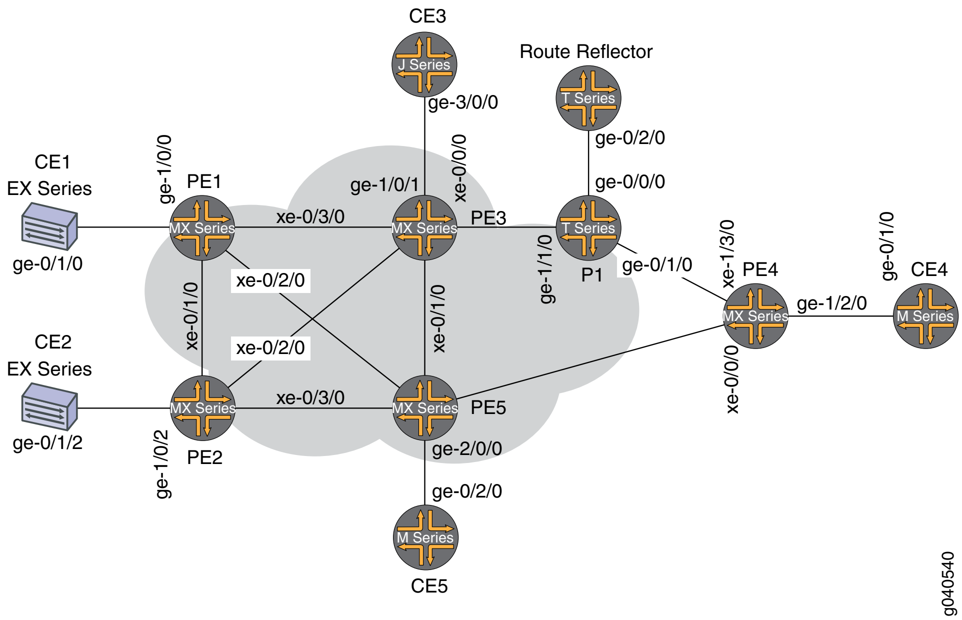
- Junos OS Release 9.3 or later
- 3 MX Series 3D Universal Edge Routers
- 1 M Series Multiservice Edge Router
- 1 T Series Core Router

- 1 EX Series Ethernet Switch
- 1 J Series Services Routers

Overview and Topology

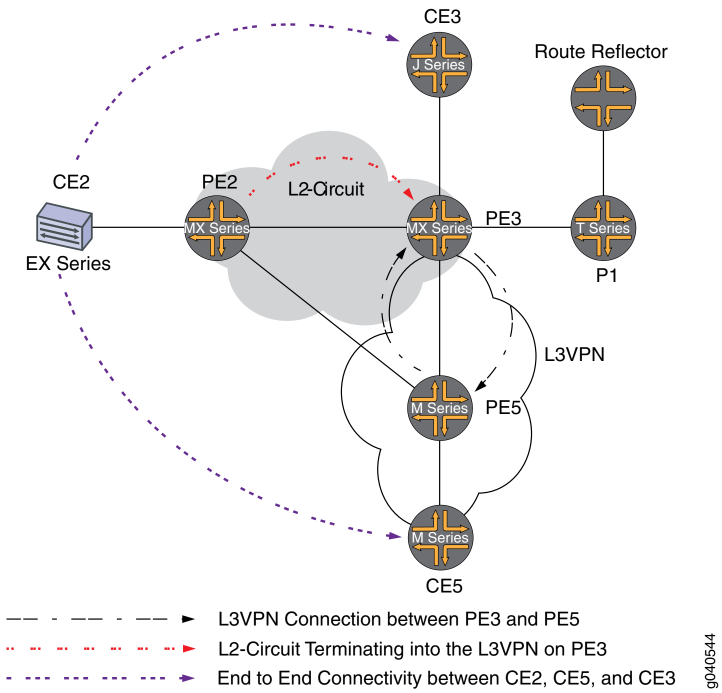
The physical topology of a Layer 2 circuit to Layer 3 VPN interconnection is shown in Figure 21 on page 193.

Figure 21: Physical Topology of a Layer 2 Circuit to Layer 3 VPN Interconnection



The logical topology of a Layer 2 circuit to Layer 3 VPN interconnection is shown in Figure 22 on page 194.

Figure 22: Logical Topology of a Layer 2 Circuit to Layer 3 VPN Interconnection



Configuration

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

In this example, the router being configured is identified using the following command prompts:

- **CE2** identifies the customer edge 2 (CE2) router
- **PE1** identifies the provider edge 1 (PE1) router
- **CE3** identifies the customer edge 3 (CE3) router
- **PE3** identifies the provider edge 3 (PE3) router
- **CE5** identifies the customer edge 5 (CE5) router
- **PE5** identifies the provider edge 5 (PE5) router

This example contains the following procedures:

- Configuring PE Router Customer-facing and Loopback Interfaces on page 195
- Configuring Core-facing Interfaces on page 196
- Configuring Protocols on page 198
- Configuring Routing Instances and Layer 2 Circuits on page 200

- Configuring the Route Reflector on page 202
- Interconnecting the Layer 2 Circuit with the Layer 3 VPN on page 203

Configuring PE Router Customer-facing and Loopback Interfaces

Step-by-Step Procedure

To begin building the interconnection, configure the interfaces on the PE routers. If your network contains provider (P) routers, configure the interfaces on the P routers also. This example shows the configuration for Router PE2, Router PE3, and Router PE5.

1. On Router PE2, configure the **ge-1/0/2** interface encapsulation. To configure the interface encapsulation, include the **encapsulation** statement and specify the **ethernet-ccc** option (**vlan-ccc** encapsulation is also supported). Configure the **ge-1/0/2.0** logical interface family for circuit cross-connect functionality. To configure the logical interface family, include the **family** statement and specify the **ccc** option. The encapsulation should be configured the same way for all routers in the Layer 2 circuit domain.

```
[edit interfaces]
ge-1/0/2 {
   encapsulation ethernet-ccc;
   unit 0 {
      family ccc;
   }
}
```

2. On Router PE2, configure the **lo0.0** interface. Include the **family** statement and specify the **inet** option. Include the **address** statement and specify **2.2.2.2/32** as the loopback IPv4 address.

```
[edit interfaces]
lo0 {
   unit 0 {
      family inet {
         address 2.2.2.2/32;
      }
   }
}
```

3. On Router PE3, configure the **ge-1/0/1** interface. Include the **family** statement and specify the **inet** option. Include the **address** statement and specify **90.90.90.1/24** as the interface address for this device.

```
[edit interfaces]
ge-1/0/1 {
   unit 0 {
      family inet {
         address 90.90.90.1/24;
      }
   }
}
```

4. On Router PE3, configure the **lo0.0** loopback interface. Include the **family** statement and specify the **inet** option. Include the **address** statement and specify **3.3.3.3/32** as the loopback IPv4 address for this router.

```
[edit interfaces]
```

```
lo0 {
  unit 0 {
    family inet {
      address 3.3.3.3/32;
    }
  }
}
```

5. On Router PE5, configure the **ge-2/0/0** interface. Include the **family** statement and specify the **inet** option. Include the **address** statement and specify **80.80.80.1/24** as the interface address.

```
[edit interfaces]
ge-2/0/0 {
  unit 0 {
    family inet {
      address 80.80.80.1/24;
    }
  }
}
```

6. On Router PE5, configure the **lo0.0** interface. Include the **family** statement and specify the **inet** option. Include the **address** statement and specify **5.5.5.5/32** as the loopback IPv4 address for this router.

```
[edit interfaces]
lo0 {
  unit 0 {
    family inet {
      address 5.5.5.5/32;
    }
  }
}
```

Configuring Core-facing Interfaces

Step-by-Step Procedure

This procedure describes how to configure the core-facing interfaces on the PE routers. This example does not include all the core-facing interfaces shown in the physical topology illustration. Enable the **mpls** and **inet** address families on the core-facing interfaces.

1. On Router PE2, configure the **xe-0/2/0** interface. Include the **family** statement and specify the **inet** address family. Include the **address** statement and specify **10.10.5.1/30** as the interface address. Include the **family** statement and specify the **mpls** address family.

```
[edit interfaces]
xe-0/2/0 {
  unit 0 {
    family inet {
      address 10.10.5.1/30;
    }
    family mpls;
  }
}
```


2. On Router PE3, configure the core-facing interfaces. Include the **family** statement and specify the **inet** address family. Include the **address** statement and specify the IPv4 addresses shown in the example as the interface addresses. Include the **family** statement and specify the **mpls** address family. In the example, the **xe-2/1/0** interface is connected to Router PE5, and the **xe-2/2/0** interface is connected to Router PE2.

```
[edit interfaces]
xe-2/0/0 {
   unit 0 {
      family inet {
         address 10.10.20.2/30;
      }
      family mpls;
   }
}
xe-2/1/0 {
   unit 0 {
      family inet {
         address 10.10.6.1/30;
      }
      family mpls;
   }
}
xe-2/2/0 {
   unit 0 {
      family inet {
         address 10.10.5.2/30;
      }
      family mpls;
   }
}
xe-2/3/0 {
   unit 0 {
      family inet {
         address 10.10.1.2/30;
      }
      family mpls;
   }
}
```

3. On Router PE5, configure the **xe-0/1/0** interface. Include the **family** statement and specify the **inet** address family. Include the **address** statement and specify **10.10.6.2/30** as the interface address. Include the **family** statement and specify the **mpls** address family.

```
[edit interfaces]
xe-0/1/0 {
   unit 0 {
      family inet {
         address 10.10.6.2/30;
      }
      family mpls;
   }
}
```

Configuring Protocols

Step-by-Step Procedure

This procedure describes how to configure the protocols used in this example. If your network contains P routers, configure the interfaces on the P routers also.

1. On Router PE3, enable OSPF as the IGP. Enable the MPLS, LDP, and BGP protocols on all interfaces except **fxp.0**. LDP is used as the signaling protocol for the Layer 2 circuit to Router PE2 . The following configuration snippet shows the protocol configuration for Router PE3:

```
[edit]
protocols {
  rsvp {
    interface all;
    interface fxp0.0 {
      disable;
    }
  }
  mpls {
    label-switched-path to-RR {
      to 7.7.7.7;
    }
    label-switched-path to-PE2 {
      to 2.2.2.2;
    }
    label-switched-path to-PE5 {
      to 5.5.5.5;
    }
    label-switched-path to-PE4 {
      to 4.4.4.4;
    }
    label-switched-path to-PE1 {
      to 1.1.1.1;
    }
    interface all;
    interface fxp0.0 {
      disable;
    }
  }
  bgp {
    group RR {
      type internal;
      local-address 3.3.3.3;
      family inet-vpn {
        unicast;
      }
      family l2vpn {
        signaling;
      }
      neighbor 7.7.7.7;
    }
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
```

```
            interface all;
            interface fxp0.0 {
                disable;
            }
        }
    }
    ldp {
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
}
```

2. On Router PE2, configure the MPLS, OSPF, and LDP protocols.

```
[edit ]
protocols {
    mpls {
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface all;
            interface fxp0.0 {
                disable;
            }
        }
    }
    ldp {
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
}
```

3. On Router PE5, enable OSPF as the IGP. Enable the MPLS, RSVP, and BGP protocols on all interfaces except **fxp.0**. Enable core-facing interfaces with the **mpls** and **inet** address families.

```
[edit]
protocols {
    rsvp {
        interface all {
            link-protection;
        }
        interface fxp0.0 {
            disable;
        }
    }
    mpls {
        label-switched-path to-RR {
```

```
        to 7.7.7.7;
      }
      label-switched-path to-PE2 {
        to 2.2.2.2;
      }
      label-switched-path to-PE3 {
        to 3.3.3.3;
      }
      label-switched-path to-PE4 {
        to 4.4.4.4;
      }
      label-switched-path to-PE1 {
        to 1.1.1.1;
      }
      interface all;
      interface fxp0.0 {
        disable;
      }
    }
    bgp {
      group to-rr {
        type internal;
        local-address 5.5.5.5;
        family inet-vpn {
          unicast;
        }
        family l2vpn {
          signaling;
        }
        neighbor 7.7.7.7;
      }
    }
    ospf {
      traffic-engineering;
      area 0.0.0.0 {
        interface all;
        interface fxp0.0 {
          disable;
        }
      }
    }
  }
```

Configuring Routing Instances and Layer 2 Circuits

Step-by-Step Procedure

This procedure describes how to configure the Layer 2 circuit and the Layer 3 VPN.

1. On Router PE2, configure the Layer 2 circuit. Include the **l2circuit** statement. Include the **neighbor** statement and specify the loopback IPv4 address of Router PE3 as the neighbor. Include the interface statement and specify **ge-1/0/2.0** as the logical interface that is participating in the Layer 2 circuit. Include the **virtual-circuit-id** statement and specify **100** as the identifier. Include the **no-control-word** statement for equipment that does not support the control word.

   ```
   [edit ]
   ```



```
protocols {
  l2circuit {
    neighbor 3.3.3.3 {
      interface ge-1/0/2.0 {
        virtual-circuit-id 100;
        no-control-word;
      }
    }
  }
}
```

2. On Router PE3, configure the Layer 2 circuit to Router PE2. Include the **l2circuit** statement. Include the **neighbor** statement and specify the loopback IPv4 address of Router PE2 as the neighbor. Include the interface statement and specify **lt-1/1/10.0** as the logical tunnel interface that is participating in the Layer 2 circuit. Include the **virtual-circuit-id** statement and specify **100** as the identifier. Include the **no-control-word** statement.

```
[edit ]
protocols {
  l2circuit {
    neighbor 2.2.2.2 {
      interface lt-1/1/10.0 {
        virtual-circuit-id 100;
        no-control-word;
      }
    }
  }
}
```

3. On Router PE3, configure the Layer 3 VPN (**L3VPN**) routing instance to Router PE5 at the **[edit routing-instances]** hierarchy level. Also configure the BGP peer group at the **[edit routing-instances L3VPN protocols]** hierarchy level.

```
[edit ]
routing-instances {
  L3VPN {
    instance-type vrf;
    interface ge-1/0/1.0;
    interface lt-1/1/10.1;
    route-distinguisher 65000:33;
    vrf-target target:65000:2;
    vrf-table-label;
    protocols {
      bgp {
        export direct;
        group ce3 {
          neighbor 90.90.90.2 {
            peer-as 100;
          }
        }
      }
    }
  }
}
```

4. On Router PE5, configure the Layer 3 VPN routing instance (**L3VPN**) at the **[edit routing-instances]** hierarchy level. Also configure the BGP peer group at the **[edit routing-instances L3VPN protocols]** hierarchy level.

```
[edit ]
routing-instances {
  L3VPN {
    instance-type vrf;
    interface ge-2/0/0.0;
    route-distinguisher 65000:5;
    vrf-target target:65000:2;
    vrf-table-label;
    protocols {
      bgp {
        group ce5 {
          neighbor 80.80.80.2 {
            peer-as 200;
          }
        }
      }
    }
  }
}
```

Configuring the Route Reflector

Step-by-Step Procedure

Although a route reflector is not required to interconnect a Layer 2 circuit with a Layer 3 VPN, this examples uses a route reflector. This procedure shows the relevant portion of the route reflector configuration.

1. Configure the route reflector with RSVP, MPLS, BGP and OSPF. The route reflector is a BGP peer with the PE routers. Notice that the BGP peer group configuration includes the **family** statement and specifies the **inet-vpn** option The **inet-vpn** option enables BGP to advertise network layer reachability information (NLRI) for the Layer 3 VPN routes. The configuration also includes the **family** statement and specifies the **l2vpn** option. The **l2vpn** option enables BGP to advertise NLRI for the Layer 2 circuit. Layer 2 circuits use the same internal BGP infrastructure as Layer 2 VPNs.

```
[edit ]
protocols {
  rsvp {
    interface all;
    interface fxp0.0 {
      disable;
    }
  }
  mpls {
    label-switched-path to-pe3 {
      to 3.3.3.3;
    }
    label-switched-path to-pe5 {
      to 5.5.5.5;
    }
    interface all;
```



```
        interface fxp0.0 {
            disable;
        }
    }
    bgp {
        group RR {
            type internal;
            local-address 7.7.7.7;
            family inet {
                unicast;
            }
            family inet-vpn {
                unicast;
            }
            family l2vpn {
                signaling;
            }
            cluster 7.7.7.7;
            neighbor 1.1.1.1;
            neighbor 2.2.2.2;
            neighbor 4.4.4.4;
            neighbor 5.5.5.5;
            neighbor 3.3.3.3;
        }
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface all;
            interface fxp0.0 {
                disable;
            }
        }
    }
}
```

Interconnecting the Layer 2 Circuit with the Layer 3 VPN

Step-by-Step Procedure

Before you can configure the logical tunnel interface in an MX Series router, you must create the tunnel services interface to be used for tunnel services.

1. Create the tunnel service interface on Router PE3. Include the **bandwidth** statement at the **[edit chassis fpc *slot-number* pic *slot-number* tunnel-services]** hierarchy level and specify the amount of bandwidth to reserve for tunnel services in gigabits per second.

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

2. On Router PE3, configure the **lt-1/1/10** logical tunnel interface unit 0.

   Router PE3 is the router that is *stitching* the Layer 2 circuit to the Layer 3 VPN using the logical tunnel interface. The configuration of the peer unit interfaces is what makes the interconnection.

   Include the **encapsulation** statement and specify the **ethernet-ccc** option. Include the **peer-unit** statement and specify the logical interface unit **1** as the peer tunnel interface. Include the **family** statement and specify the **ccc** option.

   Configure the **lt-1/1/10** logical interface unit **1** with **ethernet** encapsulation. Include the **peer-unit** statement and specify the logical interface unit **0** as the peer tunnel interface. Include the **family** statement and specify the **inet** option. Also include the **address** statement and specify **70.70.70.1/24** as the IPv4 address of the interface.

   NOTE: The peering logical interfaces must belong to the same logical tunnel interface derived from the Tunnel Services PIC.

```
[edit interfaces]
lt-1/1/10 {
  unit 0 {
    encapsulation ethernet-ccc;
    peer-unit 1;
    family ccc;
  }
  unit 1 {
    encapsulation ethernet;
    peer-unit 0;
    family inet {
      address 70.70.70.1/24;
    }
  }
}
```

3. On each router, commit the configuration.

```
user@host> commit check
configuration check succeeds
user@host> commit
```

Verifying the Layer 2 Circuit to Layer 3 VPN Interconnection

To verify that the interconnection is working properly, perform these tasks:

- Verifying the Layer 3 VPN Routes on Router PE3 on page 208
- Verifying the Layer 2 Circuit Routes on Router PE3 on page 209
- Verifying the MPLS Routes on Router PE3 on page 209
- Verifying Traffic Flow Between Router CE2 and Router CE3 on page 210
- Verifying Traffic Flow Between Router CE2 and Router CE5 on page 210

Verifying That the Layer 2 Circuit Connection to Router PE3 is Up

Purpose To verify that the Layer 2 circuit connection from Router PE2 to Router PE3 is **Up**. To also document the incoming and outgoing LDP labels and the circuit ID used by this Layer 2 circuit connection.

Action Verify that the Layer 2 circuit connection is up, using the **show l2circuit connections** command.

```
user@PE2> show l2circuit connections

Legend for connection status (St)
EI -- encapsulation invalid      NP -- interface h/w not present
MM -- mtu mismatch               Dn -- down
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
VM -- vlan id mismatch           CF -- Call admission control failure
OL -- no outgoing label          IB -- TDM incompatible bitrate
NC -- intf encaps not CCC/TCC    TM -- TDM misconfiguration
BK -- Backup Connection          ST -- Standby Connection
CB -- rcvd cell-bundle size bad  SP -- Static Pseudowire
LD -- local site signaled down   RS -- remote site standby
RD -- remote site signaled down  XX -- unknown

Legend for interface status
Up -- operational
Dn -- down
Neighbor: 3.3.3.3
    Interface                 Type  St     Time last up          # Up trans
    ge-1/0/2.0(vc 100)        rmt   Up     Jan  7 02:14:13 2010           1
      Remote PE: 3.3.3.3, Negotiated control-word: No
      Incoming label: 301488, Outgoing label: 315264
      Negotiated PW status TLV: No
      Local interface: ge-1/0/2.0, Status: Up, Encapsulation: ETHERNET
```

Meaning The output shows that the Layer 2 circuit connection from Router PE2 to Router PE3 is **Up** and the connection is using the **ge-1/0/2.0** interface. Note that the outgoing label is **315264** and the incoming label is **301488**, the virtual circuit (VC) identifier is **100** and the encapsulation is **ETHERNET**.

Verifying LDP Neighbors and Targeted LDP LSPs on Router PE2

Purpose To verify that Router PE2 has a targeted LDP LSP to Router PE3 and that Router PE2 and Router PE3 are LDP neighbors.

Action Verify that Router PE2 has a targeted LDP LSP to Router PE3 and that Router PE2 and Router PE3 are LDP neighbors, using the **show ldp neighbor** command.

```
user@PE2> show ldp neighbor
Address              Interface            Label space ID          Hold time
3.3.3.3              lo0.0                3.3.3.3:0                 38
```

Meaning The output shows that Router PE2 has an LDP neighbor with the IPv4 address of **3.3.3.3**. Address 3.3.3.3 is the lo0.0 interface address of Router PE3. Notice that Router PE2 uses the local **lo0.0** interface for the LSP.

Verifying that the routers are LDP neighbors also verifies that the targeted LSP is established.

Verifying the Layer 2 Circuit Routes on Router PE2

Purpose To verify that Router PE2 has a route for the Layer 2 circuit and that the route uses the LDP MPLS label to Router PE3.

Action Verify that Router PE2 has a route for the Layer 2 circuit and that the route uses the LDP MPLS label to Router PE3, using the **show route table mpls.0** command.

```
user@PE2> show route table mpls.0
mpls.0: 13 destinations, 13 routes (13 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                   *[MPLS/0] 1w3d 05:24:11, metric 1
                       Receive
1                   *[MPLS/0] 1w3d 05:24:11, metric 1
                       Receive
2                   *[MPLS/0] 1w3d 05:24:11, metric 1
                       Receive
300560              *[LDP/9] 16:12:23, metric 1
                     > to 10.10.2.1 via xe-0/1/0.0, Pop
300560(S=0)         *[LDP/9] 16:12:23, metric 1
                     > to 10.10.2.1 via xe-0/1/0.0, Pop
301008              *[LDP/9] 16:12:23, metric 1
                     > to 10.10.4.2 via xe-0/3/0.0, Swap 299856
301488              *[L2CKT/7] 11:07:28
                     > via ge-1/0/2.0, Pop
301536              *[LDP/9] 16:12:23, metric 1
                     > to 10.10.4.2 via xe-0/3/0.0, Pop
301536(S=0)         *[LDP/9] 16:12:23, metric 1
                     > to 10.10.4.2 via xe-0/3/0.0, Pop
301712              *[LDP/9] 12:41:22, metric 1
                     > to 10.10.5.2 via xe-0/2/0.0, Swap 315184
301728              *[LDP/9] 12:41:22, metric 1
                     > to 10.10.5.2 via xe-0/2/0.0, Pop
301728(S=0)         *[LDP/9] 12:41:22, metric 1
                     > to 10.10.5.2 via xe-0/2/0.0, Pop
ge-1/0/2.0         *[L2CKT/7] 11:07:28, metric2 1
                     > to 10.10.5.2 via xe-0/2/0.0, Push 315264
```

Meaning The output shows that Router PE2 pushes the **315264** outgoing label on the **L2CKT** route going out interface **ge-1/0/2.0**. The output also shows that Router PE2 pops the **301488** incoming label on the **L2CKT** coming from interface **ge-1/0/2.0**

Verifying That the Layer 2 Circuit Connection to Router PE2 is Up

Purpose To verify that the Layer 2 circuit connection from Router PE3 to Router PE2 is **Up**, To also document the incoming and outgoing LDP labels and the circuit ID used by this Layer 2 circuit connection.

Action Verify that the Layer 2 circuit connection is up, using the **show l2circuit connections** command.

```
user@PE3> show l2circuit connections

Layer-2 Circuit Connections:
Legend for connection status (St)
EI -- encapsulation invalid      NP -- interface h/w not present
MM -- mtu mismatch               Dn -- down
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
VM -- vlan id mismatch           CF -- Call admission control failure
OL -- no outgoing label          IB -- TDM incompatible bitrate
NC -- intf encaps not CCC/TCC    TM -- TDM misconfiguration
BK -- Backup Connection          ST -- Standby Connection
CB -- rcvd cell-bundle size bad  XX -- unknown

Legend for interface status
Up -- operational
Dn -- down
Neighbor: 2.2.2.2
    Interface                 Type  St     Time last up          # Up trans
    lt-1/1/10.0(vc 100)       rmt   Up     Jan  7 02:15:03 2010           1
      Remote PE: 2.2.2.2, Negotiated control-word: No
      Incoming label: 315264, Outgoing label: 301488
      Local interface: lt-1/1/10.0, Status: Up, Encapsulation: ETHERNET
```

Meaning The output shows that the Layer 2 circuit connection from Router PE3 to Router PE2 is **Up** and the connection is using the logical tunnel (**lt**) interface. Note that the incoming label is **315264** and the outgoing label is **301488**, the virtual circuit (VC) identifier is **100**, and that the encapsulation is **ETHERNET**.

Verifying LDP Neighbors and Targeted LDP LSPs on Router PE3

Purpose To verify that Router PE3 has a targeted LDP LSP to Router PE2 and that Router PE3 and Router PE2 are LDP neighbors.

Action Verify that Router PE2 has a targeted LDP LSP to Router PE3 and that Router PE2 and Router PE3 are LDP neighbors, using the **show ldp neighbor** command.

```
user@PE2> show ldp neighbor
Address            Interface          Label space ID         Hold time
2.2.2.2            lo0.0              2.2.2.2:0                43
4.4.4.4            lo0.0              4.4.4.4:0                33
```

Meaning The output shows that Router PE3 has an LDP neighbor with the IPv4 address of **2.2.2.2**. Address 2.2.2.2 is the lo0.0 interface address of Router PE2. The output also shows that the interface used on Router PE3 for the LSP is **lo0.0**. Verifying that the routers are LDP neighbors also verifies that the targeted LSP is established.

Verifying a BGP Peer Session with the Route Reflector on Router PE3

Purpose To verify that Router PE3 has a peer session established with the route reflector.

Action Verify that Router PE3 has a peer session established with the route reflector, using the **show bgp summary** command.

```
user@PE2> show bgp summary

Groups: 2 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l3vpn.0            1          1          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
7.7.7.7               65000       1597       1612       0       1    12:03:21 Establ
  bgp.l2vpn.0: 0/0/0/0
  bgp.l3vpn.0: 1/1/1/0
  L3VPN.inet.0: 1/1/1/0
```

Meaning The output shows that Router PE3 has a peer session with the router with the IPv4 address of **7.7.7.7**. Address 7.7.7.7 is the lo0.0 interface address of the route reflector. The output also shows that the peer session state is **Establ**, meaning that the session is established.

Verifying the Layer 3 VPN Routes on Router PE3

Purpose To verify that Router PE3 has Layer 3 VPN routes to Router CE2, Router CE3, and Router CE5.

Action Verify that Router PE3 has routes to Router CE2, Router CE3, and Router CE5 in the Layer 3 VPN route table, using the **show route table L3VPN.inet.0** command. In this example, **L3VPN** is the name configured for the routing instance.

```
user@PE3> show route table L3VPN.inet.0
L3VPN.inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

70.70.70.0/24      *[Direct/0] 11:13:59
                    > via lt-1/1/10.1
70.70.70.1/32      *[Local/0] 11:13:59
                      Local via lt-1/1/10.1
80.80.80.0/24      *[BGP/170] 11:00:41, localpref 100, from 7.7.7.7
                      AS path: I
                    > to 10.10.6.2 via xe-2/1/0.0, Push 16
90.90.90.0/24      *[Direct/0] 11:54:41
                    > via ge-1/0/1.0
90.90.90.1/32      *[Local/0] 11:54:41
                      Local via ge-1/0/1.0
```

Meaning The output shows that Router PE3 has a route to the IPv4 subnetwork address of **70.70.70.0**. Address 70.70.70.2 is the interface address of Router CE2. The output shows that Router PE3 has a route to the IPv4 subnetwork address of **80.80.80.0**. Address 80.80.80.2 is the interface address of Router CE5. The output shows that Router PE3 has a route to the IPv4 subnetwork address of **90.90.90.0**. Address 90.90.90.2 is the interface address of Router CE3.

Verifying the Layer 2 Circuit Routes on Router PE3

Purpose To verify that Router PE3 has a route to Router PE2 in the Layer 2 circuit route table.

Action Verify that Router PE3 has a route to Router PE2 in the Layer 2 circuit route table, using the **show route table l2circuit.0** command.

```
user@PE3> show route table l2circuit.0
2.2.2.2:NoCtrlWord:5:100:Local/96 (1 entry, 1 announced)
        *L2CKT  Preference: 7
                Next hop type: Indirect
                Next-hop reference count: 1
                Next hop type: Router
                Next hop: 10.10.5.1 via xe-2/2/0.0, selected
                Protocol next hop: 2.2.2.2
                Indirect next hop: 8cae0a0 -
                State: <Active Int>
                Local AS: 65000
                Age: 11:16:50   Metric2: 1
                Task: l2 circuit
                Announcement bits (1): 0-LDP
                AS path: I
                VC Label 315264, MTU 1500
```

Meaning The output shows that Router PE3 has a route to the IPv4 address of **2.2.2.2**. Address 2.2.2.2 is the lo0.0 interface address of Router PE2. Note that the VC label is **315264**. This label is the same as the incoming MPLS label displayed using the **show l2circuit connections** command.

Verifying the MPLS Routes on Router PE3

Purpose To verify that Router PE3 has a route to Router PE2 in the MPLS route table.

Action Verify Router PE3 has a route to Router PE2 in the MPLS route table, using the **show route table mpls.0** command.

```
user@PE3> show route table mpls.0
mpls.0: 21 destinations, 21 routes (21 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 1w3d 05:29:02, metric 1
                      Receive
1                  *[MPLS/0] 1w3d 05:29:02, metric 1
                      Receive
2                  *[MPLS/0] 1w3d 05:29:02, metric 1
                      Receive
16                 *[VPN/0] 12:22:45
                      to table L3VPN.inet.0, Pop
315184             *[LDP/9] 12:45:14, metric 1
                    > to 10.10.20.1 via xe-2/0/0.0, Pop
315184(S=0)        *[LDP/9] 12:45:14, metric 1
                    > to 10.10.20.1 via xe-2/0/0.0, Pop
315200             *[LDP/9] 00:03:53, metric 1
                    > to 10.10.20.1 via xe-2/0/0.0, Swap 625297
                      to 10.10.6.2 via xe-2/1/0.0, Swap 299856
315216             *[LDP/9] 12:45:14, metric 1
                    > to 10.10.6.2 via xe-2/1/0.0, Pop
```



```
315216(S=0)        *[LDP/9] 12:45:14, metric 1
                    > to 10.10.6.2 via xe-2/1/0.0, Pop
315232             *[LDP/9] 12:45:06, metric 1
                    > to 10.10.1.1 via xe-2/3/0.0, Pop
315232(S=0)        *[LDP/9] 12:45:06, metric 1
                    > to 10.10.1.1 via xe-2/3/0.0, Pop
315248             *[LDP/9] 12:45:14, metric 1
                    > to 10.10.5.1 via xe-2/2/0.0, Pop
315248(S=0)        *[LDP/9] 12:45:14, metric 1
                    > to 10.10.5.1 via xe-2/2/0.0, Pop
315264             *[L2CKT/7] 11:11:20
                    > via lt-1/1/10.0, Pop
315312             *[RSVP/7] 11:26:01, metric 1
                    > to 10.10.6.2 via xe-2/1/0.0, label-switched-path to-pe5
315312(S=0)        *[RSVP/7] 11:26:01, metric 1
                    > to 10.10.6.2 via xe-2/1/0.0, label-switched-path to-pe5
315328             *[RSVP/7] 11:26:01, metric 1
                    > to 10.10.20.1 via xe-2/0/0.0, label-switched-path to-RR
315360             *[RSVP/7] 11:26:01, metric 1
                    > to 10.10.20.1 via xe-2/0/0.0, label-switched-path to-RR
316208             *[RSVP/7] 00:03:32, metric 1
                    > to 10.10.6.2 via xe-2/1/0.0, label-switched-path
Bypass->10.10.9.1
316208(S=0)        *[RSVP/7] 00:03:32, metric 1
                    > to 10.10.6.2 via xe-2/1/0.0, label-switched-path
Bypass->10.10.9.1
lt-1/1/10.0       *[L2CKT/7] 11:11:20, metric2 1
                    > to 10.10.5.1 via xe-2/2/0.0, Push 301488
```

Meaning The output shows that Router PE3 has a route for the Layer 2 circuit and that the route uses the LDP MPLS label to Router PE2. Notice that the **301488** label is the same as the outgoing label displayed on Router PE2 using the **show l2circuit connections** command.

Verifying Traffic Flow Between Router CE2 and Router CE3

Purpose To verify that the CE routers can send and receive traffic across the interconnection.

Action Verify that Router CE2 can send traffic to and receive traffic from Router CE3 across the interconnection, using the **ping** command.

```
user@CE2>ping 90.90.90.2
PING 90.90.90.2 (90.90.90.2): 56 data bytes
64 bytes from 90.90.90.2: icmp_seq=0 ttl=63 time=0.708 ms
64 bytes from 90.90.90.2: icmp_seq=1 ttl=63 time=0.610 ms
```

Meaning The output shows that Router CE2 can send an ICMP request to and receive a response from Router CE3 across the interconnection.

Verifying Traffic Flow Between Router CE2 and Router CE5

Purpose To verify that the CE routers can send and receive traffic across the interconnection.

Action Verify that Router CE2 can send traffic to and receive traffic from Router CE5 across the interconnection, using the **ping** command.

```
user@CE2>ping 80.80.80.2
```



```
PING 80.80.80.2 (80.80.80.2): 56 data bytes
64 bytes from 80.80.80.2: icmp_seq=0 ttl=62 time=0.995 ms
64 bytes from 80.80.80.2: icmp_seq=1 ttl=62 time=1.005 ms
```

Meaning The output shows that Router CE2 can send an ICMP request to and receive a response from Router CE5 across the interconnection.

Related Documentation

- *Layer 3 VPN Overview*
- Applications for Interconnecting a Layer 2 Circuit with a Layer 3 VPN on page 192

Example: Configuring Pseudowire Redundancy for Mobile Backhaul Scenarios

Understanding Pseudowire Redundancy Mobile Backhaul Scenarios

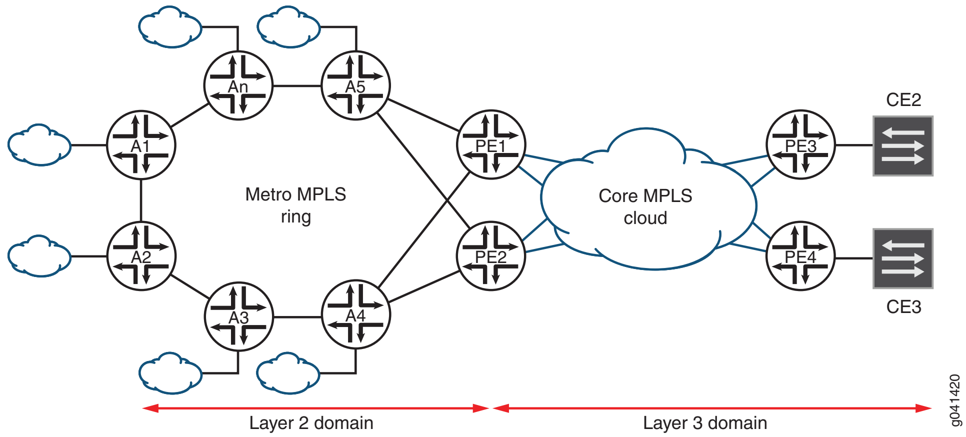
With the rising demand for mobile broadband services, telecommunication providers are seeing a sharp increase in bandwidth requirements. To keep pace with demand, operators are deploying packet-based backhaul networks that offer increased capacity at a lower cost, while providing the necessary service reliability and quality of experience that users expect.

Most of the legacy backhaul infrastructure has been traditionally built over PDH microwave, TDM T1/E1, or ATM-over-DSL links. Service providers have traditionally added subsequent TDM links to their base stations when needed to deal with bandwidth constraint scenarios. This expansion model has proven to be inefficient for the unprecedented traffic demands required by 3G and Long Term Evolution (LTE) services. As a direct consequence, operators are gradually migrating to an Ethernet-based higher capacity infrastructure in the backhaul portion of 3G and LTE topologies. Modern base stations now provide Ethernet backhaul connectivity, allowing pseudowire technologies to transport end-user content to the desired destination. As part of this Ethernet transition, service providers are increasingly demanding better resiliency mechanisms to cover the existence gap with those features provided by previous legacy technologies. With that goal in mind, Junos OS provides efficient pseudowire redundancy capabilities to those topologies where Layer 2 and Layer 3 segments are interconnected.

Sample Topology

Figure 23 on page 212 shows a sample topology.

Figure 23: Pseudowire Redundancy Mobile Backhaul Sample Topology



Benefits of Pseudowire Redundancy Mobile Backhaul

Junos OS pseudowire redundancy capabilities are as follows:

- Redundant loop-free paths to interconnect Layer 2 and Layer 3 domains.
- Layer 2 and Layer 3 domains are synchronized with regard to the elected data path.
- Traffic disruption is minimal for the following possible scenarios:
  - Access link failures
  - Node failures
  - Control-plane failures
- Traffic interruption is minimal after the failure's restoration is completed.

Layer 2 Virtual Circuit Status TLV Extension

The pseudowire status TLV is used to communicate the status of a pseudowire between PE routers. To avoid potential primary-path discrepancies, there must be a mechanism that allows all network elements to be synchronized with respect to the primary path over which traffic needs to be sent. With this goal in mind, the status TLV is extended to address this requirement.

By having the active and standby states being defined by the access routers, Junos OS mitigates potential primary path collisions, as there is a unique network element dictating the preferable forwarding path to be elected. As an added value, this allows network operators to switch forwarding paths on demand, which is quite useful for troubleshooting and network maintenance purposes.

The active and standby states are communicated to the aggregation routers by making use of an additional pseudowire state flag.

Table 5 on page 213 includes a list of the pseudowire state flags.

Table 5: Pseudowire Status Code for the Pseudowire Status TLV

Flag	Code
L2CKT_PW_STATUS_PW_FWD	0x00000000
L2CKT_PW_STATUS_PW_NOT_FWD	0x00000001
L2CKT_PW_STATUS_AC_RX_FAULT	0x00000002
L2CKT_PW_STATUS_AC_TX_FAULT	0x00000004
L2CKT_PW_STATUS_PSN_RX_FAULT	0x00000008
L2CKT_PW_STATUS_PSN_TX_FAULT	0x00000010
L2CKT_PW_STATUS_PW_FWD_STDBY	0x00000020 Indicates the standby state.
L2CKT_PW_STATUS_SWITCH_OVER	0x00000040

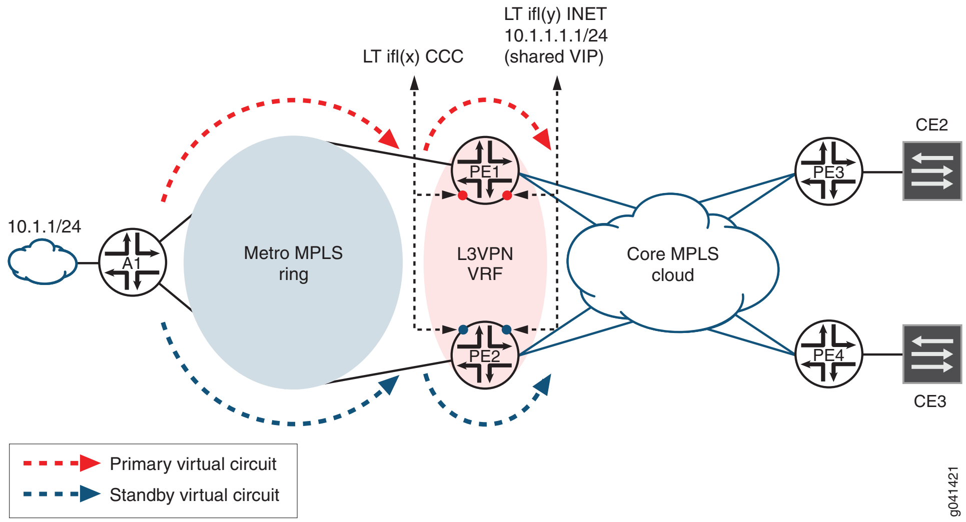
In multichassis LAG (MC-LAG)-based scenarios, this same PW_FWD_STDBY flag is used to advertise to remote PE devices which attachment circuit (AC) is being used as the active one. Upon arrival of this flag, the receiving PE device drops any pseudowire built toward the router originating this state. As we can see, this behavior denotes a slightly different semantic for the PW_FWD_STDBY flag. As a consequence, you can configure the **hot-standby-vc-on** statement to control whether the pseudowire must be constructed upon arrival of the PW_FWD_STDBY flag (in the hot-standby pseudowire scenario), or simply destroy it (in the MC-LAG scenario).

How It Works

The solution uses logical tunnel (lt-) paired interfaces for stitching the Layer 2 and Layer 3 domains.

Figure 24 on page 214 shows a diagram depicting how pseudowire redundancy in a mobile backhaul scenario works.

Figure 24: Pseudowire Redundancy Mobile Backhaul Solution



A Layer 2 pseudowire terminates on one of the logical tunnel interfaces (x), defined with the circuit cross-connect (CCC) address family configured. A Layer 3 VPN (RFC 2547) terminates the second logical tunnel interface (y), defined with the IPv4 (inet) address family. Logical tunnel interface (x) and (y) are paired. Layer 2 pseudowires established between each access router and its corresponding aggregation PE devices terminate on the logical tunnel interface defined within each PE device. This logical tunnel interface is used to establish a Layer 2 virtual circuit (VC) toward the remote end. In consequence, the CCC address family needs to be configured on it. The same applies to the remote end, where an equivalent interface needs to be defined with CCC capabilities.

This CCC logical tunnel interface created in the aggregation PE devices is paired with a second logical tunnel interface on which the INET address family is enabled. This second logical tunnel interface is configured within the context of an RFC 2547 Layer 3 VPN.

Within the scope of this document, we refer to the CCC and INET logical tunnel interfaces as LT(x) and LT(y), respectively.

The Junos OS routing protocol process (rpd) enables the stitching required to interconnect the Layer 2 VC ending in LT(x) and the associated LT(y).

In the aggregation PE routers, the routing process builds a pseudowire toward access routers, and this happens regardless of the active or standby state of the pseudowire. The same occurs in access routers, where the control and forwarding state is preestablished in both the Routing Engine and in the Packet Forwarding Engine to mitigate traffic disruption during convergence periods.

An attachment circuit (AC) is a physical or virtual circuit (VC) that attaches a CE device to a PE device. Local preference is used to provide better information than the multiple exit discriminator (MED) value provides for a packet's path selection. You can configure the local preference attribute so that it has a higher value for prefixes received from a router that provides a desired path than prefixes received from a router that provides a less desirable path. The higher the value, the more preferred the route. The local preference attribute is the metric most often used in practice to express preferences for one set of paths over another.

If the Layer 2 circuit is primary, the corresponding PE device advertises the AC's subnet with the higher local preference. All aggregation PE devices initially advertise the AC's subnet with the same local preference. You can configure a routing policy to allow a higher local preference value to be advertised if the Layer 2 VC is active.

If a pseudowire is down, LT(x) is tagged with the CCC_Down flag. When this happens, the corresponding PE device withdraws the AC subnet that was initially advertised. The LT(y) address is shared between the aggregation PE devices as a virtual instance port (VIP). No VRRP hello messages are exchanged. Both PE devices assume mastership.

Both primary and standby Layer 2 VCs are kept open to reduce traffic disruption in backup-to-primary transitions. The **hot-standby-vc-on** configuration statement allows manual activation.

Resiliency in the Layer 2 domain is provided through plain pseudowire redundancy for back-to-back connections. For other topologies, pseudowire virtual circuit connectivity verification (VCCV) is used.

Resiliency in the Layer 3 domain is provided by MPLS fast reroute and end-to-end service restoration. A restoration timer prevents having VCs in the secondary path from being switched back to the primary path immediately after the master PE device is restored.

Access routers can indicate to the aggregation routers which Layer 2 VC is considered to be active. Upon arrival at LT(x) of a status TLV message communicating a standby state, the routing process decreases the BGP's local preference value of the direct subnet represented by the LT(y) IPv4 address. At this point, BGP proceeds to advertise this local preference change to the rest of the members within the Layer 3 domain, which will then reelect the designated forwarder PE device by relying on BGP's path selection mechanisms.

A similar behavior occurs upon arrival of a status TLV message indicating a Layer 2 VC active state. In this case, the receiving PE device changes the local preference corresponding to the LT(y)'s subnet. The value to be used to either decrease or increase the subnet's local preference value is manually configured using a policy.

Example: Configuring Pseudowire Redundancy in a Mobile Backhaul Scenario

This example shows how to configure pseudowire redundancy where Layer 2 and Layer 3 segments are interconnected in a mobile backhaul scenario.

- Requirements on page 216
- Overview on page 216
- Configuration on page 217
- Verification on page 234

Requirements

This example can be configured using the following hardware and software components:

- Junos OS Release 13.2 or later
- MX Series 3D Universal Edge Routers or M Series Multiservice Edge Routers for the Provider Edge (PE) Routers
- PTX Series Packet Transport Routers acting as transit label-switched routers
- T Series Core Routers for the Core Routers

NOTE: **The PE routers could also be T Series Core Routers but that is not typical. Depending on your scaling requirements, the core routers could also be MX Series 3D Universal Edge Routers or M Series Multiservice Edge Routers. The Customer Edge (CE) devices could be other routers or switches from Juniper Networks or another vendor.**

No special configuration beyond device initialization is required before configuring this example.

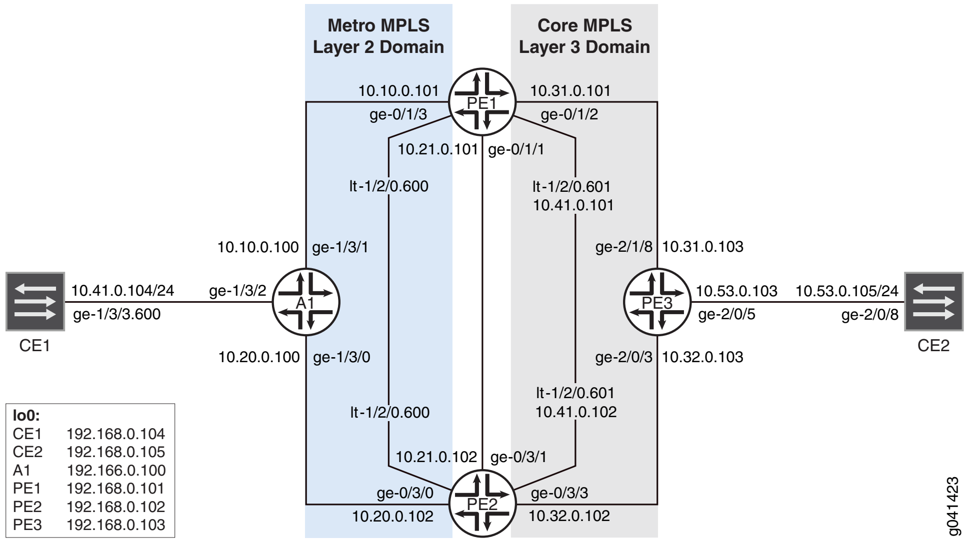
Overview

Device CE1 is a simple edge router with an IPv4 interface and a static route pointing to the PE devices. Device A1 establishes two virtual circuits (VCs) toward Device PE1 and Device PE2 by making use of the **hot-standby** statement. Device PE1 and Device PE2 terminate these VCs and enforce a policy condition over the logical tunnel IPv4 subnet. Device PE3 performs as a Layer 3 VPN provider edge device by having an IPv4 interface in a Layer 3 VPN shared with Device PE1 and Device PE2.

“CLI Quick Configuration” on page 217 shows the configuration for all of the devices in Figure 25 on page 217.

The section “Step-by-Step Procedure” on page 222 describes the steps on Device A1 and Device PE1.

Figure 25: Pseudowire Redundancy in a Mobile Backhaul Example Topology



Configuration

CLI Quick Configuration

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

Device CE1

```
set interfaces ge-1/3/3 vlan-tagging
set interfaces ge-1/3/3 unit 600 vlan-id 600
set interfaces ge-1/3/3 unit 600 family inet address 10.41.0.104/24
set interfaces lo0 unit 0 family inet address 192.168.0.104/32 primary
set interfaces lo0 unit 0 family inet address 192.168.0.104/32 primary
set protocols ospf area 0.0.0.0 interface ge-1/3/3.600
set protocols ospf area 0.0.0.0 interface lo0.0
set routing-options static route 192.168.0.0/8 next-hop 10.41.0.1
set routing-options static route 10.53.0.0/16 next-hop 10.41.0.1
set routing-options router-id 192.168.0.104
```

Device A1

```
set interfaces ge-1/3/0 unit 0 family inet address 10.20.0.100/24
set interfaces ge-1/3/0 unit 0 family iso
set interfaces ge-1/3/0 unit 0 family mpls
set interfaces ge-1/3/1 unit 0 family inet address 10.10.0.100/24
set interfaces ge-1/3/1 unit 0 family iso
set interfaces ge-1/3/1 unit 0 family mpls
set interfaces ge-1/3/2 vlan-tagging
set interfaces ge-1/3/2 encapsulation vlan-ccc
set interfaces ge-1/3/2 unit 600 encapsulation vlan-ccc
set interfaces ge-1/3/2 unit 600 vlan-id 600
set interfaces ge-1/3/2 unit 600 family ccc
set interfaces lo0 unit 0 family inet address 192.168.0.100/32 primary
set interfaces lo0 unit 0 family iso address 49.0002.0192.0168.0100.00
set routing-options router-id 192.168.0.100
set routing-options autonomous-system 64510
```

```
set routing-options forwarding-table export pplb
set protocols rsvp interface ge-1/3/0.0
set protocols rsvp interface ge-1/3/1.0
set protocols rsvp interface lo0.0
set protocols mpls interface ge-1/3/0.0
set protocols mpls interface ge-1/3/1.0
set protocols isis interface ge-1/3/0.0
set protocols isis interface ge-1/3/1.0
set protocols isis interface lo0.0
set protocols ldp interface ge-1/3/0.0
set protocols ldp interface ge-1/3/1.0
set protocols ldp interface lo0.0
set protocols l2circuit neighbor 192.168.0.101 interface ge-1/3/2.600 virtual-circuit-id 1
set protocols l2circuit neighbor 192.168.0.101 interface ge-1/3/2.600 pseudowire-status-tlv
set protocols l2circuit neighbor 192.168.0.101 interface ge-1/3/2.600 revert-time 10
  maximum 60
set protocols l2circuit neighbor 192.168.0.101 interface ge-1/3/2.600 backup-neighbor
  192.168.0.102 virtual-circuit-id 2
set protocols l2circuit neighbor 192.168.0.101 interface ge-1/3/2.600 backup-neighbor
  192.168.0.102 hot-standby
set policy-options policy-statement pplb then load-balance per-packet
```

Device PE1

```
set interfaces ge-0/1/1 unit 0 family inet address 10.21.0.101/24
set interfaces ge-0/1/1 unit 0 family iso
set interfaces ge-0/1/1 unit 0 family mpls
set interfaces ge-0/1/2 unit 0 family inet address 10.31.0.101/24
set interfaces ge-0/1/2 unit 0 family iso
set interfaces ge-0/1/2 unit 0 family mpls
set interfaces ge-0/1/3 unit 0 family inet address 10.10.0.101/24
set interfaces ge-0/1/3 unit 0 family iso
set interfaces ge-0/1/3 unit 0 family mpls
set interfaces lt-1/2/0 unit 600 encapsulation vlan-ccc
set interfaces lt-1/2/0 unit 600 vlan-id 600
set interfaces lt-1/2/0 unit 600 peer-unit 601
set interfaces lt-1/2/0 unit 601 encapsulation vlan
set interfaces lt-1/2/0 unit 601 vlan-id 600
set interfaces lt-1/2/0 unit 601 peer-unit 600
set interfaces lt-1/2/0 unit 601 family inet filter input icmp_inet
set interfaces lt-1/2/0 unit 601 family inet filter output icmp_inet
set interfaces lt-1/2/0 unit 601 family inet address 10.41.0.101/24 vrrp-group 0
  virtual-address 10.41.0.1
set interfaces lt-1/2/0 unit 601 family inet address 10.41.0.101/24 vrrp-group 0 accept-data
set interfaces lo0 unit 0 family inet address 192.168.0.101/32 primary
set interfaces lo0 unit 0 family iso address 49.0002.0192.0168.0003.00
set interfaces lo0 unit 1 family inet address 192.168.1.101/32
set routing-options router-id 192.168.0.101
set routing-options autonomous-system 64511
set protocols rsvp interface ge-0/1/1.0
set protocols rsvp interface ge-0/1/2.0
set protocols rsvp interface ge-0/1/3.0
set protocols rsvp interface lo0.0
set protocols mpls label-switched-path to_PE3 to 192.168.0.103
set protocols mpls label-switched-path to_PE2 to 192.168.0.102
set protocols mpls interface ge-0/1/1.0
set protocols mpls interface ge-0/1/2.0
set protocols mpls interface ge-0/1/3.0
```

```
set protocols bgp local-address 192.168.0.101
set protocols bgp group ibgp family inet-vpn any
set protocols bgp group ibgp peer-as 64511
set protocols bgp group ibgp neighbor 192.168.0.102
set protocols bgp group ibgp neighbor 192.168.0.103
set protocols isis interface ge-0/1/1.0
set protocols isis interface ge-0/1/2.0
set protocols isis interface ge-0/1/3.0
set protocols isis interface lo0.0
set protocols ldp interface ge-0/1/1.0
set protocols ldp interface ge-0/1/2.0
set protocols ldp interface ge-0/1/3.0
set protocols ldp interface lo0.0
set protocols l2circuit neighbor 192.168.0.100 interface lt-1/2/0.600 virtual-circuit-id 1
set protocols l2circuit neighbor 192.168.0.100 interface lt-1/2/0.600 pseudowire-status-tlv
  hot-standby-vc-on
set policy-options policy-statement l3vpn_export term primary from condition primary
set policy-options policy-statement l3vpn_export term primary then local-preference
  add 300
set policy-options policy-statement l3vpn_export term primary then community set l3vpn
set policy-options policy-statement l3vpn_export term primary then accept
set policy-options policy-statement l3vpn_export term standby from condition standby
set policy-options policy-statement l3vpn_export term standby then local-preference
  add 30
set policy-options policy-statement l3vpn_export term standby then community set l3vpn
set policy-options policy-statement l3vpn_export term standby then accept
set policy-options policy-statement l3vpn_export term default then community set l3vpn
set policy-options policy-statement l3vpn_export term default then accept
set policy-options policy-statement l3vpn_import term 1 from community l3vpn
set policy-options policy-statement l3vpn_import term 1 then accept
set policy-options policy-statement l3vpn_import term default then reject
set policy-options policy-statement ospf_export term 0 from community l3vpn
set policy-options policy-statement ospf_export term 0 then accept
set policy-options community l3vpn members target:64511:600
set policy-options condition primary if-route-exists address-family ccc lt-1/2/0.600
set policy-options condition primary if-route-exists address-family ccc table mpls.0
set policy-options condition primary if-route-exists address-family ccc peer-unit 601
set policy-options condition standby if-route-exists address-family ccc lt-1/2/0.600
set policy-options condition standby if-route-exists address-family ccc table mpls.0
set policy-options condition standby if-route-exists address-family ccc standby
set policy-options condition standby if-route-exists address-family ccc peer-unit 601
set firewall family inet filter icmp_inet interface-specific
set firewall family inet filter icmp_inet term 0 from source-address 10.41.0.101/32 except
set firewall family inet filter icmp_inet term 0 from source-address 10.0.0.0/8
set firewall family inet filter icmp_inet term 0 from protocol icmp
set firewall family inet filter icmp_inet term 0 then count icmp_inet
set firewall family inet filter icmp_inet term 0 then log
set firewall family inet filter icmp_inet term 0 then accept
set firewall family inet filter icmp_inet term 1 then accept
set routing-instances l3vpn instance-type vrf
set routing-instances l3vpn interface lt-1/2/0.601
set routing-instances l3vpn interface lo0.1
set routing-instances l3vpn route-distinguisher 192.168.1.101:64511
set routing-instances l3vpn vrf-import l3vpn_import
set routing-instances l3vpn vrf-export l3vpn_export
set routing-instances l3vpn vrf-table-label
```

```
set routing-instances l3vpn protocols ospf export ospf_export
set routing-instances l3vpn protocols ospf area 0.0.0.0 lt-1/2/0.601
set routing-instances l3vpn protocols ospf area 0.0.0.0 lo0.1
```

Device PE2

```
set interfaces ge-0/3/0 unit 0 family inet address 10.20.0.102/24
set interfaces ge-0/3/0 unit 0 family iso
set interfaces ge-0/3/0 unit 0 family mpls
set interfaces ge-0/3/1 unit 0 family inet address 10.21.0.102/24
set interfaces ge-0/3/1 unit 0 family iso
set interfaces ge-0/3/1 unit 0 family mpls
set interfaces ge-0/3/3 unit 0 family inet address 10.32.0.102/24
set interfaces ge-0/3/3 unit 0 family iso
set interfaces ge-0/3/3 unit 0 family mpls
set interfaces lt-1/2/0 unit 600 encapsulation vlan-ccc
set interfaces lt-1/2/0 unit 600 vlan-id 600
set interfaces lt-1/2/0 unit 600 peer-unit 601
set interfaces lt-1/2/0 unit 601 encapsulation vlan
set interfaces lt-1/2/0 unit 601 vlan-id 600
set interfaces lt-1/2/0 unit 601 peer-unit 600
set interfaces lt-1/2/0 unit 601 family inet filter input icmp_inet
set interfaces lt-1/2/0 unit 601 family inet filter output icmp_inet
set interfaces lt-1/2/0 unit 601 family inet address 10.41.0.102/24 vrrp-group 0
  virtual-address 10.41.0.1
set interfaces lt-1/2/0 unit 601 family inet address 10.41.0.102/24 vrrp-group 0 accept-data
set interfaces lo0 unit 0 family inet address 192.168.0.102/32 primary
set interfaces lo0 unit 0 family iso address 49.0002.0192.0168.0102.00
set interfaces lo0 unit 1 family inet address 192.168.1.102/32
set routing-options router-id 192.168.0.102
set routing-options autonomous-system 64511
set protocols rsvp interface ge-0/3/0.0
set protocols rsvp interface ge-0/3/1.0
set protocols rsvp interface ge-0/3/3.0
set protocols rsvp interface lo0.0
set protocols mpls label-switched-path to_PE1 to 192.168.0.101
set protocols mpls label-switched-path to_PE3 to 192.168.0.103
set protocols mpls interface ge-0/3/0.0
set protocols mpls interface ge-0/3/1.0
set protocols mpls interface ge-0/3/3.0
set protocols bgp local-address 192.168.0.102
set protocols bgp group ibgp family inet-vpn any
set protocols bgp group ibgp peer-as 64511
set protocols bgp group ibgp neighbor 192.168.0.101
set protocols bgp group ibgp neighbor 192.168.0.103
set protocols isis interface ge-0/3/0.0
set protocols isis interface ge-0/3/1.0
set protocols isis interface ge-0/3/3.0
set protocols isis interface lo0.0
set protocols ldp interface ge-0/3/0.0
set protocols ldp interface ge-0/3/1.0
set protocols ldp interface ge-0/3/3.0
set protocols ldp interface lo0.0
set protocols l2circuit neighbor 192.168.0.100 interface lt-1/2/0.600 virtual-circuit-id 2
set protocols l2circuit neighbor 192.168.0.100 interface lt-1/2/0.600 pseudowire-status-tlv
  hot-standby-vc-on
set policy-options policy-statement l3vpn_export term primary from condition primary
```

set policy-options policy-statement l3vpn_export term primary then local-preference add 300
set policy-options policy-statement l3vpn_export term primary then community set l3vpn
set policy-options policy-statement l3vpn_export term primary then accept
set policy-options policy-statement l3vpn_export term standby from condition standby
set policy-options policy-statement l3vpn_export term standby then local-preference add 30
set policy-options policy-statement l3vpn_export term standby then community set l3vpn
set policy-options policy-statement l3vpn_export term standby then accept
set policy-options policy-statement l3vpn_export term default then community set l3vpn
set policy-options policy-statement l3vpn_export term default then accept
set policy-options policy-statement l3vpn_import term 1 from community l3vpn
set policy-options policy-statement l3vpn_import term 1 then accept
set policy-options policy-statement l3vpn_import term default then reject
set policy-options policy-statement ospf_export term 0 from community l3vpn
set policy-options policy-statement ospf_export term 0 then accept
set policy-options community l3vpn members target:64511:600
set policy-options condition primary if-route-exists address-family ccc lt-1/2/0.600
set policy-options condition primary if-route-exists address-family ccc table mpls.0
set policy-options condition primary if-route-exists address-family ccc peer-unit 601
set policy-options condition standby if-route-exists address-family ccc lt-1/2/0.600
set policy-options condition standby if-route-exists address-family ccc table mpls.0
set policy-options condition standby if-route-exists address-family ccc standby
set policy-options condition standby if-route-exists address-family ccc peer-unit 601
set firewall family inet filter icmp_inet interface-specific
set firewall family inet filter icmp_inet term 0 from source-address 10.41.0.102/32 except
set firewall family inet filter icmp_inet term 0 from source-address 10.0.0.0/8
set firewall family inet filter icmp_inet term 0 from protocol icmp
set firewall family inet filter icmp_inet term 0 then count icmp_inet
set firewall family inet filter icmp_inet term 0 then log
set firewall family inet filter icmp_inet term 0 then accept
set firewall family inet filter icmp_inet term 1 then accept
set routing-instances l3vpn instance-type vrf
set routing-instances l3vpn interface lt-1/2/0.601
set routing-instances l3vpn interface lo0.1
set routing-instances l3vpn route-distinguisher 192.168.1.102:64511
set routing-instances l3vpn vrf-import l3vpn_import
set routing-instances l3vpn vrf-export l3vpn_export
set routing-instances l3vpn vrf-table-label
set routing-instances l3vpn protocols ospf export ospf_export
set routing-instances l3vpn protocols ospf area 0.0.0.0 interface lt-1/2/0.601
set routing-instances l3vpn protocols ospf area 0.0.0.0 interface lo0.1

Device PE3

set interfaces ge-2/0/3 unit 0 family inet address 10.32.0.103/24
set interfaces ge-2/0/3 unit 0 family iso
set interfaces ge-2/0/3 unit 0 family mpls
set interfaces ge-2/0/5 unit 0 family inet address 10.53.0.103/24
set interfaces ge-2/0/5 unit 0 family mpls
set interfaces ge-2/1/8 unit 0 family inet address 10.31.0.103/24
set interfaces ge-2/1/8 unit 0 family iso
set interfaces ge-2/1/8 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.0.103/32 primary
set interfaces lo0 unit 0 family iso address 49.0002.0192.0168.0103.00
set interfaces lo0 unit 1 family inet address 192.168.1.103/32
set routing-options router-id 192.168.0.103
set routing-options autonomous-system 64511

```
set protocols rsvp interface ge-2/0/3.0
set protocols rsvp interface ge-2/1/8.0
set protocols rsvp interface lo0.0
set protocols mpls label-switched-path to_PE1 to 192.168.0.101
set protocols mpls label-switched-path to_PE2 to 192.168.0.102
set protocols mpls interface ge-2/0/3.0
set protocols mpls interface ge-2/1/8.0
set protocols bgp local-address 192.168.0.103
set protocols bgp group ibgp family inet-vpn any
set protocols bgp group ibgp peer-as 64511
set protocols bgp group ibgp neighbor 192.168.0.101
set protocols bgp group ibgp neighbor 192.168.0.102
set protocols isis interface ge-2/0/3.0
set protocols isis interface ge-2/1/8.0
set protocols isis interface lo0.0
set protocols ldp interface ge-2/0/3.0
set protocols ldp interface ge-2/1/8.0
set protocols ldp interface lo0.0
set policy-options policy-statement l3vpn_ospf_export term 0 from protocol direct
set policy-options policy-statement l3vpn_ospf_export term 0 then accept
set policy-options policy-statement l3vpn_ospf_import term 0 from protocol bgp
set policy-options policy-statement l3vpn_ospf_import term 0 from community l3vpn
set policy-options policy-statement l3vpn_ospf_import term 0 then accept
set policy-options policy-statement ospf_export term 0 from community l3vpn
set policy-options policy-statement ospf_export term 0 then accept
set policy-options community l3vpn members target:64511:600
set routing-instances l3vpn instance-type vrf
set routing-instances l3vpn interface ge-2/0/5.0
set routing-instances l3vpn interface lo0.1
set routing-instances l3vpn route-distinguisher 192.168.0.103:64511
set routing-instances l3vpn vrf-target target:64511:600
set routing-instances l3vpn vrf-table-label
set routing-instances l3vpn protocols ospf export ospf_export
set routing-instances l3vpn protocols ospf area 0.0.0.0 interface ge-2/0/5.0
set routing-instances l3vpn protocols ospf area 0.0.0.0 interface lo0.1
```

Device CE2

```
set interfaces ge-2/0/8 unit 0 family inet address 10.53.0.105/24
set interfaces lo0 unit 0 family inet address 192.168.0.105/32 primary
set protocols ospf area 0.0.0.0 interface ge-2/0/8.0
set protocols ospf area 0.0.0.0 interface lo0.0
set routing-options router-id 192.168.0.105
```

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 Device A1:

1. Configure the interfaces.

   Enable MPLS on the core-facing interfaces. The ISO address family is also enabled, because IS-IS is used as the interior gateway protocol (IGP) in the provider network.

   On the customer-facing interface, you do not need to enable MPLS. On this interface, enable CCC encapsulation and address family CCC.


```
[edit interfaces]
user@A1# set ge-1/3/0 unit 0 family inet address 10.20.0.100/24
user@A1# set ge-1/3/0 unit 0 family iso
user@A1# set ge-1/3/0 unit 0 family mpls

user@A1# set ge-1/3/1 unit 0 family inet address 10.10.0.100/24
user@A1# set ge-1/3/1 unit 0 family iso
user@A1# set ge-1/3/1 unit 0 family mpls

user@A1# set ge-1/3/2 vlan-tagging
user@A1# set ge-1/3/2 encapsulation vlan-ccc
user@A1# set ge-1/3/2 unit 600 encapsulation vlan-ccc
user@A1# set ge-1/3/2 unit 600 vlan-id 600
user@A1# set ge-1/3/2 unit 600 family ccc

user@A1# set lo0 unit 0 family inet address 192.168.0.100/32 primary
user@A1# set lo0 unit 0 family iso address 49.0002.0192.0168.0100.00
```

2. Configure the RSVP on the core-facing interfaces and on the loopback interface.

   RSVP is used in the Layer 3 domain.

   ```
   [edit protocols rsvp]
   user@A1# set interface ge-1/3/0.0
   user@A1# set interface ge-1/3/1.0
   user@A1# set interface lo0.0
   ```

3. Configure LDP on the core-facing interfaces and on the loopback interface.

   LDP is used in Layer 2 domain.

   ```
   [edit protocols ldp]
   user@A1# set interface ge-1/3/0.0
   user@A1# set interface ge-1/3/1.0
   user@A1# set interface lo0.0
   ```

4. Configure MPLS on the core-facing interfaces.

   ```
   [edit protocols mpls]
   user@A1# set interface ge-1/3/0.0
   user@A1# set interface ge-1/3/1.0
   ```

5. Configure an interior gateway protocol, such as IS-IS or OSPF, on the core-facing interfaces and on the loopback interface.

   ```
   [edit protocols isis]
   user@A1# set interface ge-1/3/0.0
   user@A1# set interface ge-1/3/1.0
   user@A1# set interface lo0.0
   ```

6. On the interface that faces the customer edge, configure the Layer 2 circuit.

   Configure the **hot-standby** statement on those routers with both active and standby virtual circuits (VCs) (Device A1 in our topology). You must include the **pseudowire-status-tlv** statement on access routers. Without the status TLV signaling, the standby flag cannot be advertised to remote provider edge (PE) devices.

The **revert-time** statement and the **maximum** option should also be configured on access routers. Without the **revert-time** statement, traffic of all the VCs will not be transitioned to the primary path upon completion of the restoration. If a **revert-time** delay is defined but a **maximum** delay is not, then VCs are restored immediately upon the revert timer's expiration. The maximum option allows the VCs to be restored in a scattered fashion rather than all at once.

```
[edit protocols l2circuit neighbor 192.168.0.101 interface ge-1/3/2.600]
user@A1# set virtual-circuit-id 1
user@A1# set pseudowire-status-tlv
user@A1# set revert-time 10 maximum 60
user@A1# set backup-neighbor 192.168.0.102 virtual-circuit-id 2
user@A1# set backup-neighbor 192.168.0.102 hot-standby
```

7. To have the unilist next hop get pushed to other access routers, configure per-packet load balancing.

```
[edit policy-options policy-statement pplb]
user@A1# set then load-balance per-packet
```

8. Apply the per-packet load balancing policy.

```
[edit routing-options forwarding-table]
user@A1# set export pplb
```

9. Configure the autonomous system (AS) ID and the router ID.

```
[edit routing-options]
user@A1# set router-id 192.168.0.100
user@A1# set autonomous-system 64510
```

Similarly, configure any other access devices.

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 Device PE1:

1. Configure the interfaces.

Enable MPLS on the core-facing interfaces.

```
[edit interfaces]
user@PE1# set ge-0/1/1 unit 0 family inet address 10.21.0.101/24
user@PE1# set ge-0/1/1 unit 0 family iso
user@PE1# set ge-0/1/1 unit 0 family mpls

user@PE1# set ge-0/1/2 unit 0 family inet address 10.31.0.101/24
user@PE1# set ge-0/1/2 unit 0 family iso
user@PE1# set ge-0/1/2 unit 0 family mpls

user@PE1# set ge-0/1/3 unit 0 family inet address 10.10.0.101/24
user@PE1# set ge-0/1/3 unit 0 family iso
user@PE1# set ge-0/1/3 unit 0 family mpls

user@PE1# set lo0 unit 0 family inet address 192.168.0.101/32 primary
```

```
user@PE1# set lo0 unit 0 family iso address 49.0002.0192.0168.0003.00
user@PE1# set lo0 unit 1 family inet address 192.168.1.101/32
```

2. On Device PE1 and Device PE2, which are aggregation routers, configure a pair of logical tunnel interfaces to represent LT(x) and LT(y).

   The solution uses logical tunnel (lt-) paired interfaces for stitching the Layer 2 and Layer 3 domains.

   A Layer 2 pseudowire terminates on one of the logical tunnel interfaces, LT(x), defined with the circuit cross-connect (CCC) address family. A Layer 3 VPN terminates the second logical tunnel interface, LT(y), defined with the IPv4 (inet) address family. LT(x) and LT(y) are paired.

```
[edit interfaces]
user@PE1# set lt-1/2/0 unit 600 encapsulation vlan-ccc
user@PE1# set lt-1/2/0 unit 600 vlan-id 600
user@PE1# set lt-1/2/0 unit 600 peer-unit 601

user@PE1# set lt-1/2/0 unit 601 encapsulation vlan
user@PE1# set lt-1/2/0 unit 601 vlan-id 600
user@PE1# set lt-1/2/0 unit 601 peer-unit 600
user@PE1# set lt-1/2/0 unit 601 family inet filter input icmp_inet
user@PE1# set lt-1/2/0 unit 601 family inet filter output icmp_inet
```

3. (Optional) Associate a unique VRRP address with both Device PE1 and Device PE2.

   In this case, both Device PE1 and Device PE2 assume the mastership state for the defined VIP IPv4 address, so no VRRP hello message are exchanged between the routers.

```
[edit interfaces lt-1/2/0 unit 601 family inet address 10.41.0.101/24]
user@PE1# set vrrp-group 0 virtual-address 10.41.0.1
user@PE1# set vrrp-group 0 accept-data
```

4. Configure IS-IS or another IGP.

```
[edit protocols isis]
user@PE1# set interface ge-0/1/1.0
user@PE1# set interface ge-0/1/2.0
user@PE1# set interface ge-0/1/3.0
user@PE1# set interface lo0.0
```

5. Configure the MPLS on the core-facing interfaces.

```
[edit protocols mpls]
user@PE1# set interface ge-0/1/1.0
user@PE1# set interface ge-0/1/2.0
user@PE1# set interface ge-0/1/3.0
```

6. Configure label-switched paths to the other PE devices.

   BGP is a policy-driven protocol, so also configure and apply any needed routing policies. For example, you might want to export static routes into BGP.

```
[edit protocols mpls]
user@PE1# set label-switched-path to_PE3 to 192.168.0.103
user@PE1# set label-switched-path to_PE2 to 192.168.0.102
```

7. Configure LDP on the core-facing interfaces and on the loopback interface.

```
[edit protocols ldp]
user@PE1# set interface ge-0/1/1.0
user@PE1# set interface ge-0/1/2.0
user@PE1# set interface ge-0/1/3.0
user@PE1# set interface lo0.0
```

8. Configure RSVP on the core-facing interfaces and on the loopback interface.

```
[edit protocols rsvp]
user@PE1# set interface ge-0/1/1.0
user@PE1# set interface ge-0/1/2.0
user@PE1# set interface ge-0/1/3.0
user@PE1# set interface lo0.0
```

9. Configure internal BGP (IBGP).

```
[edit protocols bgp]
user@PE1# set local-address 192.168.0.101
user@PE1# set group ibgp family inet-vpn any
user@PE1# set group ibgp peer-as 64511
user@PE1# set group ibgp neighbor 192.168.0.102
user@PE1# set group ibgp neighbor 192.168.0.103
```

10. Configure the Layer 2 circuit on the logical tunnel interface.

    Configure the **hot-standby-vc-on** statement if you want a hot standby pseudowire to be established upon arrival of PW_FWD_STDBY status TLV.

```
[edit protocols l2circuit neighbor 192.168.0.100 interface lt-1/2/0.600]
user@PE1# set virtual-circuit-id 1
user@PE1# set pseudowire-status-tlv hot-standby-vc-on
```

11. Define a pair of conditions to be applied to the egress policy defined within the Layer 3 VPN instance.

    In both condition **primary** and condition **standby**, the matching route corresponds to the interface lt-1/2/0.600 (y), as this is the format in which egress routes appear in routing table mpls.0 to represent any given pseudowire.

    The difference between these conditions is in the **standby** attribute. Upon arrival of the PW_FWD_STDBY status TLV to Device PE1 or Device PE2, Junos OS matches condition **standby**, and in consequence, only term **standby** within the **l3vpn** policy will be executed. On the other hand, if the PW_FWD_STDBY status TLV is not present, the policy only matches condition **primary**, which then executes term **primary** in the **l3vpn** policy. Also, for logical tunnel-based CCC services, you must specify the logical tunnel interface, LT(y), that is associated with the logical tunnel CCC interface, LT(x). (See "Understanding Pseudowire Redundancy Mobile Backhaul Scenarios" on page 211.)

    Finally, for CCC-based conditions, Junos OS only allows mpls.0 as the matching routing table. For the **address** attribute, Junos OS only allows strings with a logical interface unit format (for example, lt-0/0/0.0).

```
[edit policy-options condition primary if-route-exists address-family ccc]
user@PE1# set lt-1/2/0.600
user@PE1# set table mpls.0
user@PE1# set peer-unit 601
```



```
[edit policy-options condition standby if-route-exists address-family ccc]
user@PE1# set lt-1/2/0.600
user@PE1# set table mpls.0
user@PE1# set standby
user@PE1# set peer-unit 601
```

12. Configure the Layer 3 VPN export policy.

    If the Layer 2 virtual circuit (VC) is primary, the corresponding provider edge (PE) routing device advertises the attachment circuit's (AC's) subnet with the higher local preference. All aggregation PE devices initially advertise the AC's subnet with the same local preference.

    This routing policy allows a higher local preference value to be advertised if the Layer 2 VC is active.

    ```
    [edit policy-options policy-statement l3vpn_export]
    user@PE1# set term primary from condition primary
    user@PE1# set term primary then local-preference add 300
    user@PE1# set term primary then community set l3vpn
    user@PE1# set term primary then accept

    user@PE1# set term standby from condition standby
    user@PE1# set term standby then local-preference add 30
    user@PE1# set term standby then community set l3vpn
    user@PE1# set term standby then accept

    user@PE1# set term default then community set l3vpn
    user@PE1# set term default then accept
    ```

13. Configure the Layer 3 VPN community members.

    ```
    [edit policy-options community l3vpn]
    user@PE1# set members target:64511:600
    ```

14. Configure the Layer 3 VPN import policy, based on the Layer 3 VPN community.

    ```
    [edit policy-options policy-statement l3vpn_import]
    user@PE1# set term 1 from community l3vpn
    user@PE1# set term 1 then accept
    user@PE1# set term default then reject
    ```

15. Configure OSPF export policy, based on the Layer 3 VPN community.

    ```
    [edit policy-options policy-statement ospf_export term 0]
    user@PE1# set from community l3vpn
    user@PE1# set then accept
    ```

16. (Optional) Configure a firewall filter to check the path taken by traffic.

    ```
    [edit firewall family inet filter icmp_inet]
    user@PE1# set interface-specific
    user@PE1# set term 0 from source-address 10.41.0.101/32 except
    user@PE1# set term 0 from source-address 10.0.0.0/8
    user@PE1# set term 0 from protocol icmp
    user@PE1# set term 0 then count icmp_inet
    user@PE1# set term 0 then log
    user@PE1# set term 0 then accept
    user@PE1# set term 1 then accept
    ```

17. Configure the routing instance.

    This routing instance is in the Layer 2 domain where Device PE1 and Device PE2 are interconnected to the metro ring over multiaccess media (Ethernet). You must include the **vrf-table-label**' statement on Device PE1 and Device PE2 to enable advertisement of the direct subnet prefix corresponding to the logical tunnel (lt-) interface toward the Layer 3 domain.

    Device PE1 and Device PE2 use OSPF for Layer 3 VPN communication with Device CE1.

    ```
    [edit routing-instances l3vpn]
    user@PE1# set instance-type vrf
    user@PE1# set interface lt-1/2/0.601
    user@PE1# set interface lo0.1
    user@PE1# set route-distinguisher 192.168.1.101:64511
    user@PE1# set vrf-import l3vpn_import
    user@PE1# set vrf-export l3vpn_export
    user@PE1# set vrf-table-label
    user@PE1# set protocols ospf export ospf_export
    user@PE1# set protocols ospf area 0.0.0.0 interface lt-1/2/0.601
    user@PE1# set protocols ospf area 0.0.0.0 interface lo0.1
    ```

18. Configure the autonomous system (AS) ID and router ID.

    ```
    [edit routing-options]
    user@PE1# set router-id 192.168.0.101
    user@PE1# set autonomous-system 64511
    ```

    Similarly, configure Device PE2.

Results

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

Device A1

```
user@A1# show interfaces
ge-1/3/0 {
   unit 0 {
      family inet {
         address 10.20.0.100/24;
      }
      family iso;
      family mpls;
   }
}
ge-1/3/1 {
   unit 0 {
      family inet {
         address 10.10.0.100/24;
      }
      family iso;
      family mpls;
   }
}
ge-1/3/2 {
```



```
   vlan-tagging;
   encapsulation vlan-ccc;
   unit 600 {
      encapsulation vlan-ccc;
      vlan-id 600;
      family ccc;
   }
}
lo0 {
   unit 0 {
      family inet {
         address 192.168.0.100/32 {
            primary;
         }
      }
      family iso {
         address 49.0002.0192.0168.0100.00;
      }
   }
}
```

```
user@A1# show protocols
rsvp {
   interface ge-1/3/0.0;
   interface ge-1/3/1.0;
   interface lo0.0;
}
mpls {
   interface ge-1/3/0.0;
   interface ge-1/3/1.0;
}
isis {
   interface ge-1/3/0.0;
   interface ge-1/3/1.0;
   interface lo0.0;
}
ldp {
   interface ge-1/3/0.0;
   interface ge-1/3/1.0;
   interface lo0.0;
}
l2circuit {
   neighbor 192.168.0.101 {
      interface ge-1/3/2.600 {
         virtual-circuit-id 1;
         pseudowire-status-tlv;
         backup-neighbor 192.168.0.102 {
            virtual-circuit-id 2;
            hot-standby;
         }
      }
   }
}
```

```
user@A1# show policy-options
policy-statement pplb {
```

```
  then {
    load-balance per-packet;
  }
}
```

```
user@A1# show routing-options
autonomous-system 64510;
router-id 192.168.0.100;
forwarding-table {
  export pplb;
}
```

Device PE1

```
user@PE1# show interfaces
ge-0/1/1 {
  unit 0 {
    family inet {
      address 10.21.0.101/24;
    }
    family iso;
    family mpls;
  }
}
ge-0/1/2 {
  unit 0 {
    family inet {
      address 10.31.0.101/24;
    }
    family iso;
    family mpls;
  }
}
ge-0/1/3 {
  unit 0 {
    family inet {
      address 10.10.0.101/24;
    }
    family iso;
    family mpls;
  }
}
lt-1/2/0 {
  unit 600 {
    encapsulation vlan-ccc;
    vlan-id 600;
    peer-unit 601;
  }
  unit 601 {
    encapsulation vlan;
    vlan-id 600;
    peer-unit 600;
    family inet {
      filter {
        input icmp_inet;
        output icmp_inet;
      }
      address 10.41.0.101/24 {
        vrrp-group 0 {
```



```
                virtual-address 10.41.0.1;
                accept-data;
            }
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 192.168.0.101/32 {
                primary;
            }
        }
        family iso {
            address 49.0002.0192.0168.0003.00;
        }
    }
    unit 1 {
        family inet {
            address 192.168.1.101/32;
        }
    }
}
```

```
user@PE1# show firewall
family inet {
    filter icmp_inet {
        interface-specific;
        term 0 {
            from {
                source-address {
                    10.41.0.101/32 except;
                    10.0.0.0/8;
                }
                protocol icmp;
            }
            then {
                count icmp_inet;
                log;
                accept;
            }
        }
        term 1 {
            then accept;
        }
    }
}
```

```
user@PE1# show protocols
rsvp {
    interface ge-0/1/1.0;
    interface ge-0/1/2.0;
    interface ge-0/1/3.0;
    interface lo0.0;
}
```



```
mpls {
   label-switched-path to_PE3 {
      to 192.168.0.103;
   }
   label-switched-path to_PE2 {
      to 192.168.0.102;
   }
   interface ge-0/1/1.0;
   interface ge-0/1/2.0;
   interface ge-0/1/3.0;
}
bgp {
   local-address 192.168.0.101;
   group ibgp {
      family inet-vpn {
         any;
      }
      peer-as 64511;
      neighbor 192.168.0.102;
      neighbor 192.168.0.103;
   }
}
isis {
   interface ge-0/1/1.0;
   interface ge-0/1/2.0;
   interface ge-0/1/3.0;
   interface lo0.0;
}
ldp {
   interface ge-0/1/1.0;
   interface ge-0/1/2.0;
   interface ge-0/1/3.0;
   interface lo0.0;
}
l2circuit {
   neighbor 192.168.0.100 {
      interface lt-1/2/0.600 {
         virtual-circuit-id 1;
         pseudowire-status-tlv hot-standby-vc-on;
      }
   }
}
```

```
user@PE1# show policy-options
policy-statement l3vpn_export {
   term primary {
      from condition primary;
      then {
         local-preference add 300;
         community set l3vpn;
         accept;
      }
   }
   term standby {
      from condition standby;
      then {
```



```
            local-preference add 30;
            community set l3vpn;
            accept;
        }
    }
    term default {
        then {
            community set l3vpn;
            accept;
        }
    }
}
policy-statement l3vpn_import {
    term 1 {
        from community l3vpn;
        then accept;
    }
    term default {
        then reject;
    }
}
policy-statement ospf_export {
    term 0 {
        from community l3vpn;
        then accept;
    }
}
community l3vpn members target:64511:600;
condition primary {
    if-route-exists {
        address-family {
            ccc {
                lt-1/2/0.600;
                table mpls.0;
                peer-unit 601;
            }
        }
    }
}
condition standby {
    if-route-exists {
        address-family {
            ccc {
                lt-1/2/0.600;
                table mpls.0;
                standby;
                peer-unit 601;
            }
        }
    }
}
```

```
user@PE1# show routing-options
router-id 192.168.0.101;
autonomous-system 64511;
```

```
user@PE1# show routing-instances
```

```
l3vpn {
   instance-type vrf;
   interface lt-1/2/0.601;
   interface lo0.1;
   route-distinguisher 192.168.1.101:64511;
   vrf-import l3vpn_import;
   vrf-export l3vpn_export;
   vrf-table-label;
   protocols {
      ospf {
         export ospf_export;
         area 0.0.0.0 {
            interface lt-1/2/0.601;
            interface lo0.1;
         }
      }
   }
}
```

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

Verification

Confirm that the configuration is working properly.

- Checking Layer 2 Circuits on page 234
- Checking the Policy Conditions on page 236

Checking Layer 2 Circuits

Purpose Upon Layer 2 virtual circuit (VC) establishment, the output of the **show l2circuit connections** command shows the active and the hot-standby VC. In addition, control-plane details are shown for the hot-standby VC.

Action From operational mode, enter the **show l2circuit connections extensive** command.

```
user@A1> show l2circuit connections extensive

Layer-2 Circuit Connections:

Legend for connection status (St)
EI -- encapsulation invalid      NP -- interface h/w not present
MM -- mtu mismatch               Dn -- down
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
VM -- vlan id mismatch           CF -- Call admission control failure
OL -- no outgoing label          IB -- TDM incompatible bitrate
NC -- intf encaps not CCC/TCC    TM -- TDM misconfiguration
BK -- Backup Connection          ST -- Standby Connection
CB -- rcvd cell-bundle size bad  SP -- Static Pseudowire
LD -- local site signaled down   RS -- remote site standby
RD -- remote site signaled down  HS -- Hot-standby Connection
XX -- unknown

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



```
Neighbor: 192.168.0.101
    Interface                 Type  St     Time last up          # Up trans
    ge-1/3/2.600(vc 1)        rmt   Up     Jan 24 11:00:26 2013           1
      Remote PE: 192.168.0.101, Negotiated control-word: Yes (Null)
      Incoming label: 299776, Outgoing label: 299776
      Negotiated PW status TLV: Yes
      local PW status code: 0x00000000, Neighbor PW status code: 0x00000000
      Local interface: ge-1/3/2.600, Status: Up, Encapsulation: VLAN
    Connection History:
        Jan 24 11:00:26 2013  status update timer
        Jan 24 11:00:26 2013  PE route changed
        Jan 24 11:00:26 2013  Out lbl Update                    299776
        Jan 24 11:00:26 2013  In lbl Update                     299776
        Jan 24 11:00:26 2013  loc intf up                 ge-1/3/2.600
Neighbor: 192.168.0.102
    Interface                 Type  St     Time last up          # Up trans
    ge-1/3/2.600(vc 2)        rmt   HS     -----                       ----
      Remote PE: 192.168.0.102, Negotiated control-word: Yes (Null)
      Incoming label: 299792, Outgoing label: 299776
      Negotiated PW status TLV: Yes
      local PW status code: 0x00000020, Neighbor PW status code: 0x00000000
      Local interface: ge-1/3/2.600, Status: Up, Encapsulation: VLAN
```

user@PE1> **show l2circuit connections extensive**

```
Layer-2 Circuit Connections:

Legend for connection status (St)
EI -- encapsulation invalid      NP -- interface h/w not present
MM -- mtu mismatch               Dn -- down
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
VM -- vlan id mismatch           CF -- Call admission control failure
OL -- no outgoing label          IB -- TDM incompatible bitrate
NC -- intf encaps not CCC/TCC    TM -- TDM misconfiguration
BK -- Backup Connection          ST -- Standby Connection
CB -- rcvd cell-bundle size bad  SP -- Static Pseudowire
LD -- local site signaled down   RS -- remote site standby
RD -- remote site signaled down  HS -- Hot-standby Connection
XX -- unknown

Legend for interface status
Up -- operational
Dn -- down
Neighbor: 192.168.0.100
    Interface                 Type  St     Time last up          # Up trans
    lt-1/2/0.600(vc 1)        rmt   Up     Jan 24 11:06:36 2013           1
      Remote PE: 192.168.0.100, Negotiated control-word: Yes (Null)
      Incoming label: 299776, Outgoing label: 299776
      Negotiated PW status TLV: Yes
      local PW status code: 0x00000000, Neighbor PW status code: 0x00000000
      Local interface: lt-1/2/0.600, Status: Up, Encapsulation: VLAN
    Connection History:
        Jan 24 11:06:36 2013  status update timer
        Jan 24 11:06:36 2013  PE route changed
        Jan 24 11:06:36 2013  Out lbl Update                    299776
        Jan 24 11:06:36 2013  In lbl Update                     299776
        Jan 24 11:06:36 2013  loc intf up                 lt-1/2/0.600
```

user@PE2> **show l2circuit connections extensive**

```
Layer-2 Circuit Connections:
```

```
Legend for connection status (St)
EI -- encapsulation invalid      NP -- interface h/w not present
MM -- mtu mismatch               Dn -- down
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
VM -- vlan id mismatch           CF -- Call admission control failure
OL -- no outgoing label          IB -- TDM incompatible bitrate
NC -- intf encaps not CCC/TCC    TM -- TDM misconfiguration
BK -- Backup Connection          ST -- Standby Connection
CB -- rcvd cell-bundle size bad  SP -- Static Pseudowire
LD -- local site signaled down   RS -- remote site standby
RD -- remote site signaled down  HS -- Hot-standby Connection
XX -- unknown

Legend for interface status
Up -- operational
Dn -- down
Neighbor: 192.168.0.100
    Interface                 Type  St     Time last up          # Up trans
    lt-1/2/0.600(vc 2)        rmt   Up     Jan 24 10:55:31 2013           1
      Remote PE: 192.168.0.100, Negotiated control-word: Yes (Null)
      Incoming label: 299776, Outgoing label: 299792
      Negotiated PW status TLV: Yes
      local PW status code: 0x00000000, Neighbor PW status code: 0x00000020
      Local interface: lt-1/2/0.600, Status: Up, Encapsulation: VLAN
    Connection History:
        Jan 24 10:55:31 2013  status update timer
        Jan 24 10:55:31 2013  PE route changed
        Jan 24 10:55:31 2013  Out lbl Update                    299792
        Jan 24 10:55:31 2013  In lbl Update                     299776
        Jan 24 10:55:31 2013  loc intf up                 lt-1/2/0.600
```

Meaning From the perspective of Device PE1 and Device PE2, a single Layer 2 circuit is established toward access routers, so there is no standby device information in the CLI output of the **show l2circuit connections** command. Note that no timing and flapping information is provided for the VC acting as the hot-standby. Junos OS only allows these counters to be tracked for the active VC.

Checking the Policy Conditions

Purpose On the PE devices, verify the state of the different conditions defined as part of the Layer3 VPN's egress policy, where 10.41.0.0/24 corresponds to the logical tunnel (y) subnet.

Action From operational mode, enter the **show policy conditions detail** command.

```
user@PE1> show policy conditions detail
Configured conditions:
Condition primary (static), event: Existence of a route in a specific routing
table
Dependent routes:
 10.41.0.0/24, generation 8
 192.168.0.104/32, generation 8

Condition standby (static), event: Existence of a route in a specific routing
table
Dependent routes:
None
```



```
Condition tables:
Table mpls.0, generation 0, dependencies 0, If-route-exists conditions: primary
(static) standby (static)
Table l3vpn.inet.0, generation 12, dependencies 2
```

```
user@PE2> show policy conditions detail
Configured conditions:
Condition primary (static), event: Existence of a route in a specific routing
table
Dependent routes:
 10.41.0.0/24, generation 18

Condition standby (static), event: Existence of a route in a specific routing
table
Dependent routes:
 10.41.0.0/24, generation 18

Condition tables:
Table mpls.0, generation 0, dependencies 0, If-route-exists conditions: primary
(static) standby (static)
Table l3vpn.inet.0, generation 367, dependencies 2
```

CHAPTER 6

Layer 2 Circuit Configuration Statements

auto-discovery-only

Syntax auto-discovery-only;

Hierarchy Level [edit logical-systems *logical-system-name* protocols bgp family l2vpn],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* family l2vpn],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address* family l2vpn],
[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols bgp family l2vpn],
[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols bgp group *group-name* family l2vpn],
[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols bgp group *group-name* neighbor *address* family l2vpn],
[edit protocols bgp family l2vpn],
[edit protocols bgp group *group-name* family l2vpn],
[edit protocols bgp group *group-name* neighbor *address* family l2vpn],
[edit routing-instances *instance-name* protocols bgp family l2vpn],
[edit routing-instances *instance-name* protocols bgp group *group-name* family l2vpn],
[edit routing-instances *instance-name* protocols bgp group *group-name* neighbor *address* family l2vpn]

Release Information Statement introduced in Junos OS Release 10.4R2.

Description Enable the router to process only the autodiscovery network layer reachability information (NLRI) update messages for VPWS and LDP-based Layer 2 VPN and VPLS update messages (BGP_L2VPN_AD_NLRI) (FEC 129).

Specifically, the **auto-discovery-only** statement notifies the routing process (rpd) to expect autodiscovery-related NLRI messages so that information can be deciphered and used by LDP, VPLS, and VPWS.

The **auto-discovery-only** statement must be configured on all provider edge (PE) routers in a VPLS or in a VPWS. If you configure route reflection, the **auto-discovery-only** statement is also required on provider (P) routers that act as the route reflector in supporting FEC 129-related updates.

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*

bandwidth (Protocols Layer 2 Circuit)

Syntax bandwidth (*bandwidth* | ct*number bandwidth*);

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

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify bandwidth allocation for a Layer 2 circuit or for the class types of a Layer 2 circuit.

Options *bandwidth*—Configure the bandwidth in bits per second for the Layer 2 circuit. You cannot configure the bandwidth for the Layer 2 circuit and for the class types at the same time.

ct*number bandwidth*—Configure the bandwidth in bits per second for a class type on the Layer 2 circuit. You can configure bandwidth for up to four class types (**ct0**, **ct1**, **ct2**, **ct3**) per Layer 2 circuit. If you configure the class types, you must configure them in order, starting with class type **ct0**.

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

Related Documentation
- Configuring Bandwidth Allocation and Call Admission Control in Layer 2 Circuits on page 63

backup-interface (Layer 2 Circuits)

Syntax backup-interface *interface-name*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols l2circuit local-switching interface *interface-name* end-interface interface *interface-name*],
[edit protocols l2circuit local-switching interface *interface-name* end-interface interface *interface-name*]

Release Information Statement introduced in Junos OS Release 12.3.

Description Specify the interface to be used by the protection pseduowire in connection protection configurations for Layer 2 circuits.

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

Related Documentation
- Example: Configuring Layer 2 Circuit Switching Protection on page 144

backup-neighbor

Syntax

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

Hierarchy Level

[edit logical-systems *logical-system-name* protocols l2circuit local-switching interface *interface-name*],
[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls neighbor *address*],
[edit protocols l2circuit local-switching interface *interface-name*],
[edit protocols l2circuit neighbor *address* interface *interface-name*],
[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 12.2 for ACX Series routers.
Statement introduced in Junos OS Release 12.3 at the **[edit protocols l2circuit local-switching interface *interface-name*]** hierarchy level.

Description

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

NOTE: VPLS is not supported on ACX Series routers. The **psn-tunnel-endpoint** statement is not supported at the **[edit protocols l2circuit local-switching interface *interface-name* end-interface interface *interface-name*]** hierarchy level.

Options

address—Specifies the address for the backup neighbor.

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 Redundant Pseudowires for Layer 2 Circuits and VPLS*
- Example: Configuring Layer 2 Circuit Switching Protection on page 144
- community (Protocols Layer 2 Circuit) on page 246

- psn-tunnel-endpoint on page 291
- virtual-circuit-id on page 303

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*

community (Protocols Layer 2 Circuit)

Syntax

```
community community-name {
  invert-match;
  members community-members;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* policy-options],
[edit logical-systems *logical-system-name* protocols l2circuit local-switching interface *interface-name* backup-neighbor *address*],
[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*],
[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name* backup-neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls neighbor *address* backup-neighbor *address*],
[edit policy-options],
[edit protocols l2circuit local-switching interface *interface-name* backup-neighbor *address*],
[edit protocols l2circuit neighbor *address* interface *interface-name*],
[edit protocols l2circuit neighbor *address* interface *interface-name* backup-neighbor *address*],
[edit routing-instances *routing-instance-name* protocols vpls neighbor *address* backup-neighbor *address*]

Release Information

Statement introduced before Junos OS Release 7.4.
Hierarchy levels associated with the **backup-neighbor** statement (pseudowire redundancy) added in Junos OS Release 9.2.
Statement introduced in Junos OS Release 12.3 at the **[edit protocols l2circuit local-switching interface *interface-name* backup-neighbor *address*]** hierarchy level.

Description

Specify the community for the Layer 2 circuit.

Options

community-name—Name of the Layer 2 circuit community.

invert-match—Invert the results of the community expression match.

members *community-members*—Specify the members of the community.

Required Privilege Level

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

Related Documentation

- Configuring the Layer 2 Circuit Community on page 57
- *Configuring Redundant Pseudowires for Layer 2 Circuits and VPLS*
- Example: Configuring Layer 2 Circuit Switching Protection on page 144

connection-protection

Syntax connection-protection;

Hierarchy Level [edit logical-systems *logical-system-name* protocols l2circuit local-switching interface *interface-name*],
[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface],
[edit protocols l2circuit local-switching interface *interface-name*],
[edit protocols l2circuit neighbor *address* interface *interface-name*]

Release Information Statement introduced in Junos OS Release 12.3.

Description Enable connection protection on the Layer 2 circuit. Connection protection enables you to configure a redundant pseudowire to act as a backup connection and can be configured between PE routers, maintaining Layer 2 circuit and VPLS services after certain types of failures. This feature helps to improve the reliability of networks where a single point of failure could interrupt service for customers.

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

Related Documentation
- Example: Configuring Layer 2 Circuit Switching Protection on page 144

control-word (Protocols Layer 2 Circuit Neighbor)

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

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

Release Information Statement introduced before Junos OS Release 7.4.

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

Options **control-word**—Enable the use of the control word.

Default: A null control word is enabled by default. You can also configure the control word explicitly using the **control-word** statement.

no-control-word—Disable the use of the control word.

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

Related Documentation
- Configuring the Control Word for Frame Relay Interfaces on page 49

description (Protocols Layer 2 Circuit Neighbor)

Syntax description *text*;

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

Release Information Statement introduced before Junos OS Release 7.4.

Description Provide a text description for the Layer 2 circuit. If the text includes one or more spaces, enclose the entire text string in quotation marks (" ").

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

Related Documentation

- Configuring the Description on page 34

egress-protection (Layer 2 circuit)

Syntax

```
egress-protection {
  protected-l2circuit {
    egress-pe address;
    ingress-pe address;
    virtual-circuit-id identifier;
  }
  protector-interface interface-name;
  protector-pe address {
    context-identifier identifier;
    lsp lsp-name;
  }
}
```

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

Release Information Statement introduced in Junos OS release 10.4.

Description Configures an egress protection virtual circuit (EPVC).

Options The other 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 an Egress Protection LSP for a Layer 2 Circuit on page 156

egress-protection (MPLS)

Syntax

```
egress-protection {
  context-identifier context-id {
    primary | protector;
    metric igp-metric-value;
    advertise-mode (stub-alias | stub-proxy);
  }
}
```

Hierarchy Level

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

Release Information

Statement introduced in Junos OS Release 10.4.
Options **primary**, **protector**, and **metric** introduced in Junos OS Release 11.4R3.
Option **advertise-mode** introduced in Junos OS Release 13.3.

Description

Enables an Edge Protection Virtual Circuit (EPVC) for the MPLS protocol.

Options

context-identifier *context-id-ip-address*—(Optional) The context identifier IPv4 address.

metric *igp-metric-value*—(Optional) The IGP metric value ranging from **2** through **16777215**.

(primary | protector)—On the primary PE router, configure as type **primary**. On the protector PE router, configure as type **protector**.

Required Privilege Level

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

Related Documentation

- *Example: Configuring Egress Protection for Layer 3 VPN Services*
- *Example: Configuring Layer 3 VPN Egress Protection with RSVP and LDP*

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

atm-pvc—Use ATM PVC encapsulation.

cisco-hdlc—Use Cisco-compatible High-Level Data Link Control (HDLC) framing.

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

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

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. 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.

frame-relay-ccc—Use Frame Relay encapsulation on CCC circuits.

frame-relay-ether-type—Use Frame Relay ether type encapsulation for compatibility with the Cisco Frame Relay.

frame-relay-ether-type-tcc—Use Frame Relay ether type TCC for Cisco-compatible Frame Relay on TCC circuits to connect different media.

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. When you use this encapsulation type, you can configure the **ccc** family only.

frame-relay-tcc—Use Frame Relay encapsulation on TCC circuits to connect different media.

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.

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-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 Encapsulation*
- *Configuring Extended VLAN Encapsulation*
- *Configuring Encapsulation for Layer 2 Wholesale VLAN Interfaces*
- Configuring Interfaces for Layer 2 Circuits on page 47
- *Configuring Interface Encapsulation on PTX Series Packet Transport Routers*
- *Configuring an MPLS-Based Layer 2 VPN (CLI Procedure)*
- *Configuring MPLS LSP Tunnel Cross-Connects Using CCC*
- *Configuring TCC*
- *Configuring VPLS Interface Encapsulation*
- *Configuring Interfaces for VPLS Routing*
- *Defining the Encapsulation for Switching Cross-Connects*
- *Configuring Q-in-Q Tunneling (CLI Procedure)*

encapsulation-type (Layer 2 Circuits)

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 local-switching interface *interface-name*],
[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*],
[edit protocols l2circuit local-switching interface *interface-name*],
[edit protocols l2circuit neighbor *address* interface *interface-name*]

Release Information Statement introduced in Junos OS Release 9.2.

Description Specify the type of Layer 2 traffic transiting the Layer 2 circuit.

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 circuit

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

ppp—PPP

satsop-e1—SATSOP-E1-based Layer 2 circuit

satsop-e3—SATSOP-E3-based Layer 2 circuit

satsop-t1—SATSOP-T1-based Layer 2 circuit

satsop-t3—SATSOP-T3-based Layer 2 circuit

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

Related Documentation

- Configuring the Encapsulation Type for the Layer 2 Circuit Neighbor Interface on page 50

end-interface

Syntax

```
end-interface {
  interface interface-name;
  no-revert;
  protect-interface interface-name;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols l2circuit local-switching interface *interface-name*],
[edit protocols l2circuit local-switching interface *interface-name*]

Release Information

Statement introduced before Junos OS Release 7.4.

Description

Specify the end interface for a local interface switch.

NOTE: **The protect interface must be configured prior to configuring the no-revert statement.**

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 Local Interface Switching in Layer 2 Circuits on page 45

fast-aps-switch

Syntax fast-aps-switch;

Hierarchy Level [edit interfaces *interface-name* sonet-options aps]

Release Information Statement introduced in Junos OS Release 12.1.

Description (M320 routers with Channelized OC3/STM1 Circuit Emulation PIC with SFP only and EX Series switches) Reduce the Automatic Protection Switching (APS) switchover time in Layer 2 circuits.

NOTE:

- Configuring this statement reduces the APS switchover time only when the Layer 2 circuit encapsulation type for the interface receiving traffic from a Layer 2 circuit neighbor is SAToP.
- When the **fast-aps-switch** statement is configured in revertive APS mode, you must configure an appropriate value for revert time to achieve reduction in APS switchover time.
- To prevent the logical interfaces in the data path from being shut down, configure appropriate hold-time values on all the interfaces in the data path that support TDM.
- The **fast-aps-switch** statement cannot be configured when the APS **annex-b** option is configured.
- The interfaces that have the **fast-aps-switch** statement configured cannot be used in virtual private LAN service (VPLS) environments.

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

Related Documentation

- Reducing APS Switchover Time in Layer 2 Circuits on page 64

flow-label-receive

Syntax flow-label-receive;

Hierarchy Level [edit protocols l2circuit neighbor *neighbor-id* interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols l2vpn site name],
[edit routing-instances *routing-instance-name* protocols l2vpn site name interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols vpls]

Release Information Statement introduced in Junos OS Release 14.1.

Description Configure the router to signal the capability to pop the flow label in the receive direction to the remote provider edge (PE) router.

Flow-aware transport of pseudowires (FAT) flow labels enable load-balancing of MPLS packets across equal-cost multipath (ECMP) paths or link aggregation groups (LAGs) without the need for deep packet inspection of the payload. FAT flow labels can be used for LDP-signaled forwarding equivalence class (FEC) 128 and FEC 129 pseudowires for virtual private LAN service (VPLS) and virtual private wire service (VPWS) networks.

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

Related Documentation

- Configuring the FAT Flow Label for FEC 128 VPWS Pseudowires for Load-Balancing MPLS Traffic on page 67
- *Configuring the FAT Flow Label for FEC 128 VPLS Pseudowires for Load-Balancing MPLS Traffic*
- Configuring the FAT Flow Label for FEC 129 VPWS Pseudowires for Load-Balancing MPLS Traffic on page 69
- *Configuring the FAT Flow Label for FEC 129 VPLS Pseudowires for Load-Balancing MPLS Traffic*

flow-label-receive-static

Syntax flow-label-receive-static;

Hierarchy Level [edit protocols l2circuit neighbor *neighbor-id* interface *interface-name*]

Release Information Statement introduced in Junos OS Release 14.1.

Description Configure the FEC 128 VPWS pseudowire to statically push the flow label on the pseudowire packets sent to the remote egress provider edge (PE) router. The ingress PE router inserts the flow label in the pseudowire packet, irrespective of the information exchanged in the signaling plane. If the egress PE router cannot handle the pseudowire packet marked with the flow label, the packet is dropped.

Flow-aware transport of pseudowires (FAT) flow labels enable load-balancing of MPLS packets across equal-cost multipath (ECMP) paths or link aggregation groups (LAGs) without the need for deep packet inspection of the payload. FAT flow labels can be used for LDP-signaled forwarding equivalence class (FEC) 128 and FEC 129 pseudowires for virtual private LAN service (VPLS) and virtual private wire service (VPWS) networks.

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

Related Documentation

- Configuring the FAT Flow Label for FEC 128 VPWS Pseudowires for Load-Balancing MPLS Traffic on page 67

flow-label-transmit

Syntax

flow-label-transmit;

Hierarchy Level

[edit protocols l2circuit neighbor *neighbor-id* interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols l2vpn site name],
[edit routing-instances *routing-instance-name* protocols l2vpn site name interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols vpls]

Release Information

Statement introduced in Junos OS Release 14.1.

Description

Configure the router to signal the capability to push the flow label in the transmit direction to the provider edge (PE) router.

Flow-aware transport of pseudowires (FAT) flow labels enable load-balancing of MPLS packets across equal-cost multipath (ECMP) paths or link aggregation groups (LAGs) without the need for deep packet inspection of the payload. FAT flow labels can be used for LDP-signaled forwarding equivalence class (FEC) 128 and FEC 129 pseudowires for virtual private LAN service (VPLS) and virtual private wire service (VPWS) networks.

Required Privilege Level

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

Related Documentation

- Configuring the FAT Flow Label for FEC 128 VPWS Pseudowires for Load-Balancing MPLS Traffic on page 67
- *Configuring the FAT Flow Label for FEC 128 VPLS Pseudowires for Load-Balancing MPLS Traffic*
- Configuring the FAT Flow Label for FEC 129 VPWS Pseudowires for Load-Balancing MPLS Traffic on page 69
- *Configuring the FAT Flow Label for FEC 129 VPLS Pseudowires for Load-Balancing MPLS Traffic*

flow-label-transmit-static

Syntax flow-label-transmit-static;

Hierarchy Level [edit protocols l2circuit neighbor *neighbor-id* interface *interface-name*]

Release Information Statement introduced in Junos OS Release 14.1.

Description Configure the router to statically pop the flow label on the pseudowire packets received from the remote egress provider (PE) router. If the incoming pseudowire packet is not marked with the flow label, the packet is dropped by the egress PE router.

Flow-aware transport of pseudowires (FAT) flow labels enable load-balancing of MPLS packets across equal-cost multipath (ECMP) paths or link aggregation groups (LAGs) without the need for deep packet inspection of the payload. FAT flow labels can be used for LDP-signaled forwarding equivalence class (FEC) 128 and FEC 129 pseudowires for virtual private LAN service (VPLS) and virtual private wire service (VPWS) networks.

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

Related Documentation

- Configuring the FAT Flow Label for FEC 128 VPWS Pseudowires for Load-Balancing MPLS Traffic on page 67

hot-standby (Protocols Layer 2 Circuit)

Syntax

hot-standby;

Hierarchy Level

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

Release Information

Statement introduced before Junos OS Release 7.4.
Hierarchy levels associated with the **backup-neighbor** statement added in Junos OS Release 9.2.

Description

Configure the pseudowire to the specified backup neighbor as the hot-standby. When you configure this statement, traffic flows over both the active and hot-standby pseudowires to the backup device (either a CE device or PE router). The backup device drops the traffic from the hot-standby pseudowire, unless the active pseudowire fails. If the active pseudowire fails, the backup device automatically switches to the hot-standby pseudowire.

Configure the **hot-standby** statement on routers that have both active and standby virtual circuits. Generally, these are access routers. On provider edge routers, configure the **hot-standby-vc-on**' statement to indicate that a hot-standby pseudowire is desired upon arrival of a PW_FWD_STDBY status-TLV.

Required Privilege Level

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

Related Documentation

- Configuring Layer 2 Circuits over Both RSVP and LDP LSPs on page 52
- *Configuring Redundant Pseudowires for Layer 2 Circuits and VPLS*
- Example: Configuring Layer 2 Circuit Switching Protection on page 144

hot-standby-vc-on (Protocols Layer 2 Circuit)

Syntax hot-standby-vc-on;

Hierarchy Level [edit logical-systems *logical-system-name* protocols l2circuit neighbor *neighbor-id* interface *interface-name* pseudowire-status-tlv],
[edit protocols l2circuit neighbor *neighbor-id* interface *interface-name* pseudowire-status-tlv]

Release Information Statement introduced in Junos OS Release 13.2.

Description On provider edge (PE) aggregation routers, configure the **hot-standby-vc-on** statement to indicate that a hot-standby pseudowire is desired upon arrival of a PW_FWD_STDBY status-tlv. This flag indicates the standby state. Configure the **hot-standby** statement on routers that have both active and standby virtual circuits. Generally, these are access routers.

The goal of the **hot-standby** statement is to reduce the amount of traffic being discarded during primary-to-backup transition periods. This statement enables the possibility of keeping both the active and standby paths open within the Layer 2 domain. By having both the active and standby VCs able to send and receive traffic, traffic loops could potentially occur within the Layer 2 domain. In consequence, Layer 2 VCs are kept open only in the PE-to-access direction. In other words, aggregation PE devices can send traffic toward access devices, but access devices send traffic exclusively through the active VC.

In this regard, the **hot-standby** statement is quite similar to the **standby** statement. The **hot-standby** statement allows for a faster forwarding-path switchover during transition periods, as compared to what is allowed by the **standby** statement.

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

Related Documentation

- Example: Configuring Pseudowire Redundancy for Mobile Backhaul Scenarios on page 211

ignore-encapsulation-mismatch

Syntax ignore-encapsulation-mismatch;

Hierarchy Level [edit logical-systems *logical-system-name* protocols l2circuit local-switching interface *interface-name*],
[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name* neighbor *neighbor-id*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls neighbor *neighbor-id*],
[edit protocols l2circuit local-switching interface *interface-name*],
[edit protocols l2circuit neighbor *address* interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols evpn interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name* neighbor *neighbor-id*],
[edit routing-instances *routing-instance-name* protocols vpls neighbor *neighbor-id*]

Release Information Statement introduced in Junos OS Release 9.2.
Statement extended to support local switching in Junos OS Release 10.4.
Statement introduced for EVPNs in Junos OS Release 13.2 for MX 3D Series.

Description Allow a Layer 2 circuit, VPLS, or EVPN to be established even though the encapsulation configured on the CE device interface does not match the encapsulation configured on the Layer 2 circuit, VPLS, or EVPN interface.

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*
- Enabling the Layer 2 Circuit When the Encapsulation Does Not Match on page 50

ignore-mtu-mismatch

Syntax ignore-mtu-mismatch;

Hierarchy Level [edit logical-systems *logical-system-name* protocols l2circuit local-switching interface *interface-name*],
[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn interface *interface-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls],
[edit protocols l2circuit local-switching interface *interface-name*],
[edit protocols l2circuit neighbor *address* interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols l2vpn interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols vpls]

Release Information Statement introduced in Junos OS Release 8.5.
Support for remote PE routers added in Junos OS Release 9.2.
Support for Layer 2 VPNs and VPLS added in Junos OS Release 10.4.

Description Ignore the MTU configuration set for the physical interface associated with the local switching interface or with the remote PE router. This allows a pseudowire to be brought up between two logical interfaces that are defined on physical interfaces with different MTU values.

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

Related Documentation

- Enabling Local Interface Switching When the MTU Does Not Match on page 46
- Configuring the MTU for Layer 2 Interfaces on page 55

install-nexthop

Syntax install-nexthop (except | lsp *lsp-name* | lsp-regex *lsp-regular-expression*);

Hierarchy Level [edit logical-systems *logical-system-name* policy-options policy-statement *policy-name* term *term-name* then],
[edit policy-options policy-statement *policy-name* term *term-name* then]

Release Information Statement introduced before Junos OS Release 7.4.

Description Select a specific label-switched path (LSP), or select an LSP from a set of similarly named LSPs as the traffic destination for the configured community. Also can prevent the installation of any matching next hops.

Options **except**—Prevent the installation of any matching next hops.

lsp ***lsp-name***—Configure a specific LSP.

lsp-regex ***lsp-regular-expression***—Configure a range of similarly named LSPs. You can use the following wildcard characters when configuring an LSP regular expression:

- Asterisk (*)—Match any characters.
- Period (.)—Match any single digit.

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

Related Documentation

- Configuring the Policy Statement for the Layer 2 Circuit Community on page 58

interface (Protocols Layer 2 Circuit)

Syntax

```
interface interface-name {
  backup-neighbor address;
  bandwidth (bandwidth | ctnumber bandwidth);
  community community-name;
  connection-protection;
  (control-word | no-control-word);
  description text;
  egress-protection;
  encapsulation-type type;
  flow-label-receive;
  flow-label-receive-static;
  flow-label-transmit;
  flow-label-transmit-static;
  ignore-encapsulation-mismatch;
  ignore-mtu-mismatch;
  mtu mtu-number;
  no-revert;
  oam;
  protect-interface interface-name;
  pseudowire-status-tlv hot-standby-vc-on;
  psn-tunnel-endpoint address;
  revert-time seconds;
  static {
  switchover-delay milliseconds;
  virtual-circuit-id identifier;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols l2circuit local-switching],
[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address*],
[edit protocols l2circuit local-switching],
[edit protocols l2circuit neighbor *address*]

Release Information

Statement introduced before Junos OS Release 7.4.

flow-label-receive-static and **flow-label-transmit-static** options introduced in Junos OS Release 14.1.

Description

Interface over which Layer 2 circuit traffic travels.

Options

interface-name—Name of the interface to configure.

> NOTE: The commit operation fails, if the same logical interface is configured for both layer 2 circuit and ccc connection.

connection-protection—Enable end-to-end protection through OAM failure detection.

flow-label-receive—Advertise capability to pop flow label in receive direction to the remote provider edge (PE) device.

flow-label-receive-static—Pop flow label on the pseudowire packets received from the remote PE device. The ingress PE inserts the flow label in the pseudowire packet, irrespective of the information exchanged in the signaling plane. If the egress PE cannot handle the pseudowire packet marked with the flow label, the packet is dropped.

flow-label-transmit— Advertise capability to push flow label in transmit direction to the remote PE device.

flow-label-transmit-static—Push flow label on the pseudowire packets sent to the remote PE device. If the incoming pseudowire packet is not marked with the flow label, the packet is dropped by the egress PE.

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 the Neighbor Interface for the Layer 2 Circuit on page 47

l2circuit

Syntax

```
l2circuit {
  local-switching {
    interface interface-name {
      description text;
      end-interface {
        interface interface-name;
        protect-interface interface-name;
      }
      ignore-mtu-mismatch;
      protect-interface interface-name;
    }
  }
  neighbor address {
    interface interface-name {
      backup-neighbor address;
      bandwidth (bandwidth | ctnumber bandwidth);
      community community-name;
      connection-protection;
      (control-word | no-control-word);
      description text;
      egress-protection;
      encapsulation-type type;
      ignore-encapsulation-mismatch;
      ignore-mtu-mismatch;
      mtu mtu-number;
      protect-interface interface-name;
      pseudowire-status-tlv hot-standby-vc-on;
      psn-tunnel-endpoint address;
      virtual-circuit-id identifier;
    }
  }
  traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
  }
}
```

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

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 11.1 for EX Series switches.

Description Enables a Layer 2 circuit.

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 ATM Trunking on Layer 2 Circuits on page 62
- Configuring Bandwidth Allocation and Call Admission Control in Layer 2 Circuits on page 63
- Configuring Interfaces for Layer 2 Circuits on page 47
- Configuring LDP for Layer 2 Circuits on page 21
- Configuring Policies for Layer 2 Circuits on page 57
- Configuring Static Layer 2 Circuits on page 56
- Tracing Layer 2 Circuit Operations on page 351

l2vpn

Syntax

```
l2vpn {
  (control-word | no-control-word);
  encapsulation-type type;
  oam {
    bfd-liveness-detection {
      detection-time {
        threshold milliseconds;
      }
      minimum-interval milliseconds;
      minimum-receive-interval milliseconds;
      multiplier number;
      no-adaptation;
      transmit-interval {
        threshold milliseconds;
        minimum-interval milliseconds;
      }
      version (1 | automatic);
    }
    ping-interval seconds;
  }
  site site-name {
    community COMM;
    control-word ;
    encapsulation-type ethernet;
    ignore-encapsulation-mismatch;
    ignore-mtu-mismatch;
    interface interface-name {
      description text;
      community COMM;
      control-word ;
      encapsulation-type ethernet;
      ignore-encapsulation-mismatch;
      ignore-mtu-mismatch;
      mtu 1500;
      no-control-word;
      oam {
        bfd-liveness-detection {
          detection-time {
            threshold milliseconds;
          }
          minimum-interval milliseconds;
          minimum-receive-interval milliseconds;
          multiplier number;
          no-adaptation;
          transmit-interval {
            threshold milliseconds;
            minimum-interval milliseconds;
          }
          version (1 | automatic);
        }
        ping-interval seconds; seconds;
      }
```

```
                remote-site-id remote-site-id;
                target-attachment-identifier identifier;
            }
            mtu 1500;
            no-control-word;
            oam {
                bfd-liveness-detection {
                    detection-time {
                        threshold milliseconds;
                    }
                    minimum-interval milliseconds;
                    minimum-receive-interval milliseconds;
                    multiplier number;
                    no-adaptation;
                    transmit-interval {
                        threshold milliseconds;
                        minimum-interval milliseconds;
                    }
                    version (1 | automatic);
                }
                ping-interval seconds; seconds;
            }
            site-identifier identifier;
            site-preference preference-value {
                backup;
                primary;
            }
            source-attachment-identifier identifier;
        }
        traceoptions {
            file filename <files number> <size size> <world-readable | no-world-readable>;
            flag flag <flag-modifier> <disable>;
        }
    }
```

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

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 11.1 for EX Series switches.

Description Enable a Layer 2 VPN routing instance on a PE router or switch.

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 the Local Site on PE Routers in Layer 2 VPNs*
- *Configuring an MPLS-Based Layer 2 VPN (CLI Procedure)*

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*

l2vpn-use-bgp-rules

Syntax l2vpn-use-bgp-rules;

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

Release Information Statement introduced in Junos OS Release 12.3.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.

Description Enable routers to use both the BGP path selection algorithm and the designated forwarder path selection algorithm when selecting the preferred path to each destination in a Layer 2 VPN or VPLS routing instance. The BGP path selection algorithm is used by all of the Provider routers participating in the routing instance. The designated forwarder path selection algorithm is used by the PE router participating in the routing instance.

Default By default, the designated forwarder path selection algorithm is used to select the best path to reach each destination within Layer 2 VPN and 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

- Enabling BGP Path Selection for Layer 2 VPNs and VPLS on page 60
- **route-distinguisher on page 294**

local-switching (Layer 2 Circuits)

Syntax

```
local-switching {
  interface interface-name {
    description text;
    encapsulation-type;
    end-interface {
      interface interface-name;
      no-revert;
      protect-interface interface-name;
    }
    ignore-encapsulation-mismatch;
    ignore-mtu-mismatch;
    no-revert;
    protect-interface interface-name;
  }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols l2circuit],
[edit protocols l2circuit]

Release Information

Statement introduced before Junos OS Release 7.4.

Description

Configure a local switching interface. A local switching interface allows you to terminate a virtual circuit on the local router.

The remaining statements are explained separately.

Required Privilege Level

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

Related Documentation

- Configuring Local Interface Switching in Layer 2 Circuits on page 45

minimum-interval (BFD Liveness Detection)

Syntax minimum-interval *milliseconds*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp bfd-liveness-detection],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn oam bfd-liveness-detection],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls neighbor *neighbor-id* oam bfd-liveness-detection],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name* neighbor *neighbor-id* oam bfd-liveness-detection],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls oam bfd-liveness-detection],
[edit protocols bgp bfd-liveness-detection],
[edit protocols bgp group *group-name* bfd-liveness-detection],
[edit protocols bgp group *group-name* neighbor *address* bgp bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection]
[edit routing-instances *routing-instance-name* protocols l2vpn oam bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols vpls neighbor *neighbor-id* oam bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name* neighbor *neighbor-id* oam bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols vpls oam bfd-liveness-detection]

Release Information
Statement introduced in Junos OS Release 8.5.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 13.2 for Layer 2 VPN and VPLS.

Description Configure the minimum interval after which the local routing device transmits hello packets and then expects to receive a reply from a neighbor with which it has established a BFD session. Optionally, instead of using this statement, you can specify the minimum transmit and receive intervals separately using the *minimum-interval* (specified under the **transmit-interval** statement) and *minimum-receive-interval* statements.

Options *milliseconds*—Specify the minimum interval value for BFD liveliness detection.

Range: 1 through 255,000

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*
- *bfd-liveness-detection*
- *minimum-receive-interval*
- *transmit-interval*

mtu

Syntax mtu *bytes*;

Hierarchy Level [edit interfaces *interface-name*],
[edit interfaces *interface-name* unit *logical-unit-number* family *family*],
[edit interfaces interface-range *name*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family *family*],
[edit logical-systems *logical-system-name* protocols l2circuit local-switching interface *interface-name* backup-neighbor *address*],
[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*],
[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name* backup-neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn interface *interface-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls],
[edit protocols l2circuit local-switching interface *interface-name* backup-neighbor *address*],
[edit protocols l2circuit neighbor *address* interface *interface-name*]
[edit protocols l2circuit neighbor *address* interface *interface-name* backup-neighbor *address*],
[edit routing-instances *routing-instance-name* protocols l2vpn interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols l2vpn site *site-name*],
[edit routing-instances *routing-instance-name* protocols vpls]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Support for Layer 2 VPNs and VPLS introduced in Junos OS Release 10.4.
Statement introduced in Junos OS Release 12.1X48 for PTX Series Packet Transport Routers.
Statement introduced in Junos OS Release 12.2 for ACX Series Universal Access Routers.
Support at the **[set interfaces interface-name unit logical-unit-number family ccc]** hierarchy level introduced in Junos OS Release 12.3R3 for MX Series routers.

Description Specify the maximum transmission unit (MTU) size for the media or protocol. The default MTU size depends on the device type. Changing the media MTU or protocol MTU causes an interface to be deleted and added again.

To route jumbo data packets on an integrated routing and bridging (IRB) interface or routed VLAN interface (RVI) on EX Series switches, you must configure the jumbo MTU size on the member physical interfaces of the VLAN that you have associated with the IRB interface or RVI, as well as on the IRB interface or RVI itself (the interface named irb or vlan, respectively).

CAUTION: For EX Series switches, setting or deleting the jumbo MTU size on an IRB interface or RVI while the switch is transmitting packets might cause packets to be dropped.

NOTE:

The MTU for an IRB interface is calculated by removing the Ethernet header overhead [6(DMAC)+6(SMAC)+2(EtherType)]. Because, the MTU is the lower value of the MTU configured on the IRB interface and the MTU configured on the IRB's associated bridge domain IFDs or IFLs, the IRB MTU is calculated as follows:

- In case of Layer 2 IFL configured with the **flexible-vlan-tagging** statement, the IRB MTU is calculated by including 8 bytes overhead (SVLAN+CVLAN).
- In case of Layer 2 IFL configured with the **vlan-tagging** statement, the IRB MTU is calculated by including a single VLAN 4 bytes overhead.

NOTE:

- If a packet whose size is larger than the configured MTU size is received on the receiving interface, the packet is eventually dropped. The value considered for MRU (maximum receive unit) size is also the same as the MTU size configured on that interface.
- Not all devices allow you to set an MTU value, and some devices have restrictions on the range of allowable MTU values. You cannot configure an MTU for management Ethernet interfaces (fxp0, em0, or me0) or for loopback, multilink, and multicast tunnel devices.
- On ACX Series routers, you can configure the protocol MTU by including the **mtu** statement at the **[edit interfaces *interface-name* unit *logical-unit-number* family inet]** or **[edit interfaces *interface-name* unit *logical-unit-number* family inet6]** hierarchy level.
  - If you configure the protocol MTU at any of these hierarchy levels, the configured value is applied to all families that are configured on the logical interface.
  - If you are configuring the protocol MTU for both **inet** and **inet6** families on the same logical interface, you must configure the same value for both the families. It is not recommended to configure different MTU size values for **inet** and **inet6** families that are configured on the same logical interface.

For more information about configuring MTU for specific interfaces and router or switch combinations, see *Configuring the Media MTU*.

Options *bytes*—MTU size.

Range: 256 through 9192 bytes, 256 through 9216 (EX Series switch interfaces), 256 through 9500 bytes (Junos OS 12.1X48R2 for PTX Series routers)

Default: 1500 bytes (INET, INET6, and ISO families), 1448 bytes (MPLS), 1514 bytes (EX Series switch interfaces)

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

Related Documentation

- *Configuring Gigabit Ethernet Interfaces (CLI Procedure)*
- *Configuring Gigabit Ethernet Interfaces (CLI Procedure)*
- *Configuring Routed VLAN Interfaces (CLI Procedure)*
- *Configuring Integrated Routing and Bridging Interfaces (CLI Procedure)*
- *Configuring the Media MTU*
- Configuring the MTU for Layer 2 Interfaces on page 55
- *Setting the Protocol MTU*

neighbor (Protocols Layer 2 Circuit)

Syntax

```
neighbor address {
  interface interface-name {
    backup-neighbor address {
      community name;
      hot-standby;
      psn-tunnel-endpoint address;
      standby;
      virtual-circuit-id number;
    }
    bandwidth (bandwidth | ctnumber bandwidth);
    community community-name;
    (control-word | no-control-word);
    description text;
    egress-protection {
      protected-l2circuit {
        egress-pe address;
        ingress-pe address;
        virtual-circuit-id identifier;
      }
      protector-interface interface-name;
      protector-pe address {
        context-identifier identifier;
        lsp lsp-name;
        }
      }
    }
    encapsulation-type type;
    ignore-encapsulation-mismatch;
    ignore-mtu-mismatch;
    mtu mtu-number;
    no-revert;
    protect-interface interface-name;
    pseudowire-status-tlv hot-standby-vc-on;
    psn-tunnel-endpoint address;
    revert-time seconds;
    static {
      incoming-label label;
      outgoing-label label;
      send-oam;
    }
    switchover-delay milliseconds;
    virtual-circuit-id identifier;
  }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols l2circuit],
[edit protocols l2circuit]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 11.1 for EX Series switches.

Description Each Layer 2 circuit is represented by the logical interface connecting the local provider edge (PE) router or switch to the local customer edge (CE) router or switch. All the Layer 2 circuits using a particular remote PE router or switch designated for remote CE routers or switches are listed under the **neighbor** statement (neighbor designates the PE router or switch). Each neighbor is identified by its IP address and is usually the end-point destination for the LSP tunnel (transporting the Layer 2 circuit).

Options *address*—IP address of a neighboring router or switch.

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 the Neighbor Interface for the Layer 2 Circuit on page 47

no-control-word (Protocols Layer 2 VPN)

Syntax no-control-word;

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

Release Information Statement introduced before Junos OS Release 7.4.

Description Disable the control word. This might be necessary on networks with equipment that does not support the control word.

NOTE: The following configuration statements are ignored for time-division multiplexing pseudowires at the [edit protocols l2vpn] hierarchy level:

- control-word
- no-control-word

Default The control word is enabled by default. Use the **no-control-word** statement to disable the control word.

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

Related Documentation

- *Disabling the Control Word for Layer 2 VPNs*
- *control-word*

no-revert (Local Switching)

Syntax no-revert;

Hierarchy Level [edit protocols l2circuit local-switching interfaces *interface-name*]
[edit protocols l2circuit local-switching interfaces *interface-name* end-interface *interface-name*]

Release Information Statement introduced in Junos OS Release 11.4.

Description (Optional) Prevent the local switching interface from reverting to the primary interface.

NOTE: **The protect interface must be configured prior to configuring the no-revert statement.**

Typically, when the primary interface goes down, the pseudowire starts using the protect interface. By default, when the primary interface comes back online, the interface is switched-over back from the protect interface to the primary interface. To prevent the switchover back to the primary interface, unless the primary interface goes down, include the **no-revert** statement. This prevents loss of traffic during the switchover.

NOTE: **If the protect interface fails, the interface is switched-over back to the primary interface, irrespective of whether or not the no-revert statement is included in the configuration.**

This statement can be configured both for the starting interface and the ending interface.

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

Related Documentation

- Configuring Local Interface Switching in Layer 2 Circuits on page 45

no-revert (Neighbor Interface)

Syntax no-revert;

Hierarchy Level [edit protocols l2circuit neighbor *address* interfaces *interface-name*],
[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interfaces *interface-name*]

Release Information Statement introduced in Junos OS Release 11.3.

Description (Optional) Prevent the protect interface from reverting to the primary interface.

NOTE: The protect interface must be configured prior to configuring the no-revert statement.

Typically, when the primary interface goes down, the pseudowire starts using the protect interface. By default, when the primary interface comes back online, the interface is switched-over back from the protect interface to the primary interface. To prevent the switchover back to the primary interface, unless the protect interface goes down, include the **no-revert** statement. This prevents loss of traffic during the switchover.

NOTE: If the protect interface fails, the interface is switched-over back to the primary interface, irrespective of whether or not the no-revert statement is included in the configuration.

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 Layer 2 Circuits on page 47

oam

Syntax

```
oam {
  ping-interval;
  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 bfd-protocol-version;
  }
  control-channel {
    pwe3–control-word;
    pseudowire-label-ttl-1;
    router-alert-label;
  }
}
```

Hierarchy Level

[edit routing-instances *routing-instance-name* protocols l2vpn],
[edit routing-instances *routing-instance-name* protocols vpls],
[edit routing-instances *routing-instance-name* protocols vpls neighbor *address*],
[edit protocols l2circuit neighbor *address* interface *interface-name*]

Release Information

Statement introduced in Junos OS Release 10.0.
Support for VPLS FEC 129 introduced in Junos OS Release 12.2.

Description

Allows you to configure bidirectional forwarding detection (BFD) and a control channel for a pseudowire, in addition to the corresponding operations and management functions to be used over that control channel. BFD provides a low resource fault detection mechanism for the continuous monitoring of the pseudowire data path and for detecting data plane failures. The **control-channel** statement is not applicable to Layer 2 circuit pseudowires.

Options

The **bfd-liveness-detection** statement and substatements are described in the *Junos OS Routing Protocols Library for Routing Devices*.

The other 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 VCCV for Layer 2 VPNs, Layer 2 Circuits, and VPLS*

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*

protect-interface

Syntax protect-interface *interface-name*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*],
[edit logical-systems *logical-system-name* protocols l2circuit local-switching interface *interface-name*],
[edit logical-systems *logical-system-name* protocols l2circuit local-switching interface *interface-name* end-interface],
[edit protocols l2circuit local-switching interface *interface-name*],
[edit protocols l2circuit neighbor *address* interface *interface-name*],
[edit protocols l2circuit local-switching interface *interface-name* end-interface]

Release Information Statement introduced before Junos OS Release 7.4.

Description Provide a backup for the protected interface in case of failure. Network traffic uses the primary interface only, as long as the primary interface functions.

Options *interface-name*—Name of the protect interface to configure.

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

Related Documentation

- Configuring the Protect Interface on page 50

protected-l2circuit

Syntax

```
protected-l2circuit {
  egress-pe address;
  ingress-pe address;
  virtual-circuit-id identifier;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name* egress-protection],
[edit protocols l2circuit neighbor *address* interface *interface-name* egress-protection]

Release Information

Statement introduced in Junos OS release 10.4.

Description

Configures the protected Layer 2 circuit as part of an egress protection virtual circuit (EPVC).

Options

egress-pe ***address***—Specify the address of the egress PE router for the protected Layer 2 circuit.

ingress-pe ***address***—Specify the address of the ingress PE router for the protected Layer 2 circuit.

virtual-circuit-id ***identifier***—Specify the virtual circuit identifier for the protected Layer 2 circuit.

Required Privilege Level

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

Related Documentation

- Example: Configuring an Egress Protection LSP for a Layer 2 Circuit on page 156

protector-interface

Syntax protector-interface *interface-name*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name* egress-protection],
[edit protocols l2circuit neighbor *address* interface *interface-name* egress-protection]

Release Information Statement introduced in Junos OS release 10.4.

Description Configures the protector interface for an egress protection LSP.

Options ***interface-name***——Name of the interface used to protect traffic for an egress protection LSP.

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

Related Documentation

- Example: Configuring an Egress Protection LSP for a Layer 2 Circuit on page 156

protector-pe

Syntax

```
protector-pe address {
  context-identifier identifier;
  lsp lsp-name;
  }
}
```

Hierarchy Level [edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name* egress-protection],
[edit protocols l2circuit neighbor *address* interface *interface-name* egress-protection]

Release Information Statement introduced in Junos OS release 10.4.

Description Configures the protector PE router for an egress protection LSP. Test.

Options ***address***——IPv4 address for the protector PE router.

context-identifier *identifier*——Identifies the context for the egress protection LSP.

lsp *lsp-name*——Specifies the LSP for the egress protection LSP.

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

Related Documentation

- Example: Configuring an Egress Protection LSP for a Layer 2 Circuit on page 156

pseudowire-status-tlv

Syntax pseudowire-status-tlv hot-standby-vc-on;

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

Release Information Statement introduced in Junos OS Release 10.0.

Description Enables the pseudowire type length variable (TLV). The pseudowire status TLV is used to communicate the status of a pseudowire back and forth between two PE routers. The pseudowire status TLV is configurable for each pseudowire connection and is disabled by default.

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 Pseudowire Status TLV on page 51
- Example: Configuring Pseudowire Redundancy for Mobile Backhaul Scenarios on page 211

psn-tunnel-endpoint

Syntax psn-tunnel-endpoint *address*;

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

Release Information Statement introduced before Junos OS Release 7.4.
Hierarchy levels associated with the **backup-neighbor** statement added in Junos OS Release 9.2.

Description Specify the endpoint of the packet switched network (PSN) tunnel on the remote PE router.

Options ***address***—Address for the tunnel endpoint.

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

Related Documentation

- Configuring Layer 2 Circuits over Both RSVP and LDP LSPs on page 52
- *Configuring Redundant Pseudowires for Layer 2 Circuits and VPLS*

revert-time (Protocols Layer 2 Circuits)

Syntax revert-time *seconds* maximum *seconds*;

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

Release Information Statement introduced in Junos OS Release 10.2.
maximum option introduced in Junos OS Release 13.2.

Description Specify a revert time for redundant Layer 2 circuits and VPLS pseudowires. When you have configured redundant pseudowires for Layer 2 circuits or VPLS, traffic is switched to the backup connection in the event that the primary connection fails. If you configure a revert time, when the configured time expires traffic is reverted to the primary path, assuming the primary path has been restored.

With the **maximum** option, specify a maximum reversion interval to add after the **revert-time** delay. If a revert-time delay is defined but a maximum timer is not defined, VCs are restored upon the revert-timer's expiration.

To reduce as much as possible the amount of traffic discarded, and potential data-path asymmetries observed during primary-to-backup transition periods, you can use this restoration timer. This restoration timer is activated when the backup path is performing as active, and then the primary path is restored. The goal is to avoid moving traffic back to the primary path right away, to make sure that the control plane's related tasks (such as IGP, LDP, RSVP, and internal BGP) have enough time to complete their updating cycle.

By enabling a gradual return of traffic to the primary path, you can ensure that the relatively-slow control-plane processing and updating does not have a negative impact on the restoration process.

The **maximum** option extends the revert timer's functionality to provide a jittered interval over which a certain number of circuits can be transitioned back to the primary path. By making use of this maximum value, you can define a time interval during which circuits are expected to switch over. As a consequence, circuits' effective transitions are scattered during restoration periods.

When making use of **revert-time *x* maximum *y*** statement, you can ensure that the corresponding circuit that is active is moved to the primary path within a time-slot (t1) such as that: x <= t1 <= y. In other words, by activating this statement, you can ensure the following:

- VCs stay in the backup path for at least x seconds after the primary path comes back up.
- VCs are moved back to the primary path before y seconds have elapsed.
- y maximum value = x maximum value * 2 = 1200 seconds.

The ideal values for x and y will are conditioned to internal aspects of your network. For this reason, there are no default values for these settings. If no revert-time is set, the default behavior is non-revertive. That is, circuits are not returned to the primary path upon restoration. They are kept on the backup path.

Default Without the **revert-time** statement, virtual circuit (VC) traffic is not transitioned to the primary path upon restoration of the primary path.

Options *seconds*—Revert time in seconds.

Range: 0 through 600 seconds

maximum *seconds*—Number of seconds to delay path restoration after the **revert-time** delay.

Range: 0 through 1200 seconds

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

Related Documentation

- *Configuring Redundant Pseudowires for Layer 2 Circuits and VPLS*

route-distinguisher

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

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

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 11.1 for EX Series switches.
Support at **[edit routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name*]** hierarchy level introduced in Junos OS Release 11.2.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.
Support at **[edit routing-instances *routing-instance-name* protocols l2vpn mesh-group *mesh-group-name*]** hierarchy level introduced in Junos OS Release 13.2.

Description Specify an identifier attached to a route, enabling you to distinguish to which VPN or VPLS the route belongs. Each routing instance must have a unique route distinguisher associated with it. The route distinguisher is used to place bounds around a VPN so that the same IP address prefixes can be used in different VPNs without having them overlap. If the instance type is **vrf**, the **route-distinguisher** statement is required.

For Layer 2 VPNs and VPLS, if you configure the **l2vpn-use-bgp-rules** statement, you must configure a unique route distinguisher for each PE router participating in the routing instance.

For other types of VPNs, we recommend that you use a unique route distinguisher for each PE router participating in specific routing instance. Although you can use the same route distinguisher on all PE routers for the same VPN routing instance, if you use a unique route distinguisher, you can determine the CE router from which a route originated within the VPN.

For Layer 2 VPNs and VPLS, if you configure mesh groups, the route distinguisher in each mesh group must be unique.

CAUTION: We strongly recommend that if you change a route distinguisher that has already been configured, make the change during a maintenance window, as follows:

1. Deactivate the routing instance.
2. Change the route distinguisher.
3. Activate the routing instance.

This is not required if you are configuring the route distinguisher for the first time.

Options ***as-number*:*number*—*as-number*** is an assigned AS number, and ***number*** is any 2-byte or 4-byte value. The AS number can be from 1 through 4,294,967,295. If the AS number is a 2-byte value, the administrative number is a 4-byte value. If the AS number is 4-byte value, the administrative number is a 2-byte value. A route distinguisher consisting of a 4-byte AS number and a 2-byte administrative number is defined as a type 2 route distinguisher in RFC 4364 *BGP/MPLS IP Virtual Private Networks (VPNs)*.

NOTE: In Junos OS Release 9.1 and later, the numeric range for AS numbers is extended to provide BGP support for 4-byte AS numbers, as defined in RFC 4893, *BGP Support for Four-octet AS Number Space*. All releases of Junos OS support 2-byte AS numbers. To configure a route distinguisher that includes a 4-byte AS number, append the letter “L” to the end of the AS number. For example, a route distinguisher with the 4-byte AS number 7,765,000 and an administrative number of 1,000 is represented as 77765000L:1000.

In Junos OS Release 9.2 and later, you can also configure a 4-byte AS number using the AS dot notation format of two integer values joined by a period: *<16-bit high-order value in decimal>*.*<16-bit low-order value in decimal>*. For example, the 4-byte AS number of 65,546 in the plain-number format is represented as 1.10 in AS dot notation format.

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

Range: 0 through 4,294,967,295 ($2^{32} - 1$). If the router you are configuring is a BGP peer of a router that does not support 4-byte AS numbers, you need to configure a local AS number. For more information, see *Using 4-Byte Autonomous System Numbers in BGP Networks Technology Overview*.

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 Route Target Filtering for VPNs*
- *Example: Configuring FEC 129 BGP Autodiscovery for VPWS*
- *Configuring EVPN Routing Instances*
- Configuring the Route Distinguisher on page 37
- *Configuring an MPLS-Based Layer 2 VPN (CLI Procedure)*
- *Configuring an MPLS-Based Layer 3 VPN (CLI Procedure)*
- **l2vpn-use-bgp-rules on page 274**

send-oam

Syntax send-oam;

Hierarchy Level [edit protocols l2circuit neighbor address interface interface-name static]

Release Information Statement introduced in Junos OS Release 9.5.
Statement introduced in Junos OS Release 12.1x48 for PTX Series Packet Transport Routers.
Statement introduced in Junos OS Release 12.2 for the ACX Series Universal Access Routers.

Description Enable the ability to ping a static pseudowire. If you configure the **send-oam** statement, it applies to the backup neighbor as well. Once you have configured this statement, you can ping the static pseudowire by issuing the **ping mpls l2circuit** command.

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

Related Documentation

- Configuring Static Layer 2 Circuits on page 56
- **ping mpls l2circuit on page 310**

source-attachment-identifier (Protocols VPWS)

Syntax source-attachment-identifier *identifier*;

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

Release Information Statement introduced in Junos OS Release 13.2.

Description For FEC 129, specify the VPWS source attachment identifier. The point-to-point nature of VPWS requires that you specify the source access individual identifier (SAII) and the target access individual identifier (TAII). This SAII-TAII pair defines a unique pseudowire between two PE devices.

Auto-discovery routes are used by BGP to allow auto-discovery of remote source access individual identifiers (SAIIs) (the sources of the point-to-point pseudowires). One auto-discovery route is advertised for each source attachment identifier (SAI) in the instance or mesh group.

The SAII is specified with the **source-attachment-identifier** statement within the FEC 129 VPWS routing instance. You configure the source attachment identifier and the interfaces to associate with that source attachment identifier. Under each interface, you can configure the TAII with the **target-attachment-identifier** statement. If the configured target identifier matches a source identifier advertised by a remote PE device by way of a BGP auto-discovery message, the pseudowire between that source-target pair is signaled. If there is no match between an advertised source identifier and the configured target identifier, the pseudowire is not established.

Options *identifier*—The numerical identifier for the Layer 2 VPN site.

Range: 1 through 4,292,967,295

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

Related Documentation

- *Example: Configuring FEC 129 BGP Autodiscovery for VPWS*
- target-attachment-identifier on page 300

standby (Protocols Layer 2 Circuit)

Syntax standby;

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

Release Information Statement introduced before Junos OS Release 7.4.
Hierarchy levels associated with the **backup-neighbor** statement added in Junos OS Release 9.2.

Description Configure the pseudowire to the specified backup neighbor as the standby. When you configure this statement, traffic flows over both the active and standby pseudowires to the backup device (either a CE device or PE router). The backup device drops the traffic from the standby pseudowire, unless the active pseudowire fails. If the active pseudowire fails, the backup device automatically switches to the standby pseudowire.

The **standby** statement is quite similar to the **hot-standby** statement introduced in Junos OS Release 12.3. The **hot-standby** statement allows for a faster forwarding-path switchover during transition periods, as compared to what is allowed by the **standby** statement.

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

Related Documentation

- Configuring Layer 2 Circuits over Both RSVP and LDP LSPs on page 52
- *Configuring Redundant Pseudowires for Layer 2 Circuits and VPLS*
- Example: Configuring Layer 2 Circuit Switching Protection on page 144

static (Protocols Layer 2 Circuit)

Syntax

```
static {
  incoming-label label;
  outgoing-label label;
  send-oam;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* l2circuit neighbor *address* interface *interface-name*],
[edit logical-systems *logical-system-name* l2circuit neighbor *address* interface *interface-name* backup-neighbor *neighbor*],
[edit protocols l2circuit neighbor *address* interface *interface-name*],
[edit protocols l2circuit neighbor *address* interface *interface-name* backup-neighbor *neighbor*]

Release Information

Statement introduced in Junos OS Release 9.5.
Statement introduced in Junos OS Release 12.1X48 for PTX Series routers.
Statement introduced in Junos OS Release 12.2 for ACX Series routers.

Description

Configures static Layer 2 circuit pseudowires. Static pseudowires are designed for networks that do not support LDP or do not have LDP enabled. You configure a static pseudowire by configuring static values for the in and out labels needed to enable a pseudowire connection.

Options

incoming-label—(Optional for PTX Series Packet Transport Routers only) Configure the Layer 2 circuit incoming static pseudowire label.

Range: 1000000 through 1048575

outgoing-label—(Optional for PTX Series Packet Transport Routers only) Configure the Layer 2 circuit outgoing static pseudowire label.

Range: 16 through 1048575

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 Static Layer 2 Circuits on page 56

target-attachment-identifier (Protocols VPWS)

Syntax target-attachment-identifier *identifier*;

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

Release Information Statement introduced in Junos OS Release 13.2.

Description For FEC 129, specify the VPWS target attachment identifier. The point-to-point nature of VPWS requires that you specify the source access individual identifier (SAII) and the target access individual identifier (TAII). This SAII-TAII pair defines a unique pseudowire between two PE devices.

Auto-discovery routes are used by BGP to allow auto-discovery of SAIIs (the sources of the point-to-point pseudowires). One auto-discovery route is advertised for each source attachment identifier (SAI) in the instance or mesh group.

The SAII is specified with the **source-attachment-identifier** statement within the FEC 129 VPWS routing instance. You configure the source attachment identifier and the interfaces to associate with that source attachment identifier. Under each interface, you can configure the TAII with the **target-attachment-identifier** statement. If the configured target identifier matches a source identifier advertised by a remote PE device by way of a BGP auto-discovery message, the pseudowire between that source-target pair is signaled. If there is no match between an advertised source identifier and the configured target identifier, the pseudowire is not established.

Options ***identifier***—The numerical identifier for the Layer 2 VPN site.

Range: 1 through 4,292,967,295

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

Related Documentation

- *Example: Configuring FEC 129 BGP Autodiscovery for VPWS*
- source-attachment-identifier on page 297

traceoptions (Protocols Layer 2 Circuit)

Syntax

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

Hierarchy Level [edit logical-systems *logical-system-name* protocols l2circuit],
[edit protocols l2circuit]

Release Information Statement introduced before Junos OS Release 7.4.

Description Trace traffic flowing through a Layer 2 circuit.

Options **disable**—(Optional) Disable the tracing operation. You can use this option to disable a single operation when you have defined a broad group of tracing operations, such as **all**.

file *filename*—Name of the file to receive the output of the tracing operation. Enclose the name in quotation marks (" ").

files *number*—(Optional) Maximum number of trace files. When a trace file named ***trace-file*** reaches its maximum size, it is renamed ***trace-file*.0**, then ***trace-file*.1**, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum number of files, you also must specify a maximum file size with the **size** option.

Range: 2 through 1000 files

Default: 2 files

flag *flag*—Tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements.

- **connections**—Layer 2 circuit connections (events and state changes)
- **error**—Error conditions
- **fec**—Layer 2 circuit advertisements received or sent by means of LDP
- **topology**—Layer 2 circuit topology changes caused by reconfiguration or advertisements received from other PE routers

flag-modifier—(Optional) Modifier for the tracing flag. You can specify the **detail** modifier if you want to provide detailed trace information.

no-world-readable—(Optional) Do not allow any user to read the log file.

size *size*—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named *trace-file* reaches this size, it is renamed *trace-file*.0. When the *trace-file* again reaches its maximum size, *trace-file*.0 is renamed *trace-file*.1 and *trace-file* is renamed *trace-file*.0. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum file size, you also must specify a maximum number of trace files with the **files** option.

Syntax: ***x*k** to specify kilobytes, ***x*m** to specify megabytes, or ***x*g** to specify gigabytes

Range: 10 KB through the maximum file size supported on your system

Default: 1 MB

world-readable—(Optional) Allow any user to read the log file.

Required Privilege Level

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

Related Documentation

- Tracing Layer 2 Circuit Operations on page 351

virtual-circuit-id

Syntax virtual-circuit-id *identifier*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols l2circuit local-switching interface *interface-name* backup-neighbor *address*],
[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*],
[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name* backup-neighbor *address*],
[edit protocols l2circuit local-switching interface *interface-name* backup-neighbor *address*],
[edit protocols l2circuit neighbor *address* interface *interface-name*],
[edit protocols l2circuit neighbor *address* interface *interface-name* backup-neighbor *address*]

Release Information Statement introduced before Junos OS Release 7.4.
Hierarchy levels for **backup-neighbor** (pseudowire redundancy) added in Junos OS Release 9.2.
Statement introduced in Junos OS Release 11.1 for EX Series switches.
Statement introduced in Junos OS Release 12.3 at the **[edit protocols l2circuit local-switching interface *interface-name* backup-neighbor *address*]** hierarchy level.

Description Uniquely identify a Layer 2 circuit for either a standard pseudowire or a redundant pseudowire.

Options *identifier*—1 through 4,294,967,295

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

Related Documentation

- Configuring the Virtual Circuit ID on page 53
- *Configuring Redundant Pseudowires for Layer 2 Circuits and VPLS*
- Example: Configuring Layer 2 Circuit Switching Protection on page 144

vrf-target

Syntax

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

Hierarchy Level

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

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 12.3 for ACX Series routers.

Description

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

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

Options

community—Community name.

import *community-name*—Allowed communities accepted from neighbors.

export *community-name*—Allowed communities sent to neighbors.

Required Privilege Level

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

Related Documentation

- Configuring a VRF Target on page 44
- *Example: Configuring FEC 129 BGP Autodiscovery for VPWS*

PART 3

Administration

CHAPTER 7

Layer 2 Circuit Reference

Supported VPWS Standards

Junos OS substantially supports the following RFCs, which define standards for VPWS and Layer 2 circuits.

- RFC 4447, *Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)*

  Junos OS does not support Section 5.3, "The Generalized PWid FEC Element."
- RFC 4448, *Encapsulation Methods for Transport of Ethernet over MPLS Networks*
- RFC 6074, *Provisioning, Auto-Discovery, and Signaling in Layer 2 Virtual Private Networks (L2VPNs)*
- RFC 6391, *Flow-Aware Transport of Pseudowires over an MPLS Packet Switched Network*
- RFC 6790, *The Use of Entropy Labels in MPLS Forwarding*

The following Internet drafts do not define standards, but provide information about Layer 2 technologies. The IETF classifies them as "Historic."

- Internet draft draft-martini-l2circuit-encap-mpls-11.txt, *Encapsulation Methods for Transport of Layer 2 Frames Over IP and MPLS Networks*

  Junos OS differs from the Internet draft in the following ways:

  - A packet with a sequence number of 0 (zero) is treated as out of sequence.
  - Any packet that does not have the next incremental sequence number is considered out of sequence.
  - When out-of-sequence packets arrive, the expected sequence number for the neighbor is set to the sequence number in the Layer 2 circuit control word.
- Internet draft draft-martini-l2circuit-trans-mpls-19.txt, *Transport of Layer 2 Frames Over MPLS*

Related Documentation

- *Supported Carrier-of-Carriers and Interprovider VPN Standards*

- *Supported Layer 2 VPN Standard*
- *Supported Layer 3 VPN Standards*
- *Supported Multicast VPN Standards*
- *Supported VPLS Standards*
- *Accessing Standards Documents on the Internet*

CHAPTER 8

Operational Commands

ping mpls l2circuit

Syntax

ping mpls l2circuit (interface *interface-name* | virtual-circuit *virtual-circuit-id* neighbor *address*)
<count *count*>
<destination *address*>
<detail>
<exp *forwarding-class*>
<logical-system (all | *logical-system-name*)>
reply-mode (application-level-control-channel | ip-udp | no-reply)
<size *bytes*>
<source *source-address*>
<sweep>
<v1>

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
The **size** and **sweep** options were introduced in Junos OS Release 9.6.
The **reply-mode** option and its suboptions are introduced in Junos OS Release 10.4R1.

Description

Check the operability of the MPLS Layer 2 circuit connections. Type Ctrl+c to interrupt a **ping mpls l2circuit** command. You can also issue this command within logical systems.

Options

count ***count***—(Optional) Number of ping requests to send. If **count** is not specified, five ping requests are sent. The range of values is 1 through 1,000,000. The default value is 5.

destination ***address***—(Optional) Specify an address other than the default (127.0.0.1/32) for the ping echo requests. The address can be anything within the 127/8 subnet.

detail—(Optional) Display detailed information about the echo requests sent and received.

exp ***forwarding-class***—(Optional) Value of the forwarding class for the MPLS ping packets.

interface ***interface-name***—Ping an interface configured for the Layer 2 circuit on the egress provider edge (PE) router.

logical-system (all | ***logical-system-name*****)**—(Optional) Perform this operation on all logical systems or on the specified logical system.

reply-mode—(Optional) Reply mode for the ping request. This option has the following suboptions:

application-level-control-channel—Reply using an application level control channel.

ip-udp—Reply using an IPv4 or IPv6 UDP packet.

no-reply—Do not reply to the ping request.

NOTE: The **reply-mode** option and its suboptions **application-level-control-channel**, **ip-udp**, and **no-reply** are also available in Junos OS Release 10.2R4 and 10.3R2.

size *bytes*—(Optional) Size of the label-switched path (LSP) ping request packet (96 through 65468 bytes). Packets are 4-byte aligned. For example, If you enter a size of 97, 98, 99, or 100, the router or switch uses a size value of 100 bytes. If you enter a packet size that is smaller than the minimum size, an error message is displayed reminding you of the 96-byte minimum.

source *source-address*—(Optional) IP address of the outgoing interface. This address is sent in the IP source address field of the ping request. If this option is not specified, the default address is usually the loopback interface (lo.0).

sweep—(Optional) Automatically determine the size of the maximum transmission unit (MTU).

v1—(Optional) Use the type 9 Layer 2 circuit type, length, and value (TLV).

virtual-circuit *virtual-circuit-id* neighbor *address*—Ping the virtual circuit identifier on the egress PE router or switch and the specified neighbor, testing the integrity of the Layer 2 circuit between the ingress and egress PE routers or switches.

Additional Information

You must configure MPLS at the **[edit protocols mpls]** hierarchy level on the egress PE router or switch (the router or switch receiving the MPLS echo packets) to ping a Layer 2 circuit.

In asymmetric MTU scenarios, the echo response may be dropped. For example, if the MTU from System A to System B is 1000 bytes, the MTU from System B to System A is 500 bytes, and the ping request packet size is 1000 bytes, the echo response is dropped because the PAD TLV is included in the echo response, making it too large.

Required Privilege Level

network

List of Sample Output

ping mpls l2circuit interface on page 311
ping mpls l2circuit virtual-circuit detail on page 311
ping mpls l2circuit interface <interface-name> reply-mode on page 312

Output Fields

When you enter this command, you are provided feedback on the status of your request. An exclamation point (**!**) indicates that an echo reply was received. A period (**.**) indicates that an echo reply was not received within the timeout period. An **x** indicates that an echo reply was received with an error code. Packets with an error code are not counted in the received packets count. They are accounted for separately.

Sample Output

ping mpls l2circuit interface

```
user@host> ping mpls l2circuit interface so-1/0/0.1
Request for seq 1, to interface 69, labels <100000, 100208>, packet size 100
Reply for seq 1, return code: Egress-ok, time: 0.439 ms
```

ping mpls l2circuit virtual-circuit detail

```
user@host> ping mpls l2circuit virtual-circuit 200 neighbor 10.255.245.122/32 detail
```

```
Request for seq 1, to interface 68, labels <100048, 100128>, packet size 100

Reply for seq 1, return code: Egress-ok time: 0.539 ms
```

ping mpls l2circuit interface <interface-name> reply-mode

```
user@host> ping mpls l2circuit interface lt-1/2/0.21 reply-mode application-level-control-channel
!!!!!
--- lsping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
```


ping mpls l2vpn

Syntax

ping mpls l2vpn (instance *instance-name* local-site-id *local-site-id-number* remote-site-id *remote-site-id-number* | interface *interface-name*)
<bottom-label-ttl>
<count *count*>
<destination *address*>
<detail>
<exp *forwarding-class*>
<fec129>
<logical-system (all | *logical-system-name*)>
reply-mode (application-level-control-channel | ip-udp | no-reply)
<size *bytes*>
<source *source-address*>
<sweep>

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
size and **sweep** options introduced in Junos OS Release 9.6.
reply-mode option and its suboptions introduced in Junos OS Release 10.4R1.
fec129 option introduced in Junos OS Release 12.2.

Description

Check the operability of MPLS Layer 2 virtual private network (VPN) connections. Type Ctrl+c to interrupt a **ping mpls l2vpn** command.

Options

bottom-label-ttl—(Optional) Display the time-to-live value for the bottom label in the label stack.

count ***count***—(Optional) Number of ping requests to send. If **count** is not specified, five ping requests are sent. The range of values is **1** through 1,000,000. The default value is **5**.

destination ***address***—(Optional) Specify an address other than the default (**127.0.0.1/32**) for the ping echo requests. The address can be anything within the **127/8** subnet.

detail—(Optional) Display detailed information about the echo requests sent and received.

exp ***forwarding-class***—(Optional) Value of the forwarding class for the MPLS ping packets.

fec129—(Optional) Ping the LSP for an FEC 129 Layer 2 VPN connection.

instance ***instance-name*** **local-site-id** ***local-site-id-number*** **remote-site-id** ***remote-site-id-number***—Ping a combination of the Layer 2 VPN routing instance name, the local site identifier, and the remote site identifier, testing the integrity of the Layer 2 VPN circuit (specified by the identifiers) between the ingress and egress provider edge (PE) routers or switches.

interface ***interface-name***—Ping an interface configured for the Layer 2 VPN on the egress PE router or switch.

logical-system (all | ***logical-system-name*****)**—(Optional) Perform this operation on all logical systems or on the specified logical system.

reply-mode—(Optional) Reply mode for the ping request. This option has the following suboptions:

application-level-control-channel—Reply using an application level control channel.

ip-udp—Reply using an IPv4 or IPv6 UDP packet.

no-reply—Do not reply to the ping request.

The **reply-mode** option and its suboptions **application-level-control-channel, ip-udp**, and **no-reply** are also available in Junos OS Release 10.2R4 and 10.3R2.

size ***bytes***—(Optional) Size of the label-switched path (LSP) ping request packet (**96** through **65468** bytes). Packets are 4-byte aligned. For example, If you enter a size of 97, 98, 99, or 100, the router or switch uses a size value of 100 bytes. If you enter a packet size that is smaller than the minimum size, an error message is displayed reminding you of the 96-byte minimum.

source ***source-address***—(Optional) IP address of the outgoing interface. This address is sent in the IP source address field of the ping request. If this option is not specified, the default address is usually the loopback interface (**lo.0**).

sweep—(Optional) Automatically determine the size of the maximum transmission unit (MTU).

Additional Information

You must configure MPLS at the **[edit protocols mpls]** hierarchy level on the egress PE router or switch (the router or switch receiving the MPLS echo packets) to ping a Layer 2 circuit.

In asymmetric MTU scenarios, the echo response may be dropped. For example, if the MTU from System A to System B is 1000 bytes, the MTU from System B to System A is 500 bytes, and the ping request packet size is 1000 bytes, the echo response is dropped because the PAD TLV is included in the echo response, making it too large.

Required Privilege Level

network

Related Documentation

- *Example: Configuring FEC 129 BGP Autodiscovery for VPWS*

List of Sample Output

ping mpls l2vpn instance on page 315
ping mpls l2vpn instance detail on page 315
ping mpls l2vpn interface <interface-name> reply-mode on page 315
ping mpls l2vpn fec129 interface <interface-name> on page 315
ping mpls l2vpn fec129 instance <instance-name> on page 315

Output Fields

When you enter this command, you are provided feedback on the status of your request. An exclamation point (**!**) indicates that an echo reply was received. A period (**.**) indicates that an echo reply was not received within the timeout period. An **x** indicates that an echo reply was received with an error code. Packets with an error code are not counted in the received packets count. They are accounted for separately.

Sample Output

ping mpls l2vpn instance

```
user@host> ping mpls l2vpn instance vpn1 remote-site-id 1 local-site-id 2
!!!!!
--- lsping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
```

ping mpls l2vpn instance detail

```
user@host> ping mpls l2vpn instance vpn1 remote-site-id 1 local-site-id 2 detail
Request for seq 1, to interface 68, labels <800001, 100176>
Reply for seq 1, return code: Egress-ok
Request for seq 2, to interface 68, labels <800001, 100176>
Reply for seq 2, return code: Egress-ok
Request for seq 3, to interface 68, labels <800001, 100176>
Reply for seq 3, return code: Egress-ok
Request for seq 4, to interface 68, labels <800001, 100176>
Reply for seq 4, return code: Egress-ok
Request for seq 5, to interface 68, labels <800001, 100176>
Reply for seq 5, return code: Egress-ok

--- lsping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
```

ping mpls l2vpn interface <interface-name> reply-mode

```
user@host> ping mpls l2vpn interface lt-1/2/0.21 reply-mode ip-udp
!!!!!
--- lsping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
```

ping mpls l2vpn fec129 interface <interface-name>

```
user@host> ping mpls l2vpn fec129 interface ge-2/0/5.0
!!!!!
--- lsping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
```

ping mpls l2vpn fec129 instance <instance-name>

```
user@PE1> ping mpls l2vpn fec129 instance FEC129-VPWS remote-id 2 remote-pe-address 2.2.2.2
local-id 1
!!!!!
--- lsping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
```

request l2circuit-switchover

Syntax

request l2circuit-switchover
<logical-system (all | logical-system-name)>
<neighbor *address*>
<virtual-circuit-id *identifier*>

Release Information

Command introduced in Junos OS Release 9.2.

Description

Manually trigger a switch from the active pseudowire to the redundant pseudowire. This command can be useful when performing network maintenance.

Options

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

neighbor *address*—(Optional) Trigger a switch of all of the active pseudowire connections with the specified neighbor to their respective redundant pseudowires.

virtual-circuit-id *identifier*—(Optional) Trigger a switch from the active pseudowire connection of the specified Layer 2 circuit to its redundant pseudowire.

Required Privilege Level

maintenance

List of Sample Output

request l2circuit-switchover virtual-circuit-id on page 316

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

request l2circuit-switchover virtual-circuit-id

```
user@host> request l2circuit-switchover virtual-circuit-id 12
```

show l2circuit connections

Syntax

show l2circuit connections
<brief | extensive | summary>
<down | up | up-down>
<history>
<interface *interface-name*>
<logical-system (all | *logical-system-name*)>
<neighbor *neighbor*>
<status>

Release Information

Command introduced before Junos OS Release 7.4.
Display enhancements in Junos OS Release 9.6.
Display enhancements in Junos OS Release 10.2.
Display enhancements in Junos OS Release 12.1.
Display enhancements in Junos OS Release 13.2.

Description

Display status information about Layer 2 virtual circuits from the local provider edge (PE) router to its neighbors.

Options

none—Display standard information about Layer 2 virtual circuits on all interfaces for all neighbors.

brief | extensive | summary—(Optional) Display the specified level of output. Use history to display information about connection history. Use status to display information about the connection and interface status.

down | up | up-down—(Optional) Display nonoperational, operational, or both kinds of connections.

history—(Optional) Display information about connection history.

interface *interface-name*—(Optional) Show all Layer 2 virtual circuits on an interface.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

neighbor *neighbor*—(Optional) IP address of a specific neighbor.

status—(Optional) Display information about the connection and interface status.

Required Privilege Level

view

List of Sample Output

show l2circuit connections on page 320
show l2circuit connections interface on page 321
show l2circuit connections extensive on page 321
show l2circuit connections extensive (Pseudowire Redundancy with Hot Standby) on page 322

Output Fields

Table 6 on page 318 lists the output fields for the **show l2circuit connections** command. Output fields are listed in the approximate order in which they appear.

Table 6: show l2circuit connections Output Fields

Field Name	Field Description
Layer-2 Circuit Connections	Displays the legends for connection and interface status.
Neighbor	Remote PE neighbor.
Interface	Logical PE-to-CE interface on which the virtual circuit is configured.
Type	VC type: **rmt** (remote) or **loc** (local).
Legend for connection status (St)	Status of the virtual circuit connection: • **EI**—The local virtual circuit interface is configured with an encapsulation that is not supported. • **MM**—The two routers do not agree on an MTU value, which causes an MTU mismatch. • **EM**—The encapsulation type received on this virtual circuit from the neighbor does not match the local virtual circuit interface encapsulation type. • **CM**—The two routers do not agree on a control word, which causes a control word mismatch. • **VM**—The remote and local VLAN IDs do not match across the Layer 2 circuit. • **OL**—No advertisement has been received for this virtual circuit from the neighbor. There is no outgoing label available for use by this virtual circuit. • **NC**—The interface is not configured as a CCC or TCC interface. • **BK**—The virtual circuit has switched to a backup connection. • **CB**—The remote PE router is advertising a different cell bundle from that configured on the local PE router. • **LD**—The connection to the local site is signaled down, because the CE-facing interface to the local site is down. • **RD**—The remote neighbor is down. It has signaled a problem using the pseudowire status code. • **NP**—The router detects that interface hardware is not present. The hardware may be offline, a PIC may not be of the desired type, or the interface may be configured in a different routing instance. • **Dn**—The virtual circuit is down. • **VC-Dn**—The virtual circuit is down because there is no tunnel LSP from the local PE router to the neighbor. • **UP**—The virtual circuit is operational. • **CF**—The router cannot find enough bandwidth to the remote router to satisfy the Layer 2 circuit bandwidth requirement. • **IB**—The bit rate is incompatible for Time Division Multiplexing (TDM). • **TDM**—TDM is not configured correctly. • **ST**—The virtual circuit has been switched to a standby connection. • **SP**—The virtual circuit connection is using a static pseudowire. • **RS**—The remote site is in a standby state. • **XX**—The virtual circuit is down for an unknown reason. This is a programming error.
Time last up	Date and time the virtual circuit was last operational.

Table 6: show l2circuit connections Output Fields *(continued)*

Field Name	Field Description
# Up trans	Number of times the virtual circuit came up.
local-interface-name	Name of the local PE-to-CE interface.
Status	Status of the local interface.
Up	Interface is operational.
Dn	Interface is not operational.
NP	Not present. Interface does not exist.
DS	Disabled. Interface has been administratively disabled.
WE	Wrong encapsulation. The interface is not configured as CCC.
UN	Interface status is initialized.
Encapsulation	Encapsulation of the local interface.
Remote PE	Prefix of the remote PE router.
Negotiated control-word	Whether the use of the control word has been negotiated for this virtual circuit: **Yes (Null)** or **No**.
Incoming label	Label used by the remote side of the virtual circuit to send packets destined to the local side. This label is routed to the local virtual circuit interface.
Outgoing label	Label used by the local side of the virtual circuit to send packets to the remote side of the virtual circuit. Packets originated on the local virtual circuit interface are encapsulated with this label before being placed on the tunnel LSP to the neighbor for this virtual circuit. This label is allocated by the neighbor and is used in demultiplexing incoming packets destined for this virtual circuit.
Negotiated PW status TLV	Displays the pseudowire status type, length, and value (TLV). TLVs are a method of encoding variable-length or optional information. If the pseudowire status TLV is used, the corresponding local or neighbor PE router status code is also displayed.
local PW status code	If the pseudowire status TLV is used, displays the local PE router status code.
Neighbor PW status code	If the pseudowire status TLV is used, displays the neighbor PE router status code.
Local interface	Name of the local interface used for the Layer 2 circuit connection.
Status	Status of the local interface (**Up** or **Down**).

Table 6: show l2circuit connections Output Fields *(continued)*

Field Name	Field Description
Encapsulation	Encapsulation configured for the local interface.
APS-active	Indicates that the interface belongs to the working circuit.
APS-inactive	Indicates that the interface belongs to the protect circuit.
Connection protection	Whether or not connection protection is configured for the Layer 2 circuit to the neighbor: **Yes** or **No**.
VC bandwidth	Bandwidth requirement of the Layer 2 circuit.
Time	Time at which the event occurred.
Connection History	Event types logged in history. • **loc intf up**—Local virtual circuit interface went up. • **loc intf down**—Local virtual circuit interface went down. • **In lbl Update**—Incoming label has been updated. • **Out lbl Update**—Outgoing label has been updated. • **PE route changed**—Route to PE router has been updated. • **PE route down**—Route to PE router is down. • **rmt side marked**—Remote side is marked. • **VC Dn**—Remote side indicated that its end of the virtual circuit is down (if the tunnel LSP from the remote side to the local side is down). • **status update timer**—Status update timer processing. It computes the state of the virtual circuit, and determines whether it should be advertised to or withdrawn from the remote side.

Sample Output

show l2circuit connections

```
user@host> show l2circuit connections
Layer-2 Circuit Connections:

Legend for connection status (St)
EI -- encapsulation invalid      NP -- interface h/w not present
MM -- mtu mismatch               Dn -- down
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
VM -- vlan id mismatch           CF -- Call admission control failure
OL -- no outgoing label          IB -- TDM incompatible bitrate
NC -- intf encaps not CCC/TCC    TM -- TDM misconfiguration
BK -- Backup Connection          ST -- Standby Connection
CB -- rcvd cell-bundle size bad  SP -- Static Pseudowire
LD -- local site signaled down   RS -- remote site standby
RD -- remote site signaled down  HS -- hot standby
XX -- unknown

Legend for interface status
Up -- operational
```

```
Dn -- down
Neighbor: 10.255.245.51
    Interface                 Type  St     Time last up          # Up trans
    ge-2/0/2.600(vc 5)        rmt   Up     Dec  7 18:11:18 2009           1
      Remote PE: 10.255.245.51, Negotiated control-word: No
      Incoming label: 299856, Outgoing label: 299808
      Negotiated PW status TLV: No
      Local interface: ge-2/0/2.600, Status: Up, Encapsulation: VLAN
```

Sample Output

show l2circuit connections interface

```
user@host> show l2circuit connections interface t1-2/0/0:1:1.0
Layer-2 Circuit Connections:

Legend for connection status (St)
EI -- encapsulation invalid      NP -- interface h/w not present
MM -- mtu mismatch               Dn -- down
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
VM -- vlan id mismatch           CF -- Call admission control failure
OL -- no outgoing label          IB -- TDM incompatible bitrate
NC -- intf encaps not CCC/TCC    TM -- TDM misconfiguration
BK -- Backup Connection          ST -- Standby Connection
CB -- rcvd cell-bundle size bad  SP -- Static Pseudowire
LD -- local site signaled down   RS -- remote site standby
RD -- remote site signaled down  HS -- hot standby
XX -- unknown

Legend for interface status
Up -- operational
Dn -- down
Neighbor: 10.1.1.1
    Interface                 Type  St     Time last up          # Up trans
    t1-2/0/0:1:1.0(vc 1)(SP)  rmt   Up     Apr 27 04:21:02 2011           1
      Remote PE: 10.1.1.1, Negotiated control-word: Yes (Non-null)
      Incoming label: 1010001, Outgoing label: 1000001
      Negotiated PW status TLV: No
      Local interface: t1-1/0/0:1:1.0, Status: Up, Encapsulation: SATOP-T1,
APS-active
      Local interface: t1-2/0/0:1:1.0, Status: Up, Encapsulation: SATOP-T1,
APS-inactive
```

Sample Output

show l2circuit connections extensive

```
user@host>show l2circuit connections extensive
Layer-2 Circuit Connections:

Legend for connection status (St)
EI -- encapsulation invalid      NP -- interface h/w not present
MM -- mtu mismatch               Dn -- down
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
VM -- vlan id mismatch           CF -- Call admission control failure
OL -- no outgoing label          IB -- TDM incompatible bitrate
NC -- intf encaps not CCC/TCC    TM -- TDM misconfiguration
BK -- Backup Connection          ST -- Standby Connection
CB -- rcvd cell-bundle size bad  SP -- Static Pseudowire
```



```
LD -- local site signaled down   RS -- remote site standby
RD -- remote site signaled down  HS -- hot standby
XX -- unknown

Legend for interface status
Up -- operational
Dn -- down
Neighbor: 10.255.49.149
    Interface                 Type  St     Time last up          # Up trans
    ae0.0(vc 100)             rmt   Up     Aug 31 09:36:12 2009           1
      Remote PE: 10.255.49.149, Negotiated control-word: Yes (Null)
      Incoming label: 299824, Outgoing label: 299776
      Negotiated PW status TLV: Yes
      local PW status code: 0x00000000, Neighbor PW status code: 0x00000000
      Local interface: ae0.0, Status: Up, Encapsulation: ETHERNET
      Connection protection: Yes
    Connection History:
        Aug 31 09:36:12 2009  status update timer
        Aug 31 09:36:12 2009  PE route changed
        Aug 31 09:36:12 2009  Out lbl Update                    299776
        Aug 31 09:36:12 2009  In lbl Update                     299824
        Aug 31 09:36:12 2009  loc intf up                        ae0.0
```

Sample Output

show l2circuit connections extensive (Pseudowire Redundancy with Hot Standby)

```
user@host>show l2circuit connections extensive
Layer-2 Circuit Connections:

Legend for connection status (St)
EI -- encapsulation invalid      NP -- interface h/w not present
MM -- mtu mismatch               Dn -- down
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
VM -- vlan id mismatch           CF -- Call admission control failure
OL -- no outgoing label          IB -- TDM incompatible bitrate
NC -- intf encaps not CCC/TCC    TM -- TDM misconfiguration
BK -- Backup Connection          ST -- Standby Connection
CB -- rcvd cell-bundle size bad  SP -- Static Pseudowire
LD -- local site signaled down   RS -- remote site standby
RD -- remote site signaled down  HS -- Hot-standby Connection
XX -- unknown

Legend for interface status
Up -- operational
Dn -- down
Neighbor: 88.0.0.101
    Interface                 Type  St     Time last up          # Up trans
    ge-1/3/2.600(vc 1)        rmt   Up     Jan 24 11:00:26 2013           1
      Remote PE: 88.0.0.101, Negotiated control-word: Yes (Null)
      Incoming label: 299776, Outgoing label: 299776
      Negotiated PW status TLV: Yes
      local PW status code: 0x00000000, Neighbor PW status code: 0x00000000
      Local interface: ge-1/3/2.600, Status: Up, Encapsulation: VLAN
    Connection History:
        Jan 24 11:00:26 2013  status update timer
        Jan 24 11:00:26 2013  PE route changed
        Jan 24 11:00:26 2013  Out lbl Update                    299776
        Jan 24 11:00:26 2013  In lbl Update                     299776
        Jan 24 11:00:26 2013  loc intf up                 ge-1/3/2.600
```



```
Neighbor: 88.0.0.102
    Interface                 Type  St     Time last up          # Up trans
    ge-1/3/2.600(vc 2)        rmt   HS     -----                       ----
      Remote PE: 88.0.0.102, Negotiated control-word: Yes (Null)
      Incoming label: 299792, Outgoing label: 299776
      Negotiated PW status TLV: Yes
      local PW status code: 0x00000020, Neighbor PW status code: 0x00000000
      Local interface: ge-1/3/2.600, Status: Up, Encapsulation: VLAN
```

show l2vpn connections

Syntax

show l2vpn connections
<brief | extensive>
<down | up | up-down>
<history>
<instance *instance*>
<instance-history>
<local-site *local-site*>
<logical-system (all | *logical-system-name*)>
<remote-site *remote-site*>
<status>
<summary>

Release Information

Command introduced before Junos OS Release 7.4.
instance-history option introduced in Junos OS Release 12.3R2.

Description

Display Layer 2 virtual private network (VPN) connections.

Options

none—Display all Layer 2 VPN connections for all routing instances.

brief | extensive—(Optional) Display the specified level of output.

down | up | up-down—(Optional) Display nonoperational, operational, or both kinds of connections.

history—(Optional) Display information about connection history.

instance *instance*—(Optional) Display connections for the specified routing instance only.

instance-history—(Optional) Display information about connection history for a particular instance.

local-site *local-site*—(Optional) Display connections for the specified Layer 2 VPN local site name or ID only.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

remote-site *remote-site*—(Optional) Display connection for the specified Layer 2 VPN remote site ID only.

status—(Optional) Display information about the connection and interface status.

summary—(Optional) Display summary of all Layer 2 VPN connections information.

Required Privilege Level

view

List of Sample Output

show l2vpn connections on page 327
show l2vpn connections extensive on page 328
show l2vpn connections extensive (VPWS) on page 328

Output Fields Table 7 on page 325 lists the output fields for the **show l2vpn connections** command. Output fields are listed in the approximate order in which they appear.

Table 7: show l2vpn connections Output Fields

Field Name	Field Description
Instance	Name of Layer 2 VPN instance.
L2vpn-id	For BGP autodiscovery, a globally unique Layer 2 VPN community identifier for the instance.
Local-ID	BGP **local-address** assigned to the local routing device.
Local site	Name of local site.
Local source-attachment-id	For FEC 129, the VPWS source attachment identifier. The point-to-point nature of VPWS requires that you specify the source access individual identifier (SAII) and the target access individual identifier (TAII). This SAII-TAII pair defines a unique pseudowire between two PE devices.
Target-attachment-id	For FEC 129, the VPWS target attachment identifier. If the configured target identifier matches a source identifier advertised by a remote PE device by way of a BGP auto-discovery message, the pseudowire between that source-target pair is signaled. If there is no match between an advertised source identifier and the configured target identifier, the pseudowire is not established.
Interface name	Name of interface.
Remote Site ID	Remote site ID.
Label Offset	Numbers within the label block that are skipped to find the next label base.
Label-base	Advertises the first label in a block of labels. A remote PE router uses this first label when sending traffic toward the advertising PE router.
Range	Advertises the label block size.
status-vector	Bit vector advertising the state of local PE-CE circuits to remote PE routers. A bit value of **0** indicates that the local circuit and LSP tunnel to the remote PE router are up, whereas a value of **1** indicates either one or both are down.
connection-site	Name of the connection site.
Type	Type of connection: **loc** (local) or **rmt** (remote).
St	Status of the connection. (For a list of possible values, see the **Legend for connection status (St)** field.)
Time last up	Time that the connection was last in the **Up** condition.
# Up trans	Number of transitions from **Down** to **Up** condition.

Table 7: show l2vpn connections Output Fields *(continued)*

Field Name	Field Description
Local circuit	Address and status of local circuit.
Remote circuit	Address and status of remote circuit.
St	Status of the Layer 2 VPN connection (corresponds with Legend for Connection Status): • **EI**—The local Layer 2 VPN interface is configured with an encapsulation that is not supported. • **EM**—The encapsulation type received on this Layer 2 VPN connection from the neighbor does not match the local Layer 2 VPN connection interface encapsulation type. • **VC-Dn**—The virtual circuit is currently down. • **CM**—The two routers do not agree on a control word, which causes a control word mismatch. • **CN**—The virtual circuit is not provisioned properly. • **OR**—The label associated with the virtual circuit is out of range. • **OL**—No advertisement has been received for this virtual circuit from the neighbor. There is no outgoing label available for use by this virtual circuit. • **LD**—All of the CE-facing interfaces to the local site are down. Therefore, the connection to the local site is signaled as down to the other PE routers. No pseudowires can be established. • **RD**—All the interfaces to the remote neighbor are down. Therefore, the remote site has been signaled as down to the other PE routers. No pseudowires can be established. • **LN**—The local site has lost path selection to the remote site and therefore no pseduowires can be established from this local site. • **RN**—The remote site has lost path selection to a local site or other remote site and therefore no pseudowires are established to this remote site. • **XX**—The Layer 2 VPN connection is down for an unn reason. This is a programming error. • **NC**—The interface encapsulation is not configured as an appropriate CCC, TCC, or Layer 2 VPN encapsulation. • **WE**—The encapsulation configured for the interface does not match the encapsulation configured for the associated connection within the Layer 2 VPN routing instance. • **NP**—The router detects that interface hardware is not present. The hardware might be offline, a PIC might not be of the desired type, or the interface might be configured in a different routing instance. • **->** —Only the outbound connection is up. • **<-** —Only the inbound connection is up. • **Up**—The Layer 2 VPN connection is operational. • **Dn**—The Layer 2 VPN connection is down. • **CF**—The router cannot find enough bandwidth to the remote router to satisfy the Layer 2 VPN connection bandwidth requirement. • **SC**—The local site identifier matches the remote site identifier. No pseudowire can be established between these two sites. You should configure different values for the local and remote site identifiers.

Table 7: show l2vpn connections Output Fields *(continued)*

Field Name	Field Description
	• **LM**—The local site identifier is not the minimum designated, meaning it is not the lowest. There is another local site with a lower site identifier. Pseudowires are not being established to this local site. and the associated local site identifier is not being used to distribute Layer 2 VPN label blocks. However, this is not an error state. Traffic continues to be forwarded to the PE router interfaces connected to the local sites when the local sites are in this state. • **RM**—The remote site identifier is not the minimum designated, meaning it is not the lowest. There is another remote site connected to the same PE router which has lower site identifier. The PE router cannot established a pseduowire to this remote site and the associated remote site identifier cannot be used to distribute VPLS label blocks. However, this is not an error state. Traffic can continue to be forwarded to the PE router interface connected to this remote site when the remote site is in this state. • **IL**—The incoming packets for the Layer 2 VPN connection have no MPLS label.
Remote PE	Address of the remote provider edge router.
Incoming label	Name of the incoming label.
Outgoing label	Name of the outgoing label.
Time	Date and time of Layer 2 VPN connection event.
Event	Type of event.
Interface/Lbl/PE	Interface, label, or PE router.

Sample Output

show l2vpn connections

```
user@host> show l2vpn connections
L2VPN Connections :
Instance : vpna
Local site: 2 (ce-2)
offset: 1, range: 3, label-base: 32768
    connection-site          Type  St  Time last up          # Up trans
    3 (3)                    loc   Up  Jul 18 20:45:46 2001           1
      Local circuit: fe-0/0/0.1, Status: Up
      Remote circuit: fe-0/0/3.0, Status: Up
    1                        rmt   Up  Jul 18 21:47:25 2001           1
      Local circuit: fe-0/0/0.0, Status: Up
      Remote PE: 192.0.2.1
      Incoming label: 32768, Outgoing label: 32769
Local site: 3 (ce-3)
offset: 1, range: 2, label-base: 33792
    connection-site          Type  St  Time last up          # Up trans
    2 (ce-b)                 loc   Up  Jul 18 20:45:46 2001           1
      Local circuit: fe-0/0/0.1, Status: Up
      Remote circuit: fe-0/0/3.0, Status: Up
    1                        rmt   Up  Jul 18 21:47:25 2001           1
```



```
      Local circuit: fe-0/0/3.1, Status: Up
      Remote PE: 192.0.2.1
      Incoming label: 33792, Outgoing label: 32770
```

show l2vpn connections extensive

```
user@host> show l2vpn connections extensive
L2VPN Connections:

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

Instance: vpn1
  Local site: SITE-A (1)
    Number of local interfaces: 1
    Number of local interfaces up: 1
    ge-1/0/0.517        2
    Label-base        Offset     Size  Range     Preference
    800000            1          2      2         100
      status-vector:  0
    connection-site           Type  St     Time last up          # Up trans
    2                         rmt   Up     Mar  5 18:24:30 2014           1
      Remote PE: 192.0.2.34, Negotiated control-word: Yes (Null)
      Incoming label: 800001, Outgoing label: 800000
      Local interface: ge-0/0/1.517, Status: Up, Encapsulation: VLAN
    Connection History:
        Apr  25 11:18:09 2014  PE route changed
        Apr  25 11:18:09 2014  Out lbl Update                    800000
        Apr  25 11:18:09 2014  In lbl Update                     800001
        Apr  25 11:18:09 2014  loc intf up                 ge-0/0/1.517
```

show l2vpn connections extensive (VPWS)

```
user@host> show l2vpn connections
Layer-2 VPN connections:

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

```
BK -- Backup connection          ST -- Standby connection
PF -- Profile parse failure      PB -- Profile busy
RS -- remote site standby        SN -- Static Neighbor
LB -- Local site not best-site   RB -- Remote site not best-site
VM -- VLAN ID mismatch

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

Instance: FEC129-VPWS
  L2vpn-id: 100:100
    Number of local interfaces: 1
    Number of local interfaces up: 1
    ge-2/0/5.0
  Local source-attachment-id: 1 (ONE)
    Target-attachment-id      Type  St     Time last up          # Up trans
    2                         rmt   Up     Nov 28 16:16:14 2012           1
      Remote PE: 198.51.100.2, Negotiated control-word: No
      Incoming label: 299792, Outgoing label: 299792
      Local interface: ge-2/0/5.0, Status: Up, Encapsulation: ETHERNET
    Connection History:
        Nov 28 16:16:14 2012  status update timer
        Nov 28 16:16:14 2012  PE route changed
        Nov 28 16:16:14 2012  Out lbl Update                    299792
        Nov 28 16:16:14 2012  In lbl Update                     299792
        Nov 28 16:16:14 2012  loc intf up                   ge-2/0/5.0
```


show route

List of Syntax

Syntax on page 330
Syntax (EX Series Switches) on page 330

Syntax

show route
<all>
<*destination-prefix*>
<logical-system (all | *logical-system-name*)>
<private>

Syntax (EX Series Switches)

show route
<all>
<*destination-prefix*>
<private>

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Option **private** introduced in Junos OS Release 9.5.
Option **private** introduced in Junos OS Release 9.5 for EX Series switches.

Description

Display the active entries in the routing tables.

Options

none—Display brief information about all active entries in the routing tables.

all—(Optional) Display information about all routing tables, including private, or internal, routing tables.

destination-prefix—(Optional) Display active entries for the specified address or range of addresses.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

private—(Optional) Display information only about all private, or internal, routing tables.

Required Privilege Level

view

Related Documentation

- *Example: Configuring RIP*
- *Example: Configuring RIPng*
- *Example: Configuring IS-IS*
- *Examples: Configuring Internal BGP Peering*
- *Examples: Configuring External BGP Peering*
- *Examples: Configuring OSPF Routing Policy*

List of Sample Output

show route on page 333
show route on page 334

show route destination-prefix on page 334
show route extensive on page 334

Output Fields Table 8 on page 331 describes the output fields for the **show route** command. Output fields are listed in the approximate order in which they appear.

Table 8: show route Output Fields

<table>
<tr><th>Field Name</th><th>Field Description</th></tr>
<tr><td>routing-table-name</td><td>Name of the routing table (for example, inet.0).</td></tr>
<tr><td>number destinations</td><td>Number of destinations for which there are routes in the routing table.</td></tr>
<tr><td>number routes</td><td>Number of routes in the routing table and total number of routes in the following states:
• active (routes that are active).
• holddown (routes that are in the pending state before being declared inactive). A holddown route was once the active route and is no longer the active route. The route is in the holddown state because a protocol still has interest in the route, meaning that the interest bit is set. A protocol might have its interest bit set on the previously active route because the protocol is still advertising the route. The route will be deleted after all protocols withdraw their advertisement of the route and remove their interest bit. A persistent holddown state often means that the interested protocol is not releasing its interest bit properly.
However, if you have configured advertisement of multiple routes (with the add-path or advertise-inactive statement), the holddown bit is most likely set because BGP is advertising the route as an active route. In this case, you can ignore the holddown state because nothing is wrong.
• hidden (routes that are not used because of a routing policy).</td></tr>
<tr><td>destination-prefix</td><td>Route destination (for example:10.0.0.1/24). Sometimes the route information is presented in another format, such as:
• MPLS-label (for example, 80001).
• interface-name (for example, ge-1/0/2).
• neighbor-address:control-word-status:encapsulation type:vc-id:source (Layer 2 circuit only. For example, 10.1.1.195:NoCtrlWord:1:1:Local/96):
  • neighbor-address—Address of the neighbor.
  • control-word-status—Whether the use of the control word has been negotiated for this virtual circuit: NoCtrlWord or CtrlWord.
  • encapsulation type—Type of encapsulation, represented by a number: (1) Frame Relay DLCI, (2) ATM AAL5 VCC transport, (3) ATM transparent cell transport, (4) Ethernet, (5) VLAN Ethernet, (6) HDLC, (7) PPP, (8) ATM VCC cell transport, (10) ATM VPC cell transport.
  • vc-id—Virtual circuit identifier.
  • source—Source of the advertisement: Local or Remote.</td></tr>
</table>

Table 8: show route Output Fields *(continued)*

Field Name	Field Description
[*protocol*, *preference*]	Protocol from which the route was learned and the preference value for the route. • **+**—A plus sign indicates the active route, which is the route installed from the routing table into the forwarding table. • **-** —A hyphen indicates the last active route. • *****—An asterisk indicates that the route is both the active and the last active route. An asterisk before a **to** line indicates the best subpath to the route. In every routing metric except for the BGP **LocalPref** attribute, a lesser value is preferred. In order to use common comparison routines, Junos OS stores the 1's complement of the **LocalPref** value in the **Preference2** field. For example, if the **LocalPref** value for Route 1 is 100, the **Preference2** value is -101. If the **LocalPref** value for Route 2 is 155, the **Preference2** value is -156. Route 2 is preferred because it has a higher **LocalPref** value and a lower **Preference2** value.
weeks:days hours:minutes:seconds	How long the route been known (for example, **2w4d 13:11:14**, or 2 weeks, 4 days, 13 hours, 11 minutes, and 14 seconds).
metric	Cost value of the indicated route. For routes within an AS, the cost is determined by the IGP and the individual protocol metrics. For external routes, destinations, or routing domains, the cost is determined by a preference value.
localpref	Local preference value included in the route.
from	Interface from which the route was received.
AS path	AS path through which the route was learned. The letters at the end of the AS path indicate the path origin, providing an indication of the state of the route at the point at which the AS path originated: • **I**—IGP. • **E**—EGP. • **?**—Incomplete; typically, the AS path was aggregated. When AS path numbers are included in the route, the format is as follows: • **[ ]**—Brackets enclose the local AS number associated with the AS path if more than one AS number is configured on the routing device, or if AS path prepending is configured. • **{ }**—Braces enclose AS sets, which are groups of AS numbers in which the order does not matter. A set commonly results from route aggregation. The numbers in each AS set are displayed in ascending order. • **( )**—Parentheses enclose a confederation. • **([ ])**—Parentheses and brackets enclose a confederation set. NOTE: In Junos OS Release 10.3 and later, the AS path field displays an unrecognized attribute and associated hexadecimal value if BGP receives attribute 128 (attribute set) and you have not configured an independent domain in any routing instance.

Table 8: show route Output Fields *(continued)*

Field Name	Field Description
validation-state	(BGP-learned routes) Validation status of the route: • **Invalid**—Indicates that the prefix is found, but either the corresponding AS received from the EBGP peer is not the AS that appears in the database, or the prefix length in the BGP update message is longer than the maximum length permitted in the database. • **Unknown**—Indicates that the prefix is not among the prefixes or prefix ranges in the database. • **Unverified**—Indicates that the origin of the prefix is not verified against the database. This is because the database got populated and the validation is not called for in the BGP import policy, although origin validation is enabled, or the origin validation is not enabled for the BGP peers. • **Valid**—Indicates that the prefix and autonomous system pair are found in the database.
to	Next hop to the destination. An angle bracket (>) indicates that the route is the selected route. If the destination is **Discard**, traffic is dropped.
via	Interface used to reach the next hop. If there is more than one interface available to the next hop, the interface that is actually used is followed by the word **Selected**. This field can also contain the following information: • **Weight**—Value used to distinguish primary, secondary, and fast reroute backup routes. Weight information is available when MPLS label-switched path (LSP) link protection, node-link protection, or fast reroute is enabled, or when the standby state is enabled for secondary paths. A lower weight value is preferred. Among routes with the same weight value, load balancing is possible. • **Balance**—Balance coefficient indicating how traffic of unequal cost is distributed among next hops when a routing device is performing unequal-cost load balancing. This information is available when you enable BGP multipath load balancing. • ***lsp-path-name***—Name of the LSP used to reach the next hop. • ***label-action***—MPLS label and operation occurring at the next hop. The operation can be **pop** (where a label is removed from the top of the stack), **push** (where another label is added to the label stack), or **swap** (where a label is replaced by another label). For VPNs, expect to see multiple **push** operations, corresponding to the inner and outer labels required for VPN routes (in the case of a direct PE-to-PE connection, the VPN route would have the inner label push only).

Sample Output

show route

```
user@host> show route
inet.0: 11 destinations, 12 routes (11 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1:65500:1:10.0.0.20/240
                   *[MVPN/70] 19:53:41, metric2 1
                      Indirect
1:65500:1:10.0.0.40/240
                   *[BGP/170] 19:53:29, localpref 100, from 10.0.0.30
                      AS path: I
                    > to 10.0.24.4 via lt-0/3/0.24, label-switched-path toD
                    [BGP/170] 19:53:26, localpref 100, from 10.0.0.33
                      AS path: I
                    > to 10.0.24.4 via lt-0/3/0.24, label-switched-path toD
1:65500:1:10.0.0.60/240
                   *[BGP/170] 19:53:29, localpref 100, from 10.0.0.30
```

```
                                AS path: I
                              > to 10.0.28.8 via lt-0/3/0.28, label-switched-path toF
                              [BGP/170] 19:53:25, localpref 100, from 10.0.0.33
                                AS path: I
                              > to 10.0.28.8 via lt-0/3/0.28, label-switched-path toF
```

show route

The following sample output shows a VPN route with composite next hops enabled. The first **Push** operation corresponds to the outer label. The second **Push** operation corresponds to the inner label.

```
user@host> show route 70.0.0.0

13979:665001.inet.0: 871 destinations, 3556 routes (871 active, 0 holddown, 0
hidden)
+ = Active Route, - = Last Active, * = Both

70.0.0.0/24        @[BGP/170] 00:28:32, localpref 100, from 10.9.9.160
                      AS path: 13980 ?, validation-state: unverified
                    > to 10.100.0.42 via ae2.0, Push 16, Push 300368(top)
                    [BGP/170] 00:28:28, localpref 100, from 10.9.9.169
                      AS path: 13980 ?, validation-state: unverified
                    > to 10.100.0.42 via ae2.0, Push 126016, Push 300368(top)
                   #[Multipath/255] 00:28:28, metric2 102
                    > to 10.100.0.42 via ae2.0, Push 16, Push 300368(top)
                      to 10.100.0.42 via ae2.0, Push 16, Push 300368(top)
```

show route destination-prefix

```
user@host> show route 172.16.0.0/12

inet.0: 10 destinations, 10 routes (9 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

172.16.0.0/12      *[Static/5] 2w4d 12:54:27
                    > to 192.168.167.254 via fxp0.0
```

show route extensive

```
user@host> show route extensive
v1.mvpn.0: 5 destinations, 8 routes (5 active, 1 holddown, 0 hidden)
1:65500:1:10.0.0.40/240 (1 entry, 1 announced)
    *BGP    Preference: 170/-101
          PMSI: Flags 0x0: Label[0:0:0]: PIM-SM: Sender 10.0.0.40 Group 225.1.1.1

           Next hop type: Indirect
           Address: 0x92455b8
           Next-hop reference count: 2
           Source: 10.0.0.30
           Protocol next hop: 10.0.0.40
           Indirect next hop: 2 no-forward
           State: <Active Int Ext>
                     Local AS: 65500 Peer AS: 65500
           Age: 3  Metric2: 1
          Validation State: unverified
           Task: BGP_65500.10.0.0.30+179
           Announcement bits (2): 0-PIM.v1 1-mvpn global task
           AS path: I (Originator) Cluster list:  10.0.0.30
           AS path:  Originator ID: 10.0.0.40
           Communities: target:65520:100
```

```
Import Accepted
Localpref: 100
Router ID: 10.0.0.30
Primary Routing Table bgp.mvpn.0
Indirect next hops: 1
     Protocol next hop: 10.0.0.40 Metric: 1
     Indirect next hop: 2 no-forward
     Indirect path forwarding next hops: 1
           Next hop type: Router
           Next hop: 10.0.24.4 via lt-0/3/0.24 weight 0x1
     10.0.0.40/32 Originating RIB: inet.3
           Metric: 1                       Node path count: 1
           Forwarding nexthops: 1
              Nexthop: 10.0.24.4 via lt-0/3/0.24
```

CHAPTER 9

Operational Commands

ping mpls l2circuit

Syntax

ping mpls l2circuit (interface *interface-name* | virtual-circuit *virtual-circuit-id* neighbor *address*)
<count *count*>
<destination *address*>
<detail>
<exp *forwarding-class*>
<logical-system (all | *logical-system-name*)>
reply-mode (application-level-control-channel | ip-udp | no-reply)
<size *bytes*>
<source *source-address*>
<sweep>
<v1>

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
The **size** and **sweep** options were introduced in Junos OS Release 9.6.
The **reply-mode** option and its suboptions are introduced in Junos OS Release 10.4R1.

Description

Check the operability of the MPLS Layer 2 circuit connections. Type Ctrl+c to interrupt a **ping mpls l2circuit** command. You can also issue this command within logical systems.

Options

count ***count***—(Optional) Number of ping requests to send. If **count** is not specified, five ping requests are sent. The range of values is 1 through 1,000,000. The default value is 5.

destination ***address***—(Optional) Specify an address other than the default (127.0.0.1/32) for the ping echo requests. The address can be anything within the 127/8 subnet.

detail—(Optional) Display detailed information about the echo requests sent and received.

exp ***forwarding-class***—(Optional) Value of the forwarding class for the MPLS ping packets.

interface ***interface-name***—Ping an interface configured for the Layer 2 circuit on the egress provider edge (PE) router.

logical-system (all | ***logical-system-name*****)**—(Optional) Perform this operation on all logical systems or on the specified logical system.

reply-mode—(Optional) Reply mode for the ping request. This option has the following suboptions:

application-level-control-channel—Reply using an application level control channel.

ip-udp—Reply using an IPv4 or IPv6 UDP packet.

no-reply—Do not reply to the ping request.

NOTE: The **reply-mode** option and its suboptions **application-level-control-channel**, **ip-udp**, and **no-reply** are also available in Junos OS Release 10.2R4 and 10.3R2.

size *bytes*—(Optional) Size of the label-switched path (LSP) ping request packet (96 through 65468 bytes). Packets are 4-byte aligned. For example, If you enter a size of 97, 98, 99, or 100, the router or switch uses a size value of 100 bytes. If you enter a packet size that is smaller than the minimum size, an error message is displayed reminding you of the 96-byte minimum.

source *source-address*—(Optional) IP address of the outgoing interface. This address is sent in the IP source address field of the ping request. If this option is not specified, the default address is usually the loopback interface (lo.0).

sweep—(Optional) Automatically determine the size of the maximum transmission unit (MTU).

v1—(Optional) Use the type 9 Layer 2 circuit type, length, and value (TLV).

virtual-circuit *virtual-circuit-id* neighbor *address*—Ping the virtual circuit identifier on the egress PE router or switch and the specified neighbor, testing the integrity of the Layer 2 circuit between the ingress and egress PE routers or switches.

Additional Information You must configure MPLS at the **[edit protocols mpls]** hierarchy level on the egress PE router or switch (the router or switch receiving the MPLS echo packets) to ping a Layer 2 circuit.

In asymmetric MTU scenarios, the echo response may be dropped. For example, if the MTU from System A to System B is 1000 bytes, the MTU from System B to System A is 500 bytes, and the ping request packet size is 1000 bytes, the echo response is dropped because the PAD TLV is included in the echo response, making it too large.

Required Privilege Level network

List of Sample Output
ping mpls l2circuit interface on page 339
ping mpls l2circuit virtual-circuit detail on page 339
ping mpls l2circuit interface <interface-name> reply-mode on page 340

Output Fields When you enter this command, you are provided feedback on the status of your request. An exclamation point (**!**) indicates that an echo reply was received. A period (**.**) indicates that an echo reply was not received within the timeout period. An **x** indicates that an echo reply was received with an error code. Packets with an error code are not counted in the received packets count. They are accounted for separately.

Sample Output

ping mpls l2circuit interface

```
user@host> ping mpls l2circuit interface so-1/0/0.1
Request for seq 1, to interface 69, labels <100000, 100208>, packet size 100
Reply for seq 1, return code: Egress-ok, time: 0.439 ms
```

ping mpls l2circuit virtual-circuit detail

```
user@host> ping mpls l2circuit virtual-circuit 200 neighbor 10.255.245.122/32 detail
```

```
Request for seq 1, to interface 68, labels <100048, 100128>, packet size 100

Reply for seq 1, return code: Egress-ok time: 0.539 ms
```

ping mpls l2circuit interface <interface-name> reply-mode

```
user@host> ping mpls l2circuit interface lt-1/2/0.21 reply-mode application-level-control-channel
!!!!!
--- lsping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
```


request l2circuit-switchover

Syntax

request l2circuit-switchover
<logical-system (all | logical-system-name)>
<neighbor *address*>
<virtual-circuit-id *identifier*>

Release Information

Command introduced in Junos OS Release 9.2.

Description

Manually trigger a switch from the active pseudowire to the redundant pseudowire. This command can be useful when performing network maintenance.

Options

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

neighbor *address*—(Optional) Trigger a switch of all of the active pseudowire connections with the specified neighbor to their respective redundant pseudowires.

virtual-circuit-id *identifier*—(Optional) Trigger a switch from the active pseudowire connection of the specified Layer 2 circuit to its redundant pseudowire.

Required Privilege Level

maintenance

List of Sample Output

request l2circuit-switchover virtual-circuit-id on page 341

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

request l2circuit-switchover virtual-circuit-id

`user@host>` **request l2circuit-switchover virtual-circuit-id 12**

show l2circuit connections

Syntax

show l2circuit connections
<brief | extensive | summary>
<down | up | up-down>
<history>
<interface *interface-name*>
<logical-system (all | *logical-system-name*)>
<neighbor *neighbor*>
<status>

Release Information

Command introduced before Junos OS Release 7.4.
Display enhancements in Junos OS Release 9.6.
Display enhancements in Junos OS Release 10.2.
Display enhancements in Junos OS Release 12.1.
Display enhancements in Junos OS Release 13.2.

Description

Display status information about Layer 2 virtual circuits from the local provider edge (PE) router to its neighbors.

Options

none—Display standard information about Layer 2 virtual circuits on all interfaces for all neighbors.

brief | extensive | summary—(Optional) Display the specified level of output. Use history to display information about connection history. Use status to display information about the connection and interface status.

down | up | up-down—(Optional) Display nonoperational, operational, or both kinds of connections.

history—(Optional) Display information about connection history.

interface *interface-name*—(Optional) Show all Layer 2 virtual circuits on an interface.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

neighbor *neighbor*—(Optional) IP address of a specific neighbor.

status—(Optional) Display information about the connection and interface status.

Required Privilege Level

view

List of Sample Output

show l2circuit connections on page 345
show l2circuit connections interface on page 346
show l2circuit connections extensive on page 346
show l2circuit connections extensive (Pseudowire Redundancy with Hot Standby) on page 347

Output Fields

Table 6 on page 318 lists the output fields for the **show l2circuit connections** command. Output fields are listed in the approximate order in which they appear.

Table 9: show l2circuit connections Output Fields

Field Name	Field Description
Layer-2 Circuit Connections	Displays the legends for connection and interface status.
Neighbor	Remote PE neighbor.
Interface	Logical PE-to-CE interface on which the virtual circuit is configured.
Type	VC type: **rmt** (remote) or **loc** (local).
Legend for connection status (St)	Status of the virtual circuit connection: • **EI**—The local virtual circuit interface is configured with an encapsulation that is not supported. • **MM**—The two routers do not agree on an MTU value, which causes an MTU mismatch. • **EM**—The encapsulation type received on this virtual circuit from the neighbor does not match the local virtual circuit interface encapsulation type. • **CM**—The two routers do not agree on a control word, which causes a control word mismatch. • **VM**—The remote and local VLAN IDs do not match across the Layer 2 circuit. • **OL**—No advertisement has been received for this virtual circuit from the neighbor. There is no outgoing label available for use by this virtual circuit. • **NC**—The interface is not configured as a CCC or TCC interface. • **BK**—The virtual circuit has switched to a backup connection. • **CB**—The remote PE router is advertising a different cell bundle from that configured on the local PE router. • **LD**—The connection to the local site is signaled down, because the CE-facing interface to the local site is down. • **RD**—The remote neighbor is down. It has signaled a problem using the pseudowire status code. • **NP**—The router detects that interface hardware is not present. The hardware may be offline, a PIC may not be of the desired type, or the interface may be configured in a different routing instance. • **Dn**—The virtual circuit is down. • **VC-Dn**—The virtual circuit is down because there is no tunnel LSP from the local PE router to the neighbor. • **UP**—The virtual circuit is operational. • **CF**—The router cannot find enough bandwidth to the remote router to satisfy the Layer 2 circuit bandwidth requirement. • **IB**—The bit rate is incompatible for Time Division Multiplexing (TDM). • **TDM**—TDM is not configured correctly. • **ST**—The virtual circuit has been switched to a standby connection. • **SP**—The virtual circuit connection is using a static pseudowire. • **RS**—The remote site is in a standby state. • **XX**—The virtual circuit is down for an unknown reason. This is a programming error.
Time last up	Date and time the virtual circuit was last operational.

Table 9: show l2circuit connections Output Fields *(continued)*

Field Name	Field Description
# Up trans	Number of times the virtual circuit came up.
local-interface-name	Name of the local PE-to-CE interface.
Status	Status of the local interface.
Up	Interface is operational.
Dn	Interface is not operational.
NP	Not present. Interface does not exist.
DS	Disabled. Interface has been administratively disabled.
WE	Wrong encapsulation. The interface is not configured as CCC.
UN	Interface status is initialized.
Encapsulation	Encapsulation of the local interface.
Remote PE	Prefix of the remote PE router.
Negotiated control-word	Whether the use of the control word has been negotiated for this virtual circuit: **Yes (Null)** or **No**.
Incoming label	Label used by the remote side of the virtual circuit to send packets destined to the local side. This label is routed to the local virtual circuit interface.
Outgoing label	Label used by the local side of the virtual circuit to send packets to the remote side of the virtual circuit. Packets originated on the local virtual circuit interface are encapsulated with this label before being placed on the tunnel LSP to the neighbor for this virtual circuit. This label is allocated by the neighbor and is used in demultiplexing incoming packets destined for this virtual circuit.
Negotiated PW status TLV	Displays the pseudowire status type, length, and value (TLV). TLVs are a method of encoding variable-length or optional information. If the pseudowire status TLV is used, the corresponding local or neighbor PE router status code is also displayed.
local PW status code	If the pseudowire status TLV is used, displays the local PE router status code.
Neighbor PW status code	If the pseudowire status TLV is used, displays the neighbor PE router status code.
Local interface	Name of the local interface used for the Layer 2 circuit connection.
Status	Status of the local interface (**Up** or **Down**).

Table 9: show l2circuit connections Output Fields *(continued)*

Field Name	Field Description
Encapsulation	Encapsulation configured for the local interface.
APS-active	Indicates that the interface belongs to the working circuit.
APS-inactive	Indicates that the interface belongs to the protect circuit.
Connection protection	Whether or not connection protection is configured for the Layer 2 circuit to the neighbor: **Yes** or **No**.
VC bandwidth	Bandwidth requirement of the Layer 2 circuit.
Time	Time at which the event occurred.
Connection History	Event types logged in history. • **loc intf up**—Local virtual circuit interface went up. • **loc intf down**—Local virtual circuit interface went down. • **In lbl Update**—Incoming label has been updated. • **Out lbl Update**—Outgoing label has been updated. • **PE route changed**—Route to PE router has been updated. • **PE route down**—Route to PE router is down. • **rmt side marked**—Remote side is marked. • **VC Dn**—Remote side indicated that its end of the virtual circuit is down (if the tunnel LSP from the remote side to the local side is down). • **status update timer**—Status update timer processing. It computes the state of the virtual circuit, and determines whether it should be advertised to or withdrawn from the remote side.

Sample Output

show l2circuit connections

```
user@host> show l2circuit connections
Layer-2 Circuit Connections:

Legend for connection status (St)
EI -- encapsulation invalid      NP -- interface h/w not present
MM -- mtu mismatch               Dn -- down
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
VM -- vlan id mismatch           CF -- Call admission control failure
OL -- no outgoing label          IB -- TDM incompatible bitrate
NC -- intf encaps not CCC/TCC    TM -- TDM misconfiguration
BK -- Backup Connection          ST -- Standby Connection
CB -- rcvd cell-bundle size bad  SP -- Static Pseudowire
LD -- local site signaled down   RS -- remote site standby
RD -- remote site signaled down  HS -- hot standby
XX -- unknown

Legend for interface status
Up -- operational
```

```
Dn -- down
Neighbor: 10.255.245.51
    Interface                 Type  St     Time last up          # Up trans
    ge-2/0/2.600(vc 5)        rmt   Up     Dec  7 18:11:18 2009           1
      Remote PE: 10.255.245.51, Negotiated control-word: No
      Incoming label: 299856, Outgoing label: 299808
      Negotiated PW status TLV: No
      Local interface: ge-2/0/2.600, Status: Up, Encapsulation: VLAN
```

Sample Output

show l2circuit connections interface

```
user@host> show l2circuit connections interface t1-2/0/0:1:1.0
Layer-2 Circuit Connections:

Legend for connection status (St)
EI -- encapsulation invalid      NP -- interface h/w not present
MM -- mtu mismatch               Dn -- down
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
VM -- vlan id mismatch           CF -- Call admission control failure
OL -- no outgoing label          IB -- TDM incompatible bitrate
NC -- intf encaps not CCC/TCC    TM -- TDM misconfiguration
BK -- Backup Connection          ST -- Standby Connection
CB -- rcvd cell-bundle size bad  SP -- Static Pseudowire
LD -- local site signaled down   RS -- remote site standby
RD -- remote site signaled down  HS -- hot standby
XX -- unknown

Legend for interface status
Up -- operational
Dn -- down
Neighbor: 10.1.1.1
    Interface                 Type  St     Time last up          # Up trans
    t1-2/0/0:1:1.0(vc 1)(SP)  rmt   Up     Apr 27 04:21:02 2011           1
      Remote PE: 10.1.1.1, Negotiated control-word: Yes (Non-null)
      Incoming label: 1010001, Outgoing label: 1000001
      Negotiated PW status TLV: No
      Local interface: t1-1/0/0:1:1.0, Status: Up, Encapsulation: SATOP-T1,
APS-active
      Local interface: t1-2/0/0:1:1.0, Status: Up, Encapsulation: SATOP-T1,
APS-inactive
```

Sample Output

show l2circuit connections extensive

```
user@host>show l2circuit connections extensive
Layer-2 Circuit Connections:

Legend for connection status (St)
EI -- encapsulation invalid      NP -- interface h/w not present
MM -- mtu mismatch               Dn -- down
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
VM -- vlan id mismatch           CF -- Call admission control failure
OL -- no outgoing label          IB -- TDM incompatible bitrate
NC -- intf encaps not CCC/TCC    TM -- TDM misconfiguration
BK -- Backup Connection          ST -- Standby Connection
CB -- rcvd cell-bundle size bad  SP -- Static Pseudowire
```

```
LD -- local site signaled down   RS -- remote site standby
RD -- remote site signaled down  HS -- hot standby
XX -- unknown

Legend for interface status
Up -- operational
Dn -- down
Neighbor: 10.255.49.149
    Interface                 Type  St     Time last up          # Up trans
    ae0.0(vc 100)             rmt   Up     Aug 31 09:36:12 2009           1
      Remote PE: 10.255.49.149, Negotiated control-word: Yes (Null)
      Incoming label: 299824, Outgoing label: 299776
      Negotiated PW status TLV: Yes
      local PW status code: 0x00000000, Neighbor PW status code: 0x00000000
      Local interface: ae0.0, Status: Up, Encapsulation: ETHERNET
      Connection protection: Yes
    Connection History:
        Aug 31 09:36:12 2009  status update timer
        Aug 31 09:36:12 2009  PE route changed
        Aug 31 09:36:12 2009  Out lbl Update                    299776
        Aug 31 09:36:12 2009  In lbl Update                     299824
        Aug 31 09:36:12 2009  loc intf up                        ae0.0
```

Sample Output

show l2circuit connections extensive (Pseudowire Redundancy with Hot Standby)

```
user@host>show l2circuit connections extensive
Layer-2 Circuit Connections:

Legend for connection status (St)
EI -- encapsulation invalid      NP -- interface h/w not present
MM -- mtu mismatch               Dn -- down
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
VM -- vlan id mismatch           CF -- Call admission control failure
OL -- no outgoing label          IB -- TDM incompatible bitrate
NC -- intf encaps not CCC/TCC    TM -- TDM misconfiguration
BK -- Backup Connection          ST -- Standby Connection
CB -- rcvd cell-bundle size bad  SP -- Static Pseudowire
LD -- local site signaled down   RS -- remote site standby
RD -- remote site signaled down  HS -- Hot-standby Connection
XX -- unknown

Legend for interface status
Up -- operational
Dn -- down
Neighbor: 88.0.0.101
    Interface                 Type  St     Time last up          # Up trans
    ge-1/3/2.600(vc 1)        rmt   Up     Jan 24 11:00:26 2013           1
      Remote PE: 88.0.0.101, Negotiated control-word: Yes (Null)
      Incoming label: 299776, Outgoing label: 299776
      Negotiated PW status TLV: Yes
      local PW status code: 0x00000000, Neighbor PW status code: 0x00000000
      Local interface: ge-1/3/2.600, Status: Up, Encapsulation: VLAN
    Connection History:
        Jan 24 11:00:26 2013  status update timer
        Jan 24 11:00:26 2013  PE route changed
        Jan 24 11:00:26 2013  Out lbl Update                    299776
        Jan 24 11:00:26 2013  In lbl Update                     299776
        Jan 24 11:00:26 2013  loc intf up                 ge-1/3/2.600
```

```
Neighbor: 88.0.0.102
    Interface                 Type  St     Time last up          # Up trans
    ge-1/3/2.600(vc 2)        rmt   HS     -----                       ----
      Remote PE: 88.0.0.102, Negotiated control-word: Yes (Null)
      Incoming label: 299792, Outgoing label: 299776
      Negotiated PW status TLV: Yes
      local PW status code: 0x00000020, Neighbor PW status code: 0x00000000
      Local interface: ge-1/3/2.600, Status: Up, Encapsulation: VLAN
```

PART 4

Troubleshooting

CHAPTER 10

Troubleshooting Layer 2 Circuits

Tracing Layer 2 Circuit Operations

To trace the creation of and changes to Layer 2 circuits, 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 protocols l2circuit]**
- **[edit logical-systems** ***logical-system-name*** **protocols l2circuit]**

Specify the following flags to trace the indicated operations on Layer 2 circuits:

- **connections**—Layer 2 circuit connections (events and state changes)
- **error**—Error conditions
- **FEC**—Layer 2 circuit advertisements received or sent using LDP
- **topology**—Layer 2 circuit topology changes caused by reconfiguration or advertisements received from other PE routers

PART 5

Index

Index