JUNIPER | Engineering
NETWORKS | Simplicity

Junos® OS

Layer 2 VPNs User Guide for EX9200 Switches

JUNOS

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

Table of Contents

Configuring Layer 2 Circuits

2

3   **Configuration Statements and Operational Commands**

About This Guide

1
PART

Common Configuration for Layer 2 VPNs

CHAPTER 1

Overview

Understanding Layer 2 VPNs

As the need to link different Layer 2 services to one another for expanded service offerings grows, Layer 2 Multiprotocol Label Switching (*MPLS*) *VPN* services are increasingly in demand.

Implementing a Layer 2 VPN on a router is similar to implementing a VPN using other Layer 2 technologies. However, for a Layer 2 VPN on a router, traffic is forwarded to the router in a Layer 2 format. It is carried by MPLS over the service provider's network, and then converted back to Layer 2 format at the receiving site. You can configure different Layer 2 formats at the sending and receiving sites. The service provisioned with Layer 2 VPNs is also known as *Virtual Private Wire Service* (*VPWS*).

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

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

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

Figure 1: Layer 2 VPN Connecting CE Routers
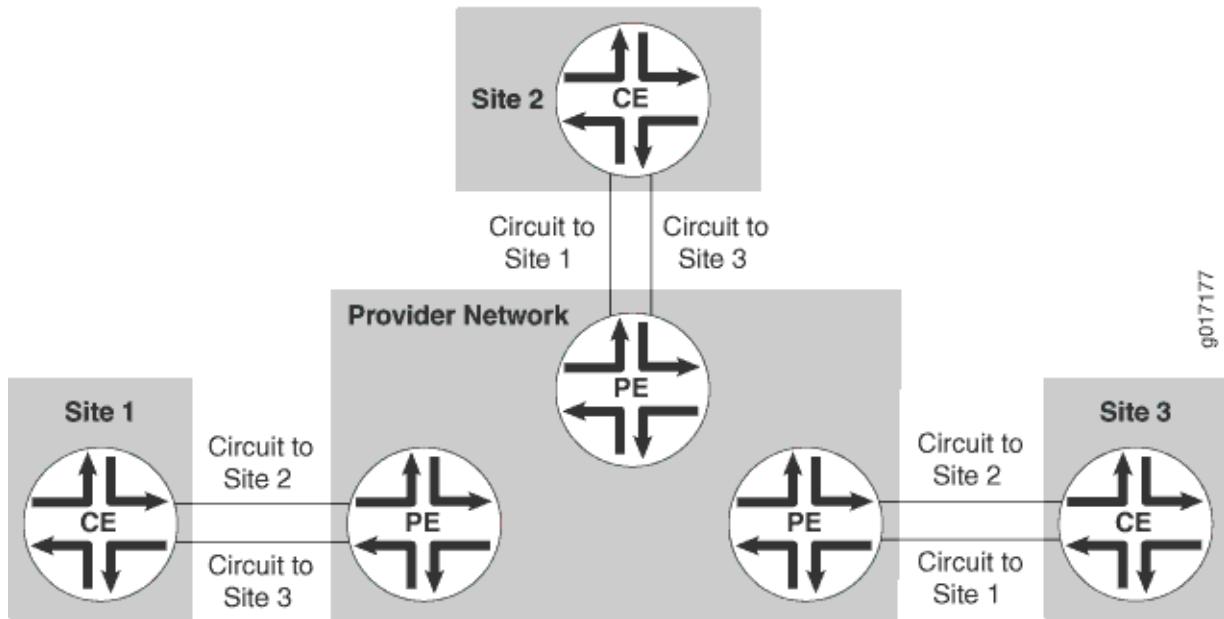
Implementing a Layer 2 MPLS VPN includes the following benefits:

- Service providers do not have to invest in separate Layer 2 equipment to provide Layer 2 VPN service. A Layer 2 MPLS VPN allows you to provide Layer 2 VPN service over an existing IP and MPLS backbone.

- You can configure the PE router to run any Layer 3 protocol in addition to the Layer 2 protocols.

- Customers who prefer to maintain control over most of the administration of their own networks might want Layer 2 VPN connections with their service provider instead of a Layer 3 VPN.

- Because Layer 2 VPNs use *BGP* as the signaling protocol, they have a simpler design and require less overhead than traditional VPNs over Layer 2 circuits. BGP signaling also enables autodiscovery of Layer 2 VPN peers. Layer 2 VPNs are similar to BGP or MPLS VPNs and *VPLS* in many respects; all three types of services employ BGP for signaling.

Layer 2 VPN Applications

Implementing a Layer 2 VPN includes the following benefits:

- Terminating a Layer 2 VPN into a Layer 2 VPN using the interworking (iw0) software interface eliminates the limitation of bandwidth on the tunnel interfaces used for these configuration scenarios. Instead of using a physical Tunnel PIC for looping the packet received from the Layer 2 VPN to another Layer 2 VPN, Junos OS is used to link both the Layer 2 VPN routes.

- Layer 2 VPNs enable the sharing of a provider's core network infrastructure between IP and Layer 2 VPN services, reducing the cost of providing those services. A Layer 2 MPLS VPN allows you to provide Layer 2 VPN service over an existing IP and MPLS backbone.

- From a service provider's point of view, a Layer 2 MPLS VPN allows the use of a single Layer 3 VPN (such as RFC 2547bis), MPLS traffic engineering, and Differentiated Services (DiffServ).

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

RELATED DOCUMENTATION

Understanding Layer 2 VPNs

Using the Layer 2 Interworking Interface to Interconnect a Layer 2 Circuit to a Layer 2 VPN

Using the Layer 2 Interworking Interface to Interconnect a Layer 2 VPN to a Layer 2 VPN

Example: Interconnecting a Layer 2 Circuit with a Layer 2 VPN

Example: Interconnecting a Layer 2 VPN with a Layer 2 VPN

Example: Interconnecting a Layer 2 VPN with a Layer 3 VPN

Supported Layer 2 VPN Standards

Junos OS substantially supports the following standards and Internet drafts, which define standards for Layer 2 virtual private networks (VPNs).

- RFC 7348, *Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks*

- Internet draft draft-kompella-l2vpn-vpls-multihoming, *Multi-homing in BGP-based Virtual Private LAN Service*

- Internet draft draft-kompella-ppvpn-l2vpn-03.txt, *Layer 2 VPNs Over Tunnels*

RELATED DOCUMENTATION

Supported Carrier-of-Carriers and Interprovider VPN Standards

Supported VPWS Standards

CHAPTER 2

Pinging VPNs

IN THIS CHAPTER

Pinging VPNs, VPLS, and Layer 2 Circuits

For testing purposes, you can ping Layer 2 VPNs, Layer 3 VPNs, and Layer 2 circuits by using the `ping mpls` command. The `ping mpls` command helps to verify that a VPN or circuit has been enabled and tests the integrity of the VPN or Layer 2 circuit connection between the PE routers. It does not test the connection between a PE router and a CE router. To ping a VPLS routing instance, you issue a `ping vpls instance` command (see *Pinging a VPLS Routing Instance*).

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

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

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

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

The `ping mpls` command has the following limitations:

- You cannot ping an IPv6 destination prefix.

- You cannot ping a VPN or Layer 2 circuit from a router that is attempting a graceful restart.

- You cannot ping a VPN or Layer 2 circuit from a logical system.

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

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

Pinging a Layer 2 VPN

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

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

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

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

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

RELATED DOCUMENTATION

Example: Configure MPLS-Based Layer 2 VPNs

Pinging a Layer 2 Circuit

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

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

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

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

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

Layer 2 VPNs Configuration Overview

IN THIS CHAPTER

Introduction to Configuring Layer 2 VPNs

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

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

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

To configure Layer 2 VPNs, include the following statements:

```
description text;
instance-type l2vpn;
interface interface-name;
route-distinguisher (as-number:id| ip-address:id);
vrf-export [ policy-names ];
vrf-import [ policy-names ];
vrf-target {
    community;
    import community-name;
```

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

You can include these statements at the following hierarchy levels:

- `[edit routing-instances routing-instance-name]`

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

For Layer 2 VPNs, only some of the statements in the `[edit routing-instances]` hierarchy are valid. For the full hierarchy, see Junos OS Routing Protocols Library.

In addition to these statements, you must configure MPLS label-switched paths (LSPs) between the PE routers, IBGP sessions between the PE routers, and an interior gateway protocol (IGP) on the PE and provider (P) routers. You must also configure the statements that are required for all types of VPN configuration.

By default, Layer 2 VPNs are disabled.

Many of the configuration procedures for Layer 2 VPNs are identical to the procedures for Layer 3 VPNs and virtual private LAN service (VPLS).

Configuring the Local Site on PE Routers in Layer 2 VPNs

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

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

> (i) **NOTE**: Not all subtasks are supported on all platforms; check the CLI on your device.

Configuring a Layer 2 VPN Routing Instance

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

> (i) **NOTE**: On the EX9200 switches, replace *encapsulation-type* with the *encapsulation* statement.

```
l2vpn {
    (control-word | no-control-word);
    encapsulation-type type;
    traceoptions {
        file filename <files number> <size size> <world-readable | no-world-readable>;
```

```
        flag flag <flag-modifier> <disable>;
    }
    site site-name {
        site-identifier identifier;
        site-preference preference-value {
            backup;
            primary;
        }
        interface interface-name {
            description text;
            remote-site-id remote-site-id;
        }
    }
}
```

You can include this statement at the following hierarchy levels:

- `[edit routing-instances routing-instance-name protocols]`

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

> (i) **NOTE**: You cannot configure a routing protocol (OSPF, RIP, IS-IS or BGP) inside a Layer 2 VPN routing instance (`instance-type l2vpn`). The Junos CLI disallows this configuration.

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

Configuring the Site

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

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

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

```
        remote-site-id remote-site-ID;
    }
  }
}
```

You can include this statement at the following hierarchy levels:

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

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

You must configure the following for each site:

- *site-name*—Name of the site.

- `site-identifier` *identifier*—Unsigned 16-bit number greater than zero that uniquely identifies the local Layer 2 VPN site. The site identifier corresponds to the remote site ID configured on another site within the same VPN.

- `interface` *interface-name*—The name of the interface and, optionally, a remote site ID for remote site connections. See "Configuring the Remote Site ID" on page 13.

Configuring the Remote Site ID

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

The remote site ID (configured with the `remote-site-id` statement) corresponds to the site ID (configured with the `site-identifier` statement) configured at a separate site. Figure 2 on page 13 illustrates the relationship between the site identifier and the remote site ID.

Figure 2: Relationship Between the Site Identifier and the Remote Site ID
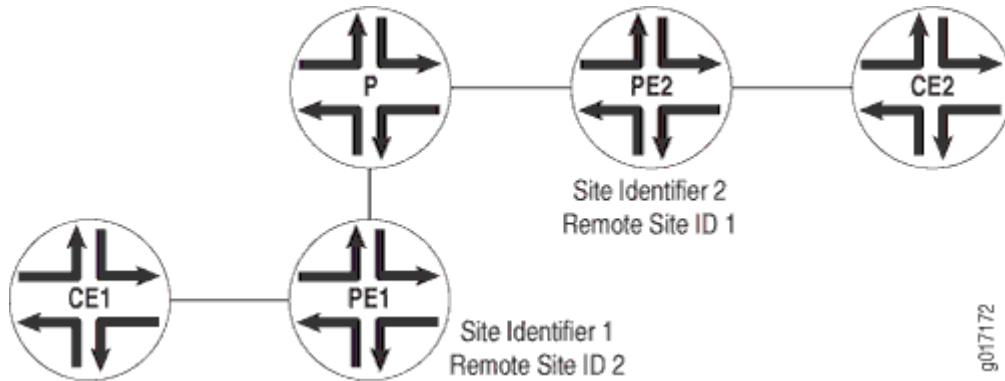
As illustrated by the figure, the configuration for Router PE1 connected to Router CE1 is as follows:

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

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

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

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

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

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

You can include this statement at the following hierarchy levels:

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

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

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

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

Configuring the Encapsulation Type

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

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

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

- `atm-cell`—ATM cell relay

- `atm-cell-port-mode`—ATM cell relay port promiscuous mode

- `atm-cell-vc-mode`—ATM virtual circuit (VC) cell relay nonpromiscuous mode

- `atm-cell-vp-mode`—ATM virtual path (VP) cell relay promiscuous mode

- `cisco-hdlc`—Cisco Systems-compatible High-Level Data Link Control (HDLC)

- `ethernet`—Ethernet

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

- `frame-relay`—Frame Relay

- `frame-relay-port-mode`—Frame Relay port mode

- `interworking`—Layer 2.5 interworking VPN

- `ppp`—Point-to-Point Protocol (PPP)

If you configure different protocols at your Layer 2 VPN sites, you need to configure a translational cross-connect (TCC) encapsulation type. For more information, see *Configuring TCC Encapsulation for Layer 2 VPNs and Layer 2 Circuits*.

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

```
encapsulation-type type;
```

For EX9200 switches, specify the encapsulation type by including the `encapsulation` statement:

```
encapsulation type;
```

You can include this statement at the following hierarchy levels:

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

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

Configuring a Site Preference and Layer 2 VPN Multihoming

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

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

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

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

You can include this statement at the following hierarchy levels:

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

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

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

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

Tracing Layer 2 VPN Traffic and Operations

To trace Layer 2 VPN protocol traffic, specify options for the `traceoptions` statement in the Layer 2 VPN configuration:

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

You can include this statement at the following hierarchy levels:

- `[edit routing-instances routing-instance-name protocols l2vpn]`

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

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

- `all`—All Layer 2 VPN tracing options.

- `connections`—Layer 2 connections (events and state changes).

- `error`—Error conditions.

- `general`—General events.

- `nlri`—Layer 2 advertisements received or sent by means of the BGP.

- `normal`—Normal events.

- `policy`—Policy processing.

- `route`—Routing information.

- `state`—State transitions.

- `task`—Routing protocol task processing.

- `timer`—Routing protocol timer processing.

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

Disabling Normal TTL Decrementing for VPNs

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

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

For more information about the `no-propagate-ttl` and `no-decrement-ttl` statements, see the MPLS Applications User Guide..

Example: Configure MPLS-Based Layer 2 VPNs

IN THIS SECTION

This example shows how to configure and validate an MPLS-based Layer 2 VPN on routers or switches running Junos OS.

> (i)   **NOTE**: Our content testing team has validated and updated this example.

You can deploy an MPLS-based Layer 2 virtual private network using routers and switches running Junos OS to interconnect customer sites with Layer 2 connectivity. Layer 2 VPNs give customers complete control over their choice of transport and routing protocols.

MPLS-based VPNs require baseline MPLS functionality in the provider network. Once basic MPLS is operational, you are able to configure VPNs that use Label-switched paths (LSPs) for transport over the provider's core.

The addition of VPN services does not affect the basic MPLS switching operations in the provider network. In fact, the provider (P) devices require only a baseline MPLS configuration because they are not VPN aware. VPN state is maintained only on the PE devices. This is a key reason why MPLS-based VPNs are so scalable.

Requirements

This example uses the following hardware and software components:

- Junos OS Release 15.1 or later

    - Revalidated on Junos OS Release 20.1R1

- Two Provider edge (PE) devices

- One provider (P) device

- Two customer edge (CE) devices

The example focuses on how to add Layer 2 VPN to a pre-existing MPLS baseline. A basic MPLS configuration is provided in case your network does not already have MPLS deployed.

To support MPLS-based VPNs the underlying MPLS baseline must provide the following functionality:

- Core-facing and loopback interfaces operational with MPLS family support

- An interior gateway protocol such as OSPF or IS-IS to provide reachability between the loopback addresses of the provider (P and PE) devices

- An MPLS signalling protocol such as LDP or RSVP to signal LSPs

- LSPs established between PE device loopback addresses

LSPs are needed between each pair of PE devices that participate in a given VPN. Its a good idea to build LSPs between all PE devices to accommodate future VPN growth. You configure LSPs at the `[edit protocols mpls]` hierarchy level. Unlike an MPLS configuration for circuit cross-connect (CCC) , you do not need to manually associate the LSP with the PE device's customer-facing (edge) interface. Instead, Layer 2 VPNs use BGP signalling to convey Layer 2 site reachability. This BGP signaling automates the mapping of remote Layer 2 VPN sites to LSP forwarding next hops. This means that with a Layer 2 VPN explicit mapping of an LSP to a PE device's edge-facing interface is not required.

For details on CCC, refer to Configuring an MPLS-Based VLAN CCC Using a Layer 2 Circuit.

Overview and Topology

A Layer 2 VPN provides complete separation between the provider and customer networks. The benefits of a Layer 2 VPN include support for nonstandard transport protocols and the isolation of link addressing and routing protocol operation between the customer and provider networks.

Definition of a VPN involves changes to the local and remote PE devices only. No additional configuration is needed on the provider devices (aside from baseline MPLS support), because these devices only provide basic MPLS switching functions. The CE devices do not use MPLS. They require only a basic interface, and if desired, protocol configuration, to operate over the Layer 2 VPN. For a Layer 2 VPN you configure the CE devices as if they were attached to a shared link.

Once an MPLS baseline is in place, you must configure the following functionality on the PE devices to establish an MPLS-based Layer 2 VPN:

- A BGP group with `family l2vpn signaling`

- A routing instance with instance type `l2vpn`

- The customer-facing interfaces on the PE devices must be configured as follows:

  - Specify `ethernet-ccc` or `vlan-ccc` physical layer encapsulation depending on whether VLAN tagging is in use.

  - Configure a matching encapsulation type in the routing instance configuration.

  - Configure the logical interface (unit) used for the Layer 2 VPN with `family ccc`.

Figure 3 on page 20 provides the topology for this MPLS-based Layer 2 VPN example. The figure details the interface names, IP addressing, and protocols used in the provider network. It also highlights the end-to-end nature of the CE device addressing and protocol stack operation. Unlike a Layer 3 VPN, CE device operation is opaque to the provider network in a Layer 2 VPN. There is no peering relationship between the CE devices and the provider network. As a result you expect the CE devices to form an OSPF adjacency *across*, *not to*, the provider network.

Figure 3: An MPLS-Based Layer 2 VPN
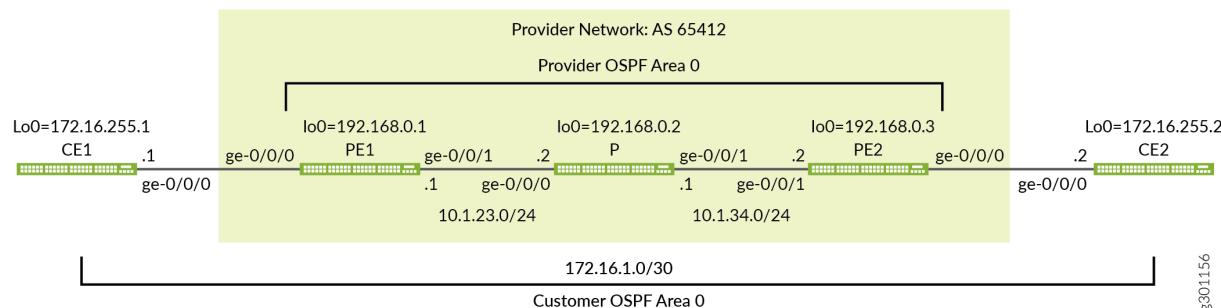
Quick Configurations

IN THIS SECTION

Use the configurations in this section to quickly get your MPLS-based Layer 2 VPN up and running. The configurations include a functional MPLS baseline to support your Layer 2 VPN. This example focuses on the VPN aspects of the configuration. Refer to the following links for additional information on the baseline MPLS functionality used in this example:

- Configuring MPLS on Provider Edge EX8200 and EX4500 Switches Using Circuit Cross-Connect

- Configuring MPLS on EX8200 and EX4500 Provider Switches

CLI Quick Configuration

> *i* **NOTE**: The device configurations omit the management interface, static routes, system logging, system services, and user login information. These parts of the configuration vary by location and are not directly related to MPLS or VPN functionality.

Edit the following commands as needed for the specifics of your environment and paste them into the local CE (CE1) device terminal window:

The complete configuration for the CE1 device.

```
set system host-name ce1
set interfaces ge-0/0/0 description "Link from CE1 to PE1"
set interfaces ge-0/0/0 unit 0 family inet address 172.16.1.1/30
set interfaces lo0 unit 0 family inet address 172.16.255.1/32
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
```

Edit the following commands as needed for the specifics of your environment and paste them into the local PE (PE1) device terminal window:

The complete configuration for PE1 device.

```
set system host-name pe1
set interfaces ge-0/0/0 description "Link from PE1 to CE1"
set interfaces ge-0/0/0 encapsulation ethernet-ccc
set interfaces ge-0/0/0 unit 0 family ccc
set interfaces ge-0/0/1 description "Link from PE1 to P-router"
set interfaces ge-0/0/1 mtu 4000
set interfaces ge-0/0/1 unit 0 family inet address 10.1.23.1/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.0.1/32
set routing-instances l2vpn1 protocols l2vpn interface ge-0/0/0.0 description "EDGE LINK BETWEEN
PE1 AND CE1"
set routing-instances l2vpn1 protocols l2vpn site CE-1 interface ge-0/0/0.0 remote-site-id 2
set routing-instances l2vpn1 protocols l2vpn site CE-1 site-identifier 1
set routing-instances l2vpn1 protocols l2vpn encapsulation-type ethernet
set routing-instances l2vpn1 instance-type l2vpn
set routing-instances l2vpn1 interface ge-0/0/0.0
set routing-instances l2vpn1 route-distinguisher 192.168.0.1:12
set routing-instances l2vpn1 vrf-target target:65412:12
set routing-options autonomous-system 65412
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 192.168.0.1
set protocols bgp group ibgp family l2vpn signaling
set protocols bgp group ibgp neighbor 192.168.0.3
set protocols mpls label-switched-path lsp_to_pe2 to 192.168.0.3
set protocols mpls interface ge-0/0/1.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols rsvp interface lo0.0
set protocols rsvp interface ge-0/0/1.0
```

The complete configuration for the P device.

```
set system host-name p
set interfaces ge-0/0/0 description "Link from P-router to PE1"
set interfaces ge-0/0/0 mtu 4000
set interfaces ge-0/0/0 unit 0 family inet address 10.1.23.2/24
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 description "Link from P-router to PE2"
```

```
set interfaces ge-0/0/1 mtu 4000
set interfaces ge-0/0/1 unit 0 family inet address 10.1.34.1/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.0.2/32
set protocols mpls interface ge-0/0/0.0
set protocols mpls interface ge-0/0/1.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols rsvp interface lo0.0
set protocols rsvp interface ge-0/0/0.0
set protocols rsvp interface ge-0/0/1.0
```

The complete configuration for the PE2 device.

```
set system host-name pe2
set interfaces ge-0/0/0 description "Link from PE2 to CE2"
set interfaces ge-0/0/0 encapsulation ethernet-ccc
set interfaces ge-0/0/0 unit 0 family ccc
set interfaces ge-0/0/1 description "Link from PE2 to P-router"
set interfaces ge-0/0/1 mtu 4000
set interfaces ge-0/0/1 unit 0 family inet address 10.1.34.2/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.0.3/32
set routing-instances l2vpn1 protocols l2vpn interface ge-0/0/0.0 description "EDGE LINK BETWEEN
PE2 AND CE2"
set routing-instances l2vpn1 protocols l2vpn site CE-2 interface ge-0/0/0.0 remote-site-id 1
set routing-instances l2vpn1 protocols l2vpn site CE-2 site-identifier 2
set routing-instances l2vpn1 protocols l2vpn encapsulation-type ethernet
set routing-instances l2vpn1 instance-type l2vpn
set routing-instances l2vpn1 interface ge-0/0/0.0
set routing-instances l2vpn1 route-distinguisher 192.168.0.3:12
set routing-instances l2vpn1 vrf-target target:65412:12
set routing-options autonomous-system 65412
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 192.168.0.3
set protocols bgp group ibgp family l2vpn signaling
set protocols bgp group ibgp neighbor 192.168.0.1
set protocols mpls label-switched-path lsp_to_pe1 to 192.168.0.1
set protocols mpls interface ge-0/0/1.0
set protocols ospf traffic-engineering
```

```
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols rsvp interface lo0.0
set protocols rsvp interface ge-0/0/1.0
```

The complete configuration for the CE2 device.

```
set system host-name ce2
set interfaces ge-0/0/0 description "Link from CE2 to PE2"
set interfaces ge-0/0/0 unit 0 family inet address 172.16.1.2/30
set interfaces lo0 unit 0 family inet address 172.16.255.2/32
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
```

Be sure to commit the configuration changes on all devices when satisfied with your work. Congratulations on your new MPLS-based Layer 2 VPN! Refer to the "Verification" on page 38 section for the steps needed to confirm your VPN is working as expected.

Configure the Local PE (PE1) Device for a MPLS-Based Layer 2 VPN

IN THIS SECTION

This section covers the steps needed to configure the PE1 device for this example. Refer to the *Example: Configure MPLS-Based Layer 2 VPNs* section for the CE device and P device configurations used in this example.

Configure the MPLS Baseline (if Needed)

Before you configure the Layer 2 VPN make sure the PE device has a working MPLS baseline. If you already having a an MPLS baseline you can skip to the step-by-step procedure to add the Layer 2 VPN to the local PE device.

- Configure the hostname.

```
[edit]
user@pe1# set system host-name pe1
```

- Configure the interfaces.

```
[edit]
user@pe1# set interfaces ge-0/0/1 description "Link from PE1 to P-router"
[edit]
user@pe1# set interfaces ge-0/0/1 mtu 4000
[edit]
user@pe1# set interfaces ge-0/0/1 unit 0 family inet address 10.1.23.1/24
[edit]
user@pe1# set interfaces ge-0/0/1 unit 0 family mpls
[edit]
user@pe1# set interfaces lo0 unit 0 family inet address 192.168.0.1/32
```

⚠️ **CAUTION**: Layer 2 VPNs don't support fragmentation in the provider network. It is critical that the provider network supports the largest frame that the CE devices can generate *after* the MPLS and virtual routing and forwarding (VRF) labels are added by the PE devices. This example leaves the CE devices at the default 1500-byte maximum transmission unit (MTU) while configuring the provider core to support a 4000 byte MTU. This configuration avoids discards by ensuring the CE devices cannot exceed the MTU in the provider's network.

- Configure the protocols.

ⓘ **NOTE**: Traffic engineering is supported for RSVP-signaled LSPs but is not required for basic MPLS switching or VPN deployment. The provided MPLS baseline uses RSVP to signal LSPs, and enables traffic engineering for OSPF. However, no path constraints are configured so you expect the LSPs to be routed over the interior gateway protocol's shortest path.

```
[edit
]user@pe1# set protocols ospf area 0.0.0.0 interface lo0.0
[edit]
user@pe1# set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
[edit]
user@pe1# set protocols ospf traffic-engineering
[edit]
user@pe1# set protocols mpls interface ge-0/0/1.0
[edit]
```

```
user@pe1# set protocols rsvp interface lo0.0
[edit]
user@pe1# set protocols rsvp interface ge-0/0/1.0
```

- Define the LSP to the remote PE device's loopback address.

```
[edit]
user@pe1#  set protocols mpls label-switched-path lsp_to_pe2 to 192.168.0.3
```

Procedure

Step-by-Step Procedure

Follow these steps to configure the PE1 device for a Layer 2 VPN.

1. Configure the edge-facing interface. Specify a physical encapsulation type of `ethernet-ccc` with `family ccc` on unit 0. This is the only valid unit number for an untagged Ethernet interface. If you are using VLAN tagging specify `vlan-ccc` encapsulation and add the CCC family to the desired unit(s).

   > 💡 **TIP**: You can configure both an MPLS-based Layer 2 VPN and an MPLS-based Layer 3 VPN on the same PE device. However, you cannot configure the same customer edge-facing interface to support both a Layer 2 VPN and a Layer 3 VPN.

   ```
   [edit]user@pe1# set interfaces ge-0/0/0 encapsulation ethernet-ccc
   [edit]
   user@pe1# set interfaces ge-0/0/0 unit 0 family ccc
   [edit]
   user@pe1# set interfaces ge-0/0/0 description "Link from PE1 to CE1"
   ```

   > ℹ️ **NOTE**: A Layer 2 VPN requires that the PE device's edge-facing interfaces be configured with CCC encapsulation at the physical device level with the CCC family configured at the unit level. The provider devices are configured in the same way whether you are deploying CCC, an MPLS-based Layer 2 VPN, or an MPLS-based Layer 3 VPN. This is because they have no edge-facing interfaces or VPN awareness.

2. Configure a BGP group for the peering between the local and remote PE devices. Use the PE device's loopback address as the local address and enable `family l2vpn signaling`.

```
[edit protocols bgp]
user@pe1# set group ibgp local-address 192.168.0.1 family l2vpn signaling
```

3. Configure the BGP group type as internal.

```
[edit protocols bgp]
user@pe1# set group ibgp type internal
```

4. Configure the remote PE device's loopback address as a BGP neighbor.

```
[edit protocols bgp]
user@pe1# set group ibgp neighbor 192.168.0.3
```

5. Configure the BGP autonomous system number.

```
[edit routing-options]
user@pe1# set autonomous-system 65412
```

6. Configure the routing instance. Start by specifying the instance name *l2vpn1*, with an `instance-type` of `l2vpn`.

```
[edit routing-instances]
user@pe1# set l2vpn1 instance-type l2vpn
```

7. Configure the PE device's customer-facing interface to belong to the routing instance.

```
[edit routing-instances]
user@pe1# set l2vpn1 interface ge-0/0/0
```

8. Configure the routing instance's route distinguisher. This setting is used to distinguish the routes sent from a particular VRF on a particular PE device. It should be unique for each routing instance on each PE device.

```
[edit routing-instances]
user@pe1# set l2vpn1 route-distinguisher 192.168.0.1:12
```

9. Configure the instance's virtual routing and forwarding (VRF) table route target. The `vrf-target` statement adds the specified community tag to all advertised routes while automatically matching the same value for route import. Configuring matching route targets on the PE devices that share a given VPN is required for proper route exchange.

```
[edit routing-instances]
user@pe1# set l2vpn1 vrf-target target:65412:12
```

> (i) **NOTE**: You can create more complex policies by explicitly configuring VRF import and export policies using the import and export options. See *vrf-import* and *vrf-export* for details.

10. Configure the `l2vpn` protocol in the instance and specify the encapsulation that is used on the edge-facing link. If the edge interface is VLAN tagged, be sure to specify `ethernet-vlan`.

```
[edit routing-instances]
user@pe1# set l2vpn1 protocols l2vpn encapsulation-type ethernet
```

11. Add the edge-facing interface under the instance's `l2vpn` stanza along with a description.

```
[edit routing-instances]
user@pe1# set l2vpn1 protocols l2vpn interface ge-0/0/0.0 description "L2vpn Link Between
PE1 and CE1"
```

12. Configure the Layer 2 VPN site information and associate the edge-facing interface with the local customer site.

```
[edit routing-instances]
user@pe1# set l2vpn1 protocols l2vpn site CE-1 site-identifier 1 interface ge-0/0/0.0
remote-site-id 2
```

> (i) **NOTE**: In this example, the site ID for the PE1 device is *1* and the site ID for the PE2 device is *2*. For the local PE device (PE1), the remote site is correctly configured with a `remote-site-id` value of 2.

13. Commit your changes at the PE1 device and return to CLI operational mode.

```
[edit]
user@pe1# commit and-quit
```

Results

Display the results of the configuration on the PE1 device. The output reflects only the functional configuration added in this example.

```
user@pe1> show configuration
interfaces {
    ge-0/0/0 {
        description "Link from PE1 to CE1";
        encapsulation ethernet-ccc;
        unit 0 {
            family ccc;
        }
    }
    ge-0/0/1 {
        description "Link from PE1 to P-router";
        mtu 4000;
        unit 0 {
            family inet {
                address 10.1.23.1/24;
            }
            family mpls;
```

```
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 192.168.0.1/32;
            }
        }
    }
}
routing-instances {
    l2vpn1 {
        protocols {
            l2vpn {
                interface ge-0/0/0.0 {
                    description "L2vpn Link Between PE1 and CE1" ;
                }
                site CE-1 {
                    interface ge-0/0/0.0 {
                        remote-site-id 2;
                    }
                    site-identifier 1;
                }
                encapsulation-type ethernet;
            }
        }
        instance-type l2vpn;
        interface ge-0/0/0.0;
        route-distinguisher 192.168.0.1:12;
        vrf-target target:65412:12;
    }
}
routing-options {
    autonomous-system 65412;
}
protocols {
    bgp {
        group ibgp {
            type internal;
            local-address 192.168.0.1;
            family l2vpn {
                signaling;
            }
```

```
            neighbor 192.168.0.3;
        }
    }
    mpls {
        label-switched-path lsp_to_pe2 {
            to 192.168.0.3;
        }
        interface ge-0/0/1.0;
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface lo0.0;
            interface ge-0/0/1.0;
        }
    }
    rsvp {
        interface lo0.0;
        interface ge-0/0/1.0;
    }
 }
```

Configure the Remote PE (PE2) Device for a MPLS-Based Layer 2 VPN

IN THIS SECTION

- Procedure | 33

This section covers the steps needed to configure the PE2 device for this example. Refer to the
Example: Configure MPLS-Based Layer 2 VPNs section for the CE device and P device configurations
used in this example.

Configure the MPLS Baseline (if Needed)

Before you configure the Layer 2 VPN make sure the PE device has a working MPLS baseline. If you
already having an MPLS baseline you can skip to the step-by-step procedure to add the Layer 2 VPN to
the local PE device.

- Configure the hostname.

```
[edit]
user@pe2# set system host-name pe2
```

- Configure the interfaces.

```
[edit]
user@pe2# set interfaces ge-0/0/1 description "Link from PE2 to P-router"
[edit]
user@pe2# set interfaces ge-0/0/1 mtu 4000
[edit]
user@pe2# set interfaces ge-0/0/1 unit 0 family inet address 10.1.34.2/24
[edit]
user@pe2# set interfaces ge-0/0/1 unit 0 family mpls
[edit]
user@pe2# set interfaces lo0 unit 0 family inet address 192.168.0.3/32
```

⚠️ **CAUTION**: Layer 2 VPNs don't support fragmentation in the provider network. It is critical that the provider network supports the largest frame that the CE devices can generate *after* the MPLS and virtual routing and forwarding (VRF) labels are added by the PE devices. This example leaves the CE devices at the default 1500-byte maximum transmission unit (MTU) while configuring the provider core to support a 4000 byte MTU. This configuration avoids discards by ensuring the CE devices cannot exceed the MTU in the provider's network.

- Configure the protocols.

ⓘ **NOTE**: Traffic engineering is supported for RSVP-signaled LSPs but is not required for basic MPLS switching or VPN deployment. The provided MPLS baseline uses RSVP to signal LSPs, and enables traffic engineering for OSPF. However, no path constraints are configured so you expect the LSPs to be routed over the interior gateway protocol's shortest path.

```
[edit]
user@pe2# set protocols ospf area 0.0.0.0 interface lo0.0
```

```
[edit]
user@pe2# set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
[edit]
user@pe2# set protocols ospf traffic-engineering
[edit]
user@pe2# set protocols mpls interface ge-0/0/1.0
[edit]
user@pe2# set protocols rsvp interface lo0.0
[edit]
user@pe2# set protocols rsvp interface ge-0/0/1.0
```

- Define the LSP to the remote PE device's loopback address.

```
[edit]
user@pe2#  set protocols mpls label-switched-path lsp_to_pe1 to 192.168.0.1
```

Procedure

Step-by-Step Procedure

Follow these steps to configure the PE2 device for a Layer 2 VPN.

1. Configure the edge-facing interface encapsulation and family. Recall this is an untagged interface, therefore only unit 0 is valid for the ccc family.

```
[edit]user@pe2# set interfaces ge-0/0/0 encapsulation ethernet-ccc
[edit]
user@pe2# set interfaces ge-0/0/0 unit 0 family ccc
[edit]
user@pe1# set interfaces ge-0/0/0 description "Link from PE2 to CE2"
```

2. Configure a BGP group. Specify the PE device's loopback address as the local address and enable family l2vpn signaling.

```
[edit protocols bgp]
user@pe2# set group ibgp local-address 192.168.0.3 family l2vpn signaling
```

3. Configure the BGP group type as internal.

```
[edit protocols bgp]
user@pe2# set group ibgp type internal
```

4. Configure the PE1 device as a BGP neighbor. Be sure to specify PE1's loopback address as the BGP neighbor.

```
[edit protocols bgp]
user@pe2# set group ibgp neighbor 192.168.0.1
```

5. Configure the BGP autonomous system number.

```
[edit routing-options]
user@pe2# set autonomous-system 65412
```

6. Configure the routing instance. Start by specifying the instance name *l2vpn1* with an `instance-type` of `l2vpn`.

```
[edit routing-instances]
user@pe2# set l2vpn1 instance-type l2vpn
```

7. Configure the PE device's customer edge-facing interface to belong to the routing instance.

```
[edit routing-instances]
user@pe2# set l2vpn1 interface ge-0/0/0
```

8. Configure the instance's route distinguisher.

```
[edit routing-instances]
user@pe2# set l2vpn1 route-distinguisher 192.168.0.3:12
```

9. Configure the instance's VPN virtual routing and forwarding (VRF) table route target. The assigned target must match the one configured at the PE1 device.

```
[edit routing-instances]
user@pe2# set l2vpn1 vrf-target target:65412:12
```

10. Configure the instance for the l2vpn protocol and specify the encapsulation used on the edge-facing link.

```
[edit routing-instances]
user@pe2# set l2vpn1 protocols l2vpn encapsulation-type ethernet
```

11. Add the PE device's edge-facing interface under the instance's l2vpn hierarchy along with a description .

```
[edit routing-instances]
user@pe2# set l2vpn1 protocols l2vpn interface ge-0/0/0.0 description "L2vpn Link Between
PE2 and CE2"
```

12. Configure the instance's Layer 2 VPN site information and list the PE device's edge-facing interface under the local site. The local site ID configured on the PE2 device must match the remote site ID you configured on the PE1 device, and vice versa.

```
[edit routing-instances]
user@pe1# set l2vpn1 protocols l2vpn site CE-2 site-identifier 2 interface ge-0/0/0.0
remote-site-id 1
```

> ⓘ **NOTE**: In this example, the site ID for the PE2 device is *2* and the site ID for the PE1 device is *1*. For the PE2 device the remote site is correctly configured with a remote-site-id value of *1*.

13. Commit your changes at the PE2 device and return to CLI operational mode.

```
[edit]
user@pe1# commit and-quit
```

Results

Display the results of the configuration on the PE2 device.

```
user@pe2# show
```

```
interfaces {
    ge-0/0/0 {
        description "Link from PE2 to CE2";
        encapsulation ethernet-ccc;
        unit 0 {
            family ccc;
        }
    }
    ge-0/0/1 {
        description "Link from PE2 to P-router";
        mtu 4000;
        unit 0 {
            family inet {
                address 10.1.34.2/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 192.168.0.3/32;
            }
        }
    }
}
routing-instances {
    l2vpn1 {
        protocols {
            l2vpn {
                interface ge-0/0/0.0 {
                    description "L2vpn Link Between PE2 and CE2" ;
                }
                site CE-2 {
                    interface ge-0/0/0.0 {
```

```
                    remote-site-id 1;
                }
                site-identifier 2;
            }
            encapsulation-type ethernet;
        }
    }
    instance-type l2vpn;
    interface ge-0/0/0.0;
    route-distinguisher 192.168.0.3:12;
    vrf-target target:65412:12;
    }
}
routing-options {
    autonomous-system 65412;
}
protocols {
    bgp {
        group ibgp {
            type internal;
            local-address 192.168.0.3;
            family l2vpn {
                signaling;
            }
            neighbor 192.168.0.1;
        }
    }
    mpls {
        label-switched-path lsp_to_pe1 {
            to 192.168.0.1;
        }
        interface ge-0/0/1.0;
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface lo0.0;
            interface ge-0/0/1.0;
        }
    }
    rsvp {
        interface lo0.0;
        interface ge-0/0/1.0;
```

```
    }
}
```

Verification

Perform these tasks to verify that the MPLS-based Layer 2 VPN works properly:

Verify Provider OSPF Adjacencies and Route Exchange

Purpose

Confirm the OSPF protocol is working properly in the provider network by verifying adjacency status and OSPF learned routes to the loopback addresses of the remote provider devices. Proper IGP operation is critical for the successful establishment of MPLS LSPs.

Action

```
user@pe1> show ospf neighbor

Address          Interface         State          ID             Pri  Dead
10.1.23.2        ge-0/0/1.0        Full           192.168.0.2    128  38
```

```
user@pe1> show route protocol ospf | match 192.168
```

```
192.168.0.2/32      *[OSPF/10] 1w5d 20:48:59, metric 1
192.168.0.3/32      *[OSPF/10] 2w0d 00:08:30, metric 2
```

Meaning

The output shows that the PE1 device has established an OSPF adjacency to the P device (192.168.0.2). It also shows that the P and remote PE device loopback addresses (192.168.0.2) and (192.168.0.3) are learned via OSPF at the local PE device.

Verify MPLS and RSVP Interface Settings

Purpose

Verify that the RSVP and MPLS protocols are configured to operate on the PE device's core-facing interfaces. This step also verifies that `family mpls` is correctly configured at the unit level of the core-facing interfaces.

Action

```
user@pe1> show mpls interface

Interface       State       Administrative groups (x: extended)
ge-0/0/1.0      Up          <none>
```

```
user@pe1> show rsvp interface

RSVP interface: 2 active

                    Active  Subscr- Static      Available   Reserved   Highwater
Interface       State  resv    iption  BW          BW          BW         mark
ge-0/0/1.0         Up      1    100%  1000Mbps    1000Mbps    0bps       0bps
lo0.0              Up      0    100%  0bps        0bps        0bps       0bps
```

Meaning

The output shows that MPLS and RSVP are correctly configured on the local PE device's core-facing and loopback interfaces.

Verify RSVP Signaled LSPs

Purpose

Verify that the RSVP sessions (ingress and egress) are properly established between the PE devices.

Action

```
user@pe1> show rsvp session

To              From          State   Rt Style Labelin Labelout LSPname
192.168.0.3     192.168.0.1   Up       0  1 FF      -    299888 lsp_to_pe2
Total 1 displayed, Up 1, Down 0

Egress RSVP: 1 sessions
To              From          State   Rt Style Labelin Labelout LSPname
192.168.0.1     192.168.0.3   Up       0  1 FF      3         - lsp_to_pe1
Total 1 displayed, Up 1, Down 0


Transit RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Meaning

The output shows that both the ingress and egress RSVP sessions are correctly established between the PE devices. Successful LSP establishment indicates the MPLS baseline is operational.

Verify BGP Session Status

Purpose

Verify that the BGP session between the PE devices is correctly established with support for Layer 2 VPN network layer reachability information (NLRI).

Action

```
user@pe1> show bgp summary
Threading mode: BGP I/O
Groups: 1 Peers: 1 Down peers: 0
Table         Tot Paths  Act Paths Suppressed   History Damp State   Pending
```

```
bgp.l2vpn.0
                        1          1        0        0        0        0
Peer                   AS      InPkt   OutPkt     OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
192.168.0.3         65412          6        5        0        0         1:34 Establ
  bgp.l2vpn.0: 1/1/1/0
  l2vpn1.l2vpn.0: 1/1/1/0
```

Meaning

The output shows the BGP session to the remote PE device (192.168.0.3) has been correctly established (Establ), and through the Up/Dwn field, how long the session has been in the current state (1:34). It also shows the number of BGP packets sent to (5) and received from (6) the remote PE device. The flaps field confirms that no state transitions have occurred (0), indicating the session is stable. Also note that Layer 2 VPN NLRI is correctly exchanged between the PE devices. This output confirms the BGP peering between the PE devices is ready to support a Layer 2 VPN.

Verify Layer 2 VPN Routes in the Routing Table

Purpose

Verify that the routing table on the PE1 device is populated with the Layer 2 VPN routes used to forward traffic between the CE devices.

Action

```
user@pe1> show route table bgp.l2vpn.0

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

192.168.0.3:12:2:1/96
                  *[BGP/170] 00:51:36, localpref 100, from 192.168.0.3
                     AS path: I, validation-state: unverified
                   >  to 10.1.23.2 via ge-0/0/1.0, label-switched-path lsp_to_pe2
```

```
user@pe1> show route table l2vpn1.l2vpn.0

l2vpn1.l2vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
```

```
+ = Active Route, - = Last Active, * = Both


192.168.0.1:12:1:1/96
                  *[L2VPN/170/-101] 01:48:30, metric2 1
                      Indirect
192.168.0.3:12:2:1/96
                  *[BGP/170] 00:51:57, localpref 100, from 192.168.0.3
                    AS path: I, validation-state: unverified
                  >  to 10.1.23.2 via ge-0/0/1.0, label-switched-path lsp_to_pe2
```

Meaning

The command `show route table bgp.l2vpn.0` displays all Layer 2 VPN routes that have been received on the PE device. The command `show route table l2vpn1.l2vpn.0` shows the Layer 2 VPN routes that have been imported into the `l2vpn1` routing instance as a result of a matching route target. The `l2vpn1.l2vpn.0` table contains both the local PE device's Layer 2 VPN route as well as a remote route learned via the BGP peering to the remote PE device. Both tables show the remote Layer 2 VPN route is correctly associated with the `lsp_to_pe2` LSP as a forwarding next hop. The outputs confirm the local PE device has learned about the remote customer site from the PE2 device. It also shows that it can forward Layer 2 VPN traffic to the PE2 device using MPLS transport over the provider network.

Verify Layer 2 VPN Connection Status

Purpose

Verify the status of the Layer 2 VPN connection.

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           HS -- Hot-standby Connection


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


Instance: l2vpn1
Edge protection: Not-Primary
  Local site: CE-1 (1)
    connection-site         Type  St     Time last up         # Up trans
    2                       rmt   Up     Jul 28 10:47:18 2020          1
      Remote PE: 192.168.0.3, Negotiated control-word: Yes (Null)
      Incoming label: 800009, Outgoing label: 800006
      Local interface: ge-0/0/0.0, Status: Up, Encapsulation: ETHERNET
      Flow Label Transmit: No, Flow Label Receive: No
```

Meaning

The `St` field in the output shows that the Layer 2 VPN connection to `Remote PE 192.168.0.3` at `connection-site 2` is `Up`. The output also confirms the PE device's edge-facing interface name `ge-0/0/0.0` and operational status as `up`. You also verify that Ethernet encapsulation is configured on the PE device's customer-facing interface. This is the correct encapsulation for the untagged Ethernet interfaces used in this example. The verification steps performed thus far indicate that the Layer 2 VPN's control plane is operational. You verify the data plane of the Layer 2 VPN in the following steps.

Ping the Remote PE Device Using the Layer 2 VPN Connection

Purpose

Verify Layer 2 VPN connectivity between the local and remote PE devices. Two forms of the `ping mpls l2vpn` command are shown. Both test Layer 2 VPN routing and MPLS forwarding between the PE

devices. The first command assumes a single remote site while the second specifies the local and remote site identifiers, which is useful when testing a multi-site Layer 2 VPN. This is because the remote site ID can be used to target the desired remote PE device.

> **(i)** **NOTE**: The `ping mpls l2vpn` command validates Layer 2 VPN route exchange and MPLS forwarding between the PE devices. This is done by generating traffic from the local PE's Layer 2 VPN routing instance to the remote PE device's 127.0.0.1 loopback address. This command does not validate the operation of the CE device interfaces or their configuration. This is because CE device operation is opaque to the provider network in a Layer 2 VPN.

Action

```
user@pe1> ping mpls l2vpn interface ge-0/0/0.0 reply-mode ip-udp

!!!!!
--- lsping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
```

```
user@pe1> ping mpls l2vpn instance l2vpn1 remote-site-id 2 local-site-id 1 detail

Request for seq 1, to interface 334, labels <800002, 299840>, packet size 88
Reply for seq 1, return code: Egress-ok, time: 593.784 ms
        Local transmit time: 2020-07-13 16:15:55 UTC 241.357 ms
        Remote receive time: 2020-07-13 16:15:55 UTC 835.141 ms
Request for seq 2, to interface 334, labels <800002, 299840>, packet size 88
Reply for seq 2, return code: Egress-ok, time: 591.700 ms
        Local transmit time: 2020-07-13 16:15:56 UTC 241.405 ms
        Remote receive time: 2020-07-13 16:15:56 UTC 833.105 ms
Request for seq 3, to interface 334, labels <800002, 299840>, packet size 88
Reply for seq 3, return code: Egress-ok, time: 626.084 ms
        Local transmit time: 2020-07-13 16:15:57 UTC 241.407 ms
        Remote receive time: 2020-07-13 16:15:57 UTC 867.491 ms
Request for seq 4, to interface 334, labels <800002, 299840>, packet size 88
Reply for seq 4, return code: Egress-ok, time: 593.061 ms
        Local transmit time: 2020-07-13 16:15:58 UTC 241.613 ms
        Remote receive time: 2020-07-13 16:15:58 UTC 834.674 ms
Request for seq 5, to interface 334, labels <800002, 299840>, packet size 88
Reply for seq 5, return code: Egress-ok, time: 594.192 ms
```

```
        Local transmit time: 2020-07-13 16:15:59 UTC 241.357 ms
        Remote receive time: 2020-07-13 16:15:59 UTC 835.549 ms


--- lsping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
```

Meaning

The output confirms that the Layer 2 VPN forwarding plane is operating correctly between the PE devices.

Verify End-to-End Operation of the CE Devices Over the Layer 2 VPN

Purpose

Verify Layer 2 VPN connectivity between the CE devices. This step confirms the CE devices have operational interfaces and are properly configured for Layer 2 connectivity. This is done by verifying the CE devices have established an OSPF adjacency and are able to pass traffic end-to-end between their loopback addresses.

Action

```
user@ce1> show ospf neighbor

Address          Interface              State          ID              Pri  Dead
172.16.1.2       ge-0/0/0.0             Full           172.16.255.2    128   32
```

```
user@ce1> show ospf route | match 172

172.16.255.2/32    *[OSPF/10] 01:34:50, metric 1
                   >  to 172.16.1.2 via ge-0/0/0.0
```

```
user@ce1> ping 172.16.255.2 size 1472 do-not-fragment count 2

PING 172.16.255.2 (172.16.255.2): 1472 data bytes
1480 bytes from 172.16.255.2: icmp_seq=0 ttl=64 time=4.404 ms
1480 bytes from 172.16.255.2: icmp_seq=1 ttl=64 time=5.807 ms
```

```
--- 172.16.255.2 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 4.404/5.106/5.807/0.702 ms
```

Meaning

The output shows that Layer 2 VPN connectivity is working correctly between the CE devices. It confirms that the local CE device has established an OSPF adjacency over the provider core to the remote CE device `172.16.1.2`, and that the local CE device has learned a route to the remote CE device's loopback address `172.16.255.2` via OSPF. The output also shows that the CE devices are able to pass 1500-byte IP packets without evoking local fragmentation. The successful pings also verify the frames did not exceed the MTU supported by the provider's network.

> **(i)** **NOTE**: The `size` argument added to the `ping` command generates 1472 bytes of echo data. An additional 8 bytes of Internet Control Message Protocol (ICMP) and 20 bytes of IP header are added to bring the total packet size to 1500-bytes. Adding the `do-not-fragment` switch ensures the CE device cannot perform fragmentation based on its local MTU. This method confirms that no fragmentation is possible, or needed, when sending standard length Ethernet frames between the CE devices.

RELATED DOCUMENTATION

Example: Configuring MPLS on EX8200 and EX4500 Switches

Example: Configure a Basic MPLS-Based Layer 3 VPN

CHAPTER 4

Configuring Layer 2 Interfaces

IN THIS CHAPTER

Configuring CCC Encapsulation for Layer 2 VPNs

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

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

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

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

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

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

- [edit interfaces *interface-name*]

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

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

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

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

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

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

Configuring TCC Encapsulation for Layer 2 VPNs and Layer 2 Circuits

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

> (i) **NOTE**: The EX9200 switches only use ethernet and ethernet-vlan encapsulation types.

For information about how to configure encapsulations for Layer 2 circuits, see *Configuring the Interface Encapsulation Type for Layer 2 Circuits*

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

> **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.

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

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

On the EX9200 switches, replace `encapsulation-type` with the `encapsulation` statement:

```
encapsulation tcc-encapsulation;
```

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

- `[edit interfaces interface-name]`

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

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

- `[edit interfaces interface-name unit logical-unit-number]`

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

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

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

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

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

You can include these statements at the following hierarchy levels:

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

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

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

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

Ethernet TCC is supported on interfaces that carry IP version 4 (IPv4) traffic only. However, Ethernet TCC encapsulation is not supported on 8-port, 12-port, and 48-port Fast Ethernet PICs.

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.

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, the advertised MTU will be zero unless an MTU value is explicitly set at the `[edit routing-instances` *`routing-instance-name`* `protocols (`*`l2vpn`* `) 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.

- 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

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

mtu

Disabling the Control Word for Layer 2 VPNs

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

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

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

You can include this statement at the following hierarchy levels:

- `[edit routing-instances routing-instance-name protocols l2vpn]`

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

For more information about configuring the control word, see *Configuring the Control Word for Layer 2 Circuits* and the *Layer 2 Circuits User Guide*.

RELATED DOCUMENTATION

Configuring the Control Word for Layer 2 Circuits

control-word

l2vpn

Configuring Path Selection for Layer 2 VPNs and VPLS

IN THIS CHAPTER

Understanding BGP Path Selection

IN THIS SECTION

For each prefix in the routing table, the routing protocol process selects a single best path. After the best path is selected, the route is installed in the routing table. The best path becomes the active route if the same prefix is not learned by a protocol with a lower (more preferred) global preference value, also known as the administrative distance. The algorithm for determining the active route is as follows:

1. Verify that the next hop can be resolved.

2. Choose the path with the lowest preference value (routing protocol process preference).

   Routes that are not eligible to be used for forwarding (for example, because they were rejected by routing policy or because a next hop is inaccessible) have a preference of −1 and are never chosen.

3. Prefer the path with higher local preference.

   For non-BGP paths, choose the path with the lowest **preference2** value.

4. If the accumulated interior gateway protocol (AIGP) attribute is enabled, add the IGP metric and prefer the path with the lower AIGP attribute.

5. Prefer the path with the shortest autonomous system (AS) path value (skipped if the `as-path-ignore` statement is configured).

   A confederation segment (sequence or set) has a path length of 0. An AS set has a path length of 1.

6. Prefer the route with the lower origin code.

   Routes learned from an IGP have a lower origin code than those learned from an exterior gateway protocol (EGP), and both have lower origin codes than incomplete routes (routes whose origin is unknown).

7. Prefer the path with the lowest multiple exit discriminator (MED) metric.

   Depending on whether nondeterministic routing table path selection behavior is configured, there are two possible cases:

   - If nondeterministic routing table path selection behavior is not configured (that is, if the `path-selection cisco-nondeterministic` statement is not included in the BGP configuration), for paths with the same neighboring AS numbers at the front of the AS path, prefer the path with the lowest MED metric. To always compare MEDs whether or not the peer ASs of the compared routes are the same, include the `path-selection always-compare-med` statement.

   - If nondeterministic routing table path selection behavior is configured (that is, the `path-selection cisco-nondeterministic` statement is included in the BGP configuration), prefer the path with the lowest MED metric.

   Confederations are not considered when determining neighboring ASs. A missing MED metric is treated as if a MED were present but zero.

   > ⓘ **NOTE**: MED comparison works for single path selection within an AS (when the route does not include an AS path), though this usage Is uncommon.

   By default, only the MEDs of routes that have the same peer autonomous systems (ASs) are compared. You can configure routing table path selection options to obtain different behaviors.

8. Prefer strictly internal paths, which include IGP routes and locally generated routes (static, direct, local, and so forth).

9. Prefer strictly external BGP (EBGP) paths over external paths learned through internal BGP (IBGP) sessions.

10. Prefer the path whose next hop is resolved through the IGP route with the lowest metric. BGP routes that are resolved through IGP are preferred over unreachable or rejected routes.

> **(i)** **NOTE**: A path is considered a BGP equal-cost path (and will be used for forwarding) if a tie-break is performed after the previous step. All paths with the same neighboring AS, learned by a multipath-enabled BGP neighbor, are considered.
>
> BGP multipath does not apply to paths that share the same MED-plus-IGP cost yet differ in IGP cost. Multipath path selection is based on the IGP cost metric, even if two paths have the same MED-plus-IGP cost.

11. If both paths are external, prefer the oldest path, in other words, the path that was learned first. This is done to minimize route-flapping. This rule is not used if any one of the following conditions is true:

    - **path-selection external-router-id** is configured.

    - Both peers have the same router ID.

    - Either peer is a confederation peer.

    - Neither path is the current active path.

12. Prefer the path from the peer with the lowest router ID. For any path with an originator ID attribute, substitute the originator ID for the router ID during router ID comparison.

13. Prefer the path with the shortest cluster list length. The length is 0 for no list.

14. Prefer the path from the peer with the lowest peer IP address.

15. Prefer a primary route over a secondary route. A primary route is one that belongs to the routing table. A secondary route is one that is added to the routing table through an export policy.

Routing Table Path Selection

The shortest AS path step of the algorithm, by default, evaluates the length of the AS path and determines the active path. You can configure an option that enables Junos OS to skip this step of the algorithm by including the **as-path-ignore** option.

> **(i)** **NOTE**: Starting with Junos OS Release 14.1R8, 14.2R7, 15.1R4, 15.1F6, and 16.1R1, the **as-path-ignore** option is supported for routing instances.

The routing process path selection takes place before BGP hands off the path to the routing table to makes its decision. To configure routing table path selection behavior, include the `path-selection` statement:

```
path-selection {
    (always-compare-med | cisco-non-deterministic | external-router-id);
    as-path-ignore;
    l2vpn-use-bgp-rules;
    med-plus-igp {
        igp-multiplier number;
        med-multiplier number;
    }
}
```

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

Routing table path selection can be configured in one of the following ways:

- Emulate the Cisco IOS default behavior (**cisco-non-deterministic**). This mode evaluates routes in the order that they are received and does not group them according to their neighboring AS. With `cisco-non-deterministic` mode, the active path is always first. All inactive, but eligible, paths follow the active path and are maintained in the order in which they were received, with the most recent path first. Ineligible paths remain at the end of the list.

  As an example, suppose you have three path advertisements for the 192.168.1.0 /24 route:

  - Path 1—learned through EBGP; AS Path of 65010; MED of 200

  - Path 2—learned through IBGP; AS Path of 65020; MED of 150; IGP cost of 5

  - Path 3—learned through IBGP; AS Path of 65010; MED of 100; IGP cost of 10

  These advertisements are received in quick succession, within a second, in the order listed. Path 3 is received most recently, so the routing device compares it against path 2, the next most recent advertisement. The cost to the IBGP peer is better for path 2, so the routing device eliminates path 3 from contention. When comparing paths 1 and 2, the routing device prefers path 1 because it is received from an EBGP peer. This allows the routing device to install path 1 as the active path for the route.

  > **NOTE**: We do not recommend using this configuration option in your network. It is provided solely for interoperability to allow all routing devices in the network to make consistent route selections.

- Always comparing MEDs whether or not the peer ASs of the compared routes are the same (**always-compare-med**).

- Override the rule that If both paths are external, the currently active path is preferred (**external-router-id**). Continue with the next step (Step) in the path-selection process.

- Adding the IGP cost to the next-hop destination to the MED value before comparing MED values for path selection (`med-plus-igp`).

  BGP multipath does not apply to paths that share the same MED-plus-IGP cost, yet differ in IGP cost. Multipath path selection is based on the IGP cost metric, even if two paths have the same MED-plus-IGP cost.

BGP Table path selection

The following parameters are followed for BGP's path selection:

1. Prefer the highest local-preference value.

2. Prefer the shortest AS-path length.

3. Prefer the lowest origin value.

4. Prefer the lowest MED value.

5. Prefer routes learned from an EBGP peer over an IBGP peer.

6. Prefer best exit from AS.

7. For EBGP-received routes, prefer the current active route.

8. Prefer routes from the peer with the lowest Router ID.

9. Prefer paths with the shortest cluster length.

10. Prefer routes from the peer with the lowest peer IP address. Steps 2, 6 and 12 are the RPD criteria.

Effects of Advertising Multiple Paths to a Destination

BGP advertises only the active path, unless you configure BGP to advertise multiple paths to a destination.

Suppose a routing device has in its routing table four paths to a destination and is configured to advertise up to three paths (**add-path send path-count 3**). The three paths are chosen based on path selection criteria. That is, the three best paths are chosen in path-selection order. The best path is the active path. This path is removed from consideration and a new best path is chosen. This process is repeated until the specified number of paths is reached.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.1R8	Starting with Junos OS Release 14.1R8, 14.2R7, 15.1R4, 15.1F6, and 16.1R1, the **as-path-ignore** option is supported for routing instances.

RELATED DOCUMENTATION

Example: Ignoring the AS Path Attribute When Selecting the Best Path

Examples: Configuring BGP MED

Example: Advertising Multiple BGP Paths to a Destination

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.

> (i) **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 the Layer 2 VPN or VPLS 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 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.

RELATED DOCUMENTATION

Understanding BGP Path Selection

VPLS Path Selection Process for PE Routers

path-selection

route-distinguisher

CHAPTER 6

Creating Backup Connections with Redundant Pseudowires

Redundant Pseudowires for Layer 2 Circuits and VPLS

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

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

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

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

Types of Redundant Pseudowire Configurations

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

- You can configure a single active pseudowire. The PE router configured as the primary neighbor is given preference and this connection is the one used for customer traffic. For the LDP signalling, labels are exchanged for both incoming and outgoing traffic with the primary neighbor. The LDP label advertisement is accepted from the backup neighbor, but no label advertisement is forwarded to it, leaving the pseudowire in an incomplete state. The pseudowire to the backup neighbor is completed only when the primary neighbor fails. The decision to switch between the two pseudowires is made by the device configured with the redundant pseudowires. The primary remote PE router is unaware of the redundant configuration, ensuring that traffic is always switched using just the active pseudowire.

- Alternatively, you can configure two active pseudowires, one to each of the PE routers. Using this approach, control plane signalling is completed and active pseudowires are established with both the primary and backup neighbors. However, the data plane forwarding is done only over a one of the pseudowires (designated as the active pseudowire by the local device). The other pseudowire is on standby. The active pseudowire is preferably established with the primary neighbor and can switch to the backup pseudowire if the primary fails.

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

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

- For the single active pseudowire configuration, it takes more time (compared to the two active pseudowire configuration) to switchover to the backup pseudowire when a failure is detected. This approach requires additional control plane signalling to complete the pseudowire with the backup neighbor and traffic can be lost during the switchover from primary to backup.

- If you configure two active pseudowires, bandwidth is lost on the link carrying the backup pseudowire between the remote PE router and the local device. Traffic is always duplicated over both the active and standby pseudowires. The single active pseudowire configuration does not waste bandwidth in this fashion.

Pseudowire Failure Detection

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

- Manual switchover (user initiated)

- Remote PE router withdraws the label advertisement

- LSP to the remote PE router goes down

- LDP session with the remote PE router goes down

- Local configuration changes

- Periodic pseudowire OAM procedure fails (Layer 2 circuit-based MPLS ping to the PE router fails)

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

When a failure is detected, traffic is switched from the failed active pseudowire to the redundant pseudowire. The redundant pseudowire is then designated as the active pseudowire. The switch is nonreversible, meaning that once the redundant pseudowire assumes the role of the active pseudowire at the time of a failover, it remains as the active pseudowire even though the previously active pseudowire comes up again.

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

RELATED DOCUMENTATION

Example: Configuring H-VPLS Without VLANs

Configuring Redundant Pseudowires for Layer 2 Circuits and VPLS

IN THIS SECTION

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

For an overview of how redundant pseudowires work, see *Redundant Pseudowires for Layer 2 Circuits and VPLS*.

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

Configuring Pseudowire Redundancy on the PE Router

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

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

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

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

The `backup-neighbor` statement includes the following configuration options:

- `community`—Specifies the community for the backup neighbor.

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

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

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

Configuring the Switchover Delay for the Pseudowires

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

```
switchover-delay milliseconds;
```

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

Configuring a Revert Time for the Redundant Pseudowire

You can specify a revert time for redundant Layer 2 circuit and VPLS pseudowires. When you have configured redundant pseudowires for Layer 2 circuits or VPLS, traffic is switched to the backup pseudowire in the event that the primary pseudowire fails. If you configure a revert time, when the configured time expires traffic is reverted back to the primary pseudowire, assuming the primary pseudowire has been restored.

To configure a revert time for redundant pseudowires, specify the time in seconds using the `revert-time` statement:

```
revert-time (Protocols Layer 2 Circuits) seconds maximum seconds;
```

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.

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

RELATED DOCUMENTATION

Example: Configuring Pseudowire Redundancy in a Mobile Backhaul Scenario

Example: Configuring H-VPLS Without VLANs

CHAPTER 7

Monitoring Layer 2 VPNs Using BFD

IN THIS CHAPTER

Configuring BFD for Layer 2 VPN and VPLS

The following procedure describes how to configure Bidirectional Forwarding Detection (BFD) for Layer 2 VPN and VPLS. For VPNs, you configure the BFD sessions on the interfaces carrying traffic from the PE routers to the CE routers.

The BFD protocol is a simple hello mechanism that detects failures in a network. Hello packets are sent at a specified, regular interval. A neighbor failure is detected when the routing device stops receiving a reply after a specified interval. BFD works with a wide variety of network environments and topologies. The failure detection timers for BFD have shorter time limits than default failure detection mechanisms for BGP, so they provide faster detection.

The BFD failure detection timers are adaptive and can be adjusted to be faster or slower. The lower the BFD failure detection timer value, the faster the failure detection and vice versa. For example, the timers can adapt to a higher value if the adjacency fails (that is, the timer detects failures more slowly). 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 interval by two if the local BFD instance is the reason for the session flap. The transmission 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.

1. You can enable BFD failure detection. The BFD failure detection timers are adaptive and can be adjusted to be faster or slower. The lower the BFD failure detection timer value, the faster the failure detection and vice versa. For example, the timers can adapt to a higher value if the adjacency fails (that is, the timer detects failures more slowly). 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.

To enable BFD failure detection and specify the threshold for the adaptation of the BFD session detection time, specify a time in milliseconds using the threshold statement. When the detection time adapts to a value equal to or greater than the threshold, a single trap and a single system log message are sent.

> **(i) NOTE**: The threshold time must be equal to or greater than the value specified in the minimum-interval or the minimum-receive-interval statement.

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.

2. You can specify 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. You specify the interval in milliseconds using the *minimum-interval* statement.

   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.

3. You can 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. Specify the number of milliseconds using the *minimum-receive-interval* statement.

4. You can specify that an interface be declared down when a certain number of hello packets have not been received from a neighboring router through that interface. Specify the number of hello packets by including the *multiplier* statement.

5. You can configure BFD sessions not to adapt to changing network conditions by including the *no-adaptation* statement. We recommend that you *do not* disable BFD adaptation unless it is preferable to have BFD adaptation disabled in your network.

6. Specify the transmit interval options for `bfd-liveness-detection` statement by including the *transmit-interval* statement. The negotiated transmit interval for a peer is the interval between the sending of BFD packets to peers. The receive interval for a peer is the minimum time that it requires between packets sent from its peer; the receive interval is not negotiated between peers. To determine the transmit interval, each peer compares its configured minimum transmit interval with its peer's minimum receive interval. The larger of the two numbers is accepted as the transmit interval for that peer.

   The `transmit-interval` statement specifies how often BFD statements are transmitted and includes the following options:

- `minimum-interval` *milliseconds*—Specify the minimum interval in milliseconds at which the local routing device transmits hello packets to a neighbor with which it has established a BFD session.

- `threshold` *milliseconds*—Specify the threshold for the adaptation of the BFD session transmit interval. When the transmit interval adapts to a value greater than the threshold, a single trap and a single system message are sent.

> **NOTE**: The threshold value specified in the `threshold` statement must be greater than the value specified in the `minimum-interval` statement for the `transmit-interval` statement.

7. Specify the BFD version by including the *version* statement. You can set BFD to version 1 or allow BFD to determine what version it needs to be by including the `automatic` option.

RELATED DOCUMENTATION

bfd-liveness-detection

clear bfd adaptation

BFD Support for VCCV for Layer 2 VPNs, Layer 2 Circuits, and VPLS

Bidirectional Forwarding Detection (BFD) support for virtual circuit connectivity verification (VCCV) enables you to configure a control channel for a pseudowire, in addition to the corresponding operations, administration, and management functions to be used over that control channel. For platform and Junos OS support information for BFD for VCCV or any other Junos OS feature, use Feature Explorer.

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)*. Alternatively, you can use a ping operation to detect pseudowire failures. However, the processing resources required for a ping operation are greater than what is needed for BFD. In addition, BFD is capable of detecting data plane failure faster than a VCCV ping. BFD for pseudowires is supported for Layer 2 circuits (LDP-based), Layer 2 VPNs (BGP-based), and VPLS (LDP-based or BGP-based).

Junos OS has a distributed model for the BFD for VCCV. In legacy Junos OS releases the BFD for VCCV followed a Routing Engine-based implementation. In current Junos OS releases, the BFD for VCCV follows a distributed implementation over PIC concentrators, such as DPC, FPC, and MPC.

For distributed BFD, you need to configure the lo0 interface with unit 0 and the appropriate family enabled.

> **NOTE**: For the distributed BFD for VCCV to work, you must configure MPLS family (`family mpls`) on the loopback interface.
>
> ```
> user@router# set interfaces lo0 unit 0 family mpls
> ```

> **NOTE**: On ACX Series routers, to enable BFD sessions over VCCV, you need configure control word. For VPLS, include the `control-word` statement at the [`edit routing-instances` *routing-instance-name* `protocols vpls`] hierarchy level. For Layer 2 VPN, include the `control-word` statement at the [`edit routing-instances` *routing-instance-name* `protocols l2vpn`] hierarchy level. For Layer 2 Circuit, include the `control-word` statement at the [`edit protocols l2circuit neighbor` *neighbor-ip-address* `interface` *interface-name*] hierarchy level. The `control-word` statement must also be configured at the peer device for control word negotiation to happen between the pseudowire peers.

The periodic packet management process (ppmd) on the PIC concentrators handles the periodic packet management (send and receive) for BFD for VCCV. This enables Junos OS to create more BFD for VCCV sessions, and to reduce the time taken for error detection. Similarly, the distributed implementation improves the performance of Routing Engines because the Routing Engine resources used for BFD for VCCV implementation become available for Routing Engine-related applications when the BFD for VCCV-related processing moves to the PIC concentrators. The distributed BFD for VCCV implementation also enables the BFD for VCCV sessions to remain across graceful restarts.

RELATED DOCUMENTATION

Configuring BFD for VCCV for Layer 2 VPNs, Layer 2 Circuits, and VPLS

Configuring BFD for VCCV for Layer 2 VPNs, Layer 2 Circuits, and VPLS

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 operation to detect pseudowire failures. However, the processing resources required for a ping operation 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), Layer 2 VPNs (BGP-based), and VPLS (LDP-based or BGP-based).

To configure OAM and BFD for Layer 2 VPNs, include the `oam` statement and sub-statements at the `[edit routing-instances` *routing-instance-name* `protocols l2vpn]` hierarchy level:

```
oam {
    bfd-liveness-detection;
    ping-interval ;
    ping-multiplier;
}
```

You can configure many of the same OAM statements for VPLS and Layer 2 circuits:

- To enable OAM for VPLS, configure the `oam` statement and substatements at the `[edit routing-instances` *routing-instance-name* `protocols vpls]` hierarchy level and at the `[edit routing-instances` *routing-instance-name* `protocols vpls neighbor` *address*`]` hierarchy level. The `pwe3-control-word` statement configured at the `[edit routing-instances` *routing-instance-name* `protocols l2vpn oam control-channel]` hierarchy level is not applicable to VPLS configurations.

- To enable OAM for Layer 2 circuits, configure the `oam` statement and substatements at the `[edit protocols l2circuit neighbor` *address* `interface` *interface-name*`]` hierarchy level. The `control-channel` statement and sub-statements configured at the `[edit routing-instances routing-instance-name protocols l2vpn oam]` hierarchy level do not apply to Layer 2 circuit configurations.

You can use the `show ldp database extensive` command to display information about the VCCV control channel and the `show bfd session extensive` command to display information about BFD for Layer 2 VPNs, Layer 2 circuits, and VPLS.

RELATED DOCUMENTATION

Junos OS Routing Protocols Library

2
PART

Configuring Layer 2 Circuits

CHAPTER 8

Overview

IN THIS CHAPTER

Layer 2 Circuit Overview

A Layer 2 circuit is a point-to-point Layer 2 connection transported using Multiprotocol Label Switching (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. illustrates the components of a Layer 2 circuit.

Figure 4: Components of a Layer 2 Circuit
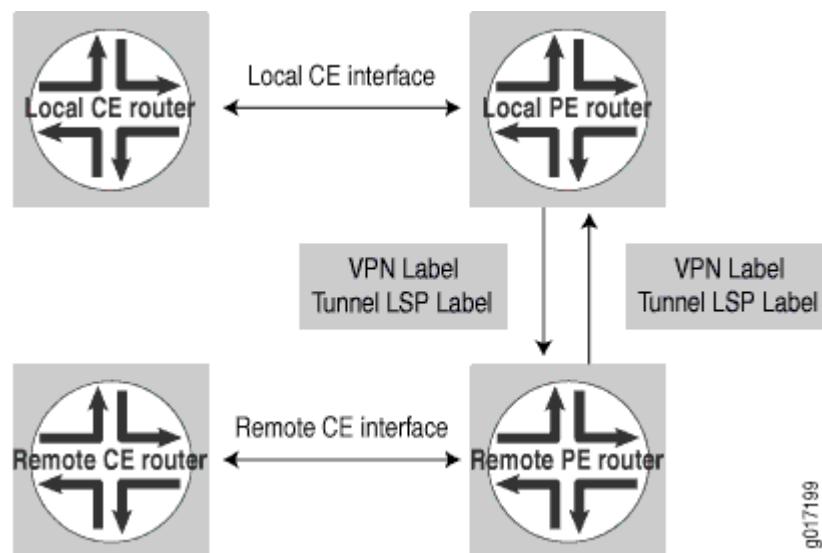
To establish a Layer 2 circuit, the Label Distribution Protocol (LDP) is used as the signaling protocol to advertise the ingress label to the remote PE routers. For this purpose, a targeted remote LDP neighbor session is established using the extended discovery mechanism described in LDP, and the session 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

Understanding Layer 3 VPNs

Layer 2 VPN Applications

Applications for Interconnecting a Layer 2 Circuit with a Layer 3 VPN

Example: Interconnecting a Layer 2 Circuit with a Layer 2 Circuit

Example: Interconnecting a Layer 2 Circuit with a Layer 3 VPN

Example: Interconnecting a Layer 2 Circuit with a Layer 2 VPN

Layer 2 Circuits Configuration Overview

IN THIS CHAPTER

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 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

Configuring Local Interface Switching in Layer 2 Circuits

IN THIS SECTION

- Configuring the Interfaces for the Local Interface Switch | **77**
- Enabling Local Interface Switching When the MTU Does Not Match | **78**

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]`

> *(i)* **NOTE**: ACX Series routers do not support the `[edit logical-systems]` hierarchy level.

The following sections describe how to configure local interface switching:

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*.

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]`

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

Configuring Interfaces for Layer 2 Circuits

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

> **NOTE**: Not all subtasks are supported on all platforms; check the CLI on your device.

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 *encapsulation* 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 79.

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.

> **ⓘ** **NOTE**: On the EX9200 switches, replace `encapsulation-type` with the *encapsulation* statement.

```
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*.

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 82.

  > (i) **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 Configuring Frame Relay Control Bit Translation.

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 MTU Advertised for a Layer 2 Circuit

By default, the MTU used to advertise a Layer 2 circuit is determined by taking the interface MTU for the associated physical interface and subtracting the encapsulation overhead for sending IP packets based on the encapsulation.

However, encapsulations that support multiple logical interfaces (and multiple Layer 2 circuits) rely on the same interface MTU (since they are all associated with the same physical interface). This can prove to be a limitation for VLAN Layer 2 circuits using the same Ethernet interface or for Layer 2 circuit DLCIs using the same Frame Relay interface.

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

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

- An explicitly configured MTU is signaled to the remote PE device. The configured MTU is also compared to the MTU received from the remote PE device. If there is a conflict, the Layer 2 circuit is taken down.

- If you configure an MTU for an ATM cell relay interface on an ATM II PIC, the configured MTU is used to compute the cell bundle size advertised for that Layer 2 circuit, instead of the default interface MTU.

- A configured MTU is used only in the control plane. It is not enforced in the data plane. You need to ensure that the CE device for a given Layer 2 circuit uses the correct MTU for data transmission.

To configure the MTU for a Layer 2 circuit, include the `mtu` statement at the `[edit protocols l2circuit neighbor address interface interface-name]` hierarchy level.

```
mtu mtu-number;
```

Enabling the Layer 2 Circuit When the MTU Does Not Match

You can configure the Junos OS to allow a Layer 2 circuit to be established even though the MTU configured on the PE router does not match the MTU configured on the remote PE router by including the `ignore-mtu-mismatch` statement at the `[edit protocols l2circuit neighbor address interface interface-name]` hierarchy level.

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*.

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 192.0.2.0;
        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 `192.0.2.0`. 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 Pseudowire Redundancy on the PE Router*.

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 MPLS Applications User Guide and Junos OS Network Interfaces Library for Routing Devices.

> (i) **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 more information about how to configure ATM2 IQ interfaces, see theJunos 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 Policies for Layer 2 Circuits

IN THIS SECTION

- Configuring the Layer 2 Circuit Community | **89**
- Configuring the Policy Statement for the Layer 2 Circuit Community | **90**
- Verifying the Layer 2 Circuit Policy Configuration | **92**

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*.

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 Routing Policies, Firewall Filters, and Traffic Policers User Guide.

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 MPLS Applications User Guide.

Configuring Protection Features for Layer 2 Circuits

IN THIS CHAPTER

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: Egress Protection LSP
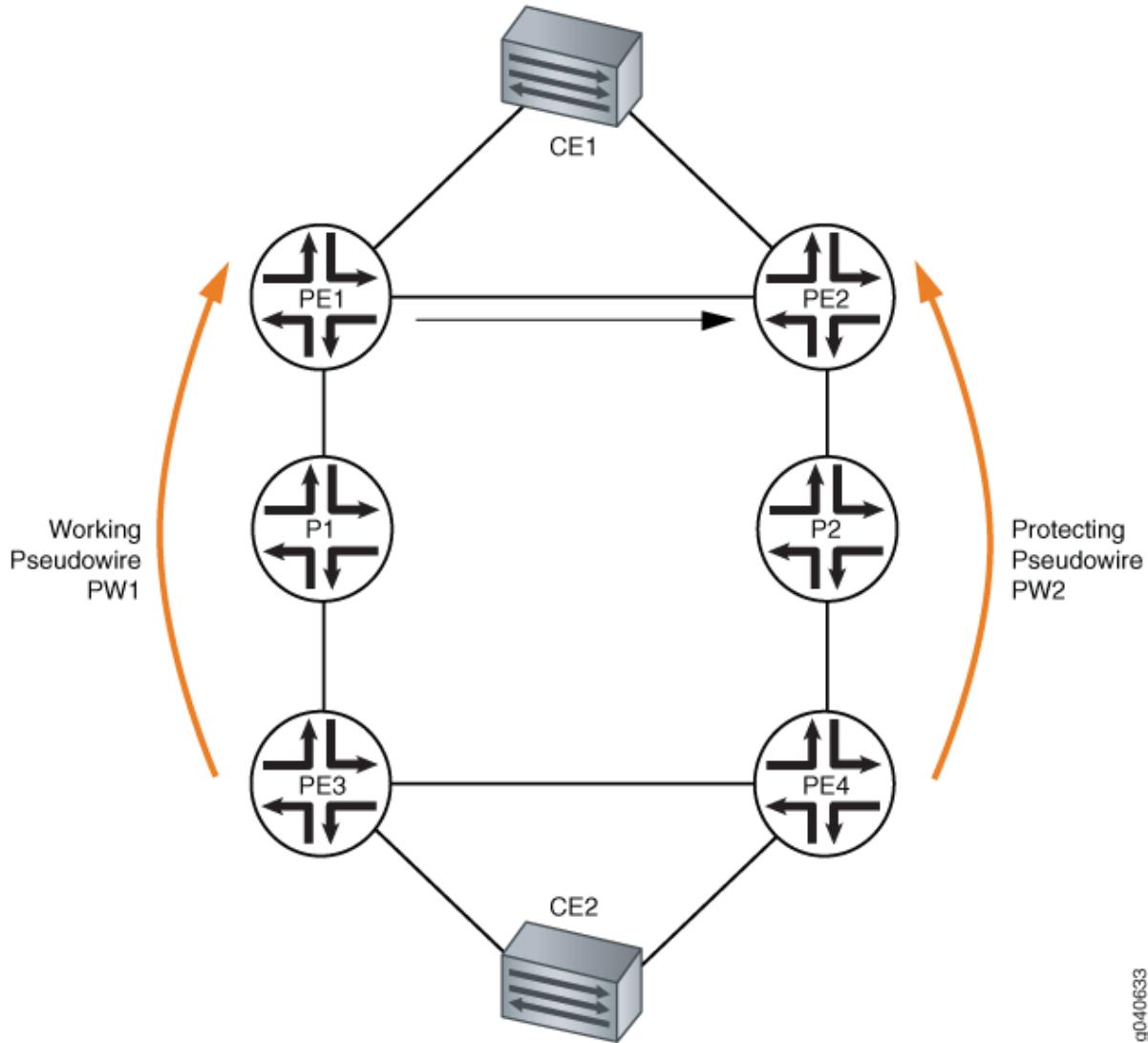
Device CE1 is multihomed to router PE1 and router PE2. Device CE2 is multihomed to router PE3 and router PE4. There are two paths connecting devices CE1 and CE2. The working path is CE2-PE3-P1-PE1-CE1, using pseudowire PW1. The protecting path is CE2-PE4-P2-PE2-CE1, using pseudowire PW2. Normally, traffic 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 , 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.

Example: Configuring Layer 2 Circuit Switching Protection

IN THIS SECTION

Unlike Layer 2 circuit protect interfaces (see *Example: Configuring Layer 2 Circuit Protect Interfaces*), 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 5G Universal Routing Platforms

- 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 6 on page 97) or indirectly using a pseudowire configured through an intermediate PE router (as shown in Figure 7 on page 98 and Figure 8 on page 98). 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.

> (i) **NOTE**: Non-stop routing (*NSR*) and graceful routing engine switchover (*GRES*) do not support Layer 2 circuit switching protection.

Topology

Figure 6 on page 97 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 6: Connection Protection Enabled Between Router PE1 and Router PE2
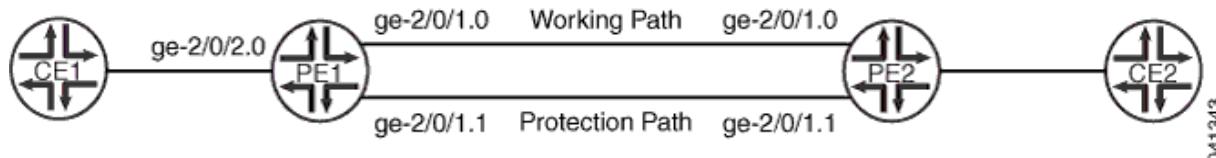
In Figure 7 on page 98 and Figure 8 on page 98, there are two OAM sessions running between Router PE1 and Router PE2. For Figure 2, one OAM session is configured over the working path between Router PE1 and Router PE2. The other OAM session is configured over the protection path between Router PE1 and Router PE3 to Router PE2.

Figure 7: Connection Protection Using a Pseudowire Configured through Router PE3 as the Protection Path
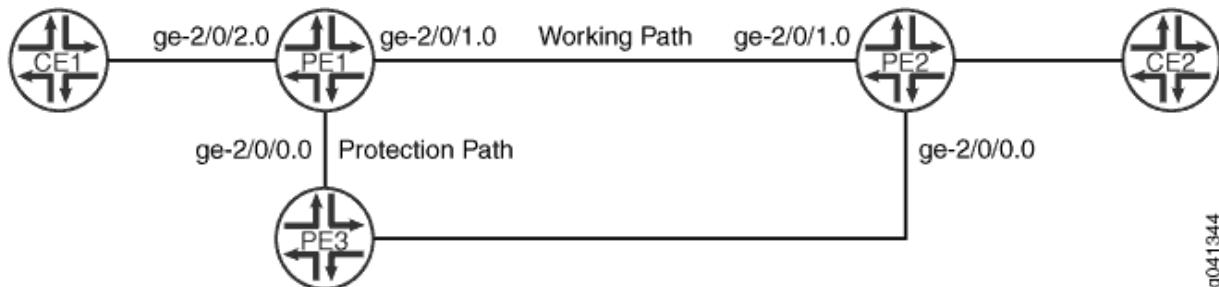
For Figure 8 on page 98, one OAM session is configured over the working path, the pseudowire between Router PE1 and Router PE3, then to Router PE2. The other OAM session is configured on the protect path between Router PE1 and Router PE2.

Figure 8: Connection Protection Using a Pseudowire Configured through Router PE3 as the Working Path
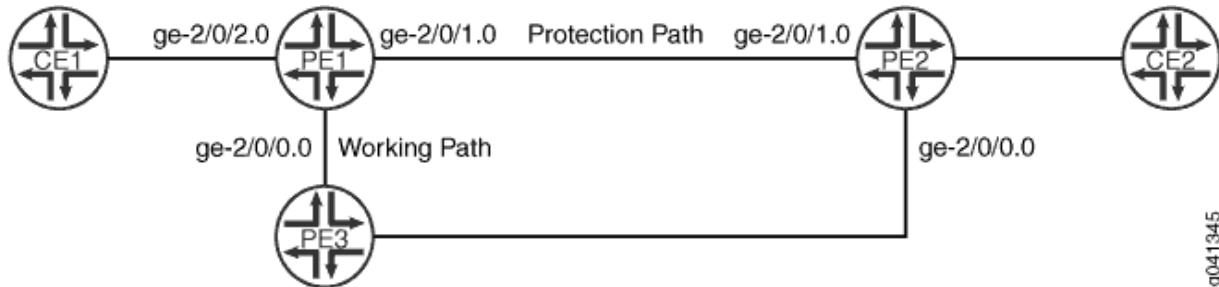
Configuration

IN THIS SECTION

The following sections describe how to configure each of the variations of Layer 2 circuit connection protection:

Configuring Connection Protection Between Two PE Routers

Step-by-Step Procedure

To configure Layer 2 Circuit switching protection as shown in on Router PE1:

1. Configure the Layer 2 circuit on Router PE1.

   ```
   [edit protocols l2circuit]
   user@PE1# set local-switching interface ge-2/0/2.0 connection-protection
   user@PE1# set local-switching interface ge-2/0/2.0 end-interface interface ge-2/0/1.0
   user@PE1# 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]
   user@PE1# set policy-statement protection-policy then load-balance per-packet
   ```

3. Enable the routing policy on Router PE1.

   ```
   [edit routing-options]
   user@PE1# set 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]
   user@PE1# 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]
   user@PE1# set connectivity-fault-management maintenance-domain l2circuit-example-md
   ```

```
maintenance-association working continuity-check interval 100ms
user@PE1# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association working mep 1000 interface ge-2/0/1.0
user@PE1# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association working mep 1000 interface working
user@PE1# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association working mep 1000 direction down
user@PE1# 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]
user@PE1# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association protection continuity-check interval 100ms
user@PE1# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association protection mep 1001 interface ge-2/0/1.1
user@PE1# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association protection mep 1001 interface protect
user@PE1# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association protection mep 1001 direction down
user@PE1# set 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.

```
[edit protocols oam ethernet]
user@PE2# 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]
user@PE2# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association working continuity-check interval 100ms
user@PE2# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association working mep 103 interface ge-2/0/1.0
user@PE2# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association working mep 103 interface working
user@PE2# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association working mep 103 direction down
```

```
user@PE2# 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]
user@PE2# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association protection continuity-check interval 100ms
user@PE2# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association protection mep 104 interface ge-2/0/1.1
user@PE2# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association protection mep 104 interface protect
user@PE2# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association protection mep 104 direction down
user@PE2# 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;
        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;
            }
        }
    }
}
```

Verifying that OAM CFM Connections are Active

Purpose

Verify that the CFM connections are active on each of the PE routers.

Action

Execute the following command on each of the PE routers.

1. Verify that the CFM working connection on Router PE1 is active.

```
user@ PE1> show oam ethernet connectivity-fault-management mep-database maintenance-domain
l2circuit-example-md maintenance-association working
Interface status: Active, Link status: Up
```

2. Verify that the CFM protect connection on Router PE1 is active

```
user@ PE2> show oam ethernet connectivity-fault-management mep-database maintenance-domain
l2circuit-example-md maintenance-association protection
Interface status: Active, Link status: Up
```

3. Verify that the CFM working connection on Router PE2 is active.

```
user@ PE2> show oam ethernet connectivity-fault-management mep-database maintenance-domain
l2circuit-example-md maintenance-association working
Interface status: Active, Link status: Up
```

4. Verify that the CFM protect connection on Router PE2 is active.

```
user@ PE2> show oam ethernet connectivity-fault-management mep-database maintenance-domain
l2circuit-example-md maintenance-association protection
Interface status: Active, Link status: Up
```

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 on Router PE1:

1. Configure the Layer 2 circuit on Router PE1.

```
 [edit protocols l2circuit]
 user@PE1# set local-switching interface ge-2/0/2.0 connection-protection
 user@PE1# set local-switching interface ge-2/0/2.0 backup-neighbor 192.0.2.2 virtual-
 circuit-id 2
 user@PE1# set local-switching interface ge-2/0/2.0 backup-neighbor 192.0.2.2 community
```

```
example
user@PE1# set 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]
user@PE1# set policy-statement load-balance then load-balance per-packet
user@PE1# set policy-statement protection-policy term protect from community example
user@PE1# set policy-statement protection-policy term protect then install-nexthop lsp-
regex lsp-protect-*
```

3. Configure the community.

```
[edit policy-options]
user@PE1# set community example members 65100:10
```

4. Configure the routing options on Router PE1.

```
[edit routing-options]
user@PE1# set forwarding-table export load-balance
```

5. 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]
user@PE1# set connectivity-fault-management maintenance-domain l2circuit-example-md level 5
```

6. Configure OAM on Router PE1 for the working path.

```
[edit protocols oam ethernet]
user@PE1# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association working mep 1000 interface ge-2/0/1.0
user@PE1# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association working mep 1000 direction down
user@PE1# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association working mep 1000 remote-mep 103
```

7. Configure OAM on Router PE1 for the protection path.

```
[edit protocols oam ethernet]
user@PE1# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association protection mep 1001 interface ge-2/0/0.0
user@PE1# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association protection mep 1001 direction down
user@PE1# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association protection mep 1001 remote-mep 104
```

8. Configure OAM on Router PE2 to setup the maintenance domain.

```
[edit protocols oam ethernet]
user@PE2# set connectivity-fault-management maintenance-domain l2circuit-example-md level 5
```

9. Configure OAM on Router PE2 for the working path.

```
[edit protocols oam ethernet]
user@PE2# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association working mep 103 interface ge-2/0/1.0
user@PE2# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association working mep 103 direction down
user@PE2# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association working mep 103 remote-mep 1000
```

10. Configure OAM on Router PE2 for the protection path.

```
[edit protocols oam ethernet]
user@PE2# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association protection mep 104 interface ge-2/0/0.0
user@PE2# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association protection mep 104 direction down
user@PE2# 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 192.0.2.2 {
            virtual-circuit-id 2;
            community example;
        }
        end-interface {
            interface ge-2/0/1.0;
        }
    }
}
```

```
user@host> show policy-options
community example members 65100:10;
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;
        }
    }
  }
}
```

Verifying that OAM CFM Connections are Active

Purpose

Verify that the CFM connections are active on each of the PE routers.

Action

Execute the following command on each of the PE routers.

1. Verify that the CFM working connection on Router PE1 is active.

```
user@ PE1> show oam ethernet connectivity-fault-management mep-database maintenance-domain
l2circuit-example-md maintenance-association working
Interface status: Active, Link status: Up
```

2. Verify that the CFM protect connection on Router PE1 is active

```
user@ PE2> show oam ethernet connectivity-fault-management mep-database maintenance-domain
l2circuit-example-md maintenance-association protection
Interface status: Active, Link status: Up
```

3. Verify that the CFM working connection on Router PE2 is active.

```
user@ PE2> show oam ethernet connectivity-fault-management mep-database maintenance-domain
l2circuit-example-md maintenance-association working
Interface status: Active, Link status: Up
```

4. Verify that the CFM protect connection on Router PE2 is active.

```
user@ PE2> show oam ethernet connectivity-fault-management mep-database maintenance-domain
l2circuit-example-md maintenance-association protection
Interface status: Active, Link status: Up
```

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 on Router PE1:

1. Configure the Layer 2 circuit on Router PE1.

```
[edit protocols l2circuit]
user@PE1# set neighbor 192.0.2.2 interface ge-2/0/2.0 virtual-circuit-id 2
user@PE1# set neighbor 192.0.2.2 interface ge-2/0/2.0 community example
user@PE1# set neighbor 192.0.2.2 interface ge-2/0/2.0 connection-protection
user@PE1# set neighbor 192.0.2.2 interface ge-2/0/2.0 backup-neighbor 192.0.2.3 virtual-
circuit-id 3
user@PE1# set neighbor 192.0.2.2 interface ge-2/0/2.0 backup-neighbor 192.0.2.3 standby
```

2. Configure the policies on Router PE1.

```
[edit policy-options]
user@PE1# set policy-statement load-balance then load-balance per-packet
user@PE1# set policy-statement protection-policy term protect from community example
user@PE1# set policy-statement protection-policy term protect then install-nexthop lsp-
regex lsp-primary
```

3. Configure the community.

```
[edit policy-options]
user@PE1# set community example members 65100:10
```

4. Configure the routing options on Router PE1.

```
[edit routing-options]
user@PE1# set forwarding-table export load-balance
```

5. 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]
user@PE1# set connectivity-fault-management maintenance-domain l2circuit-example-md level 5
```

6. Configure OAM on Router PE1 for the working path.

```
[edit protocols oam ethernet]
user@PE1# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association working mep 1000 interface ge-2/0/0.0
user@PE1# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association working mep 1000 direction down
user@PE1# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association working mep 1000 remote-mep 103
```

7. Configure OAM on Router PE1 for the protection path.

```
[edit protocols oam ethernet]
user@PE1# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association protection mep 1001 interface ge-2/0/1.0
user@PE1# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association protection mep 1001 direction down
user@PE1# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association protection mep 1001 remote-mep 104
```

8. Configure OAM on Router PE2 to setup the maintenance domain.

```
[edit protocols oam ethernet]
user@PE2# set connectivity-fault-management maintenance-domain l2circuit-example-md level 5
```

9. Configure OAM on Router PE2 for the working path.

```
[edit protocols oam ethernet]
user@PE2# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association working mep 103 interface ge-2/0/0.0
user@PE2# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association working mep 103 direction down
user@PE2# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association working mep 103 remote-mep 1000
```

10. Configure OAM on Router PE2 for the protection path.

```
[edit protocols oam ethernet]
user@PE2# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association protection mep 104 interface ge-2/0/1.0
user@PE2# set connectivity-fault-management maintenance-domain l2circuit-example-md
maintenance-association protection mep 104 direction down
user@PE2# 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 192.0.2.2 {
    interface ge-2/0/2.0 {
        virtual-circuit-id 2;
        community example;
        connection-protection;
        backup-neighbor 192.0.2.3 {
            virtual-circuit-id 3;
            standby;
        }
```

```
        }
    }
```

```
user@host> show policy-options
community example members 65100:10;
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;
            }
        }
    }
}
```

Verifying that OAM CFM Connections are Active

Purpose

Verify that the CFM connections are active on each of the PE routers.

Action

Execute the following command on each of the PE routers.

1. Verify that the CFM working connection on Router PE1 is active.

```
user@ PE1> show oam ethernet connectivity-fault-management mep-database maintenance-domain
l2circuit-example-md maintenance-association working
Interface status: Active, Link status: Up
```

2. Verify that the CFM protect connection on Router PE1 is active

```
user@ PE2> show oam ethernet connectivity-fault-management mep-database maintenance-domain
l2circuit-example-md maintenance-association protection
Interface status: Active, Link status: Up
```

3. Verify that the CFM working connection on Router PE2 is active.

```
user@ PE2> show oam ethernet connectivity-fault-management mep-database maintenance-domain
l2circuit-example-md maintenance-association working
Interface status: Active, Link status: Up
```

4. Verify that the CFM protect connection on Router PE2 is active.

```
user@ PE2> show oam ethernet connectivity-fault-management mep-database maintenance-domain
l2circuit-example-md maintenance-association protection
Interface status: Active, Link status: Up
```

RELATED DOCUMENTATION

Example: Configuring Layer 2 Circuit Protect Interfaces

CHAPTER 11

Monitoring Layer 2 Circuits with BFD

IN THIS CHAPTER

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 192.0.2.1 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 192.0.2.1 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 192.0.2.1 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 192.0.2.1 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 192.0.2.1 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 192.0.2.1 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 192.0.2.1 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

Example: Configuring BFD for VCCV for Layer 2 Circuits

IN THIS SECTION

- Requirements | **119**
- Overview | **120**
- Configuration | **121**
- Verification | **127**

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 5G Universal Routing Platforms

- 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.

Figure 9: BFD for VCCV for Layer 2 Circuits
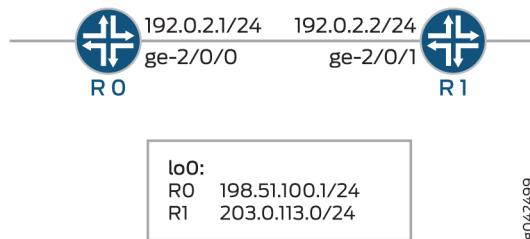
Configuration

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 192.0.2.1/24
set interfaces ge-2/0/0 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 198.51.100.0/24
set routing-options nonstop-routing
set routing-options static route 203.0.113.0/24 next-hop 192.0.2.2
set routing-options router-id 198.51.100.0
set protocols rsvp interface ge-2/0/0.0
set protocols mpls label-switched-path lsp1 to 203.0.113.0
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 203.0.113.0 interface ge-1/1/9.0 virtual-circuit-id 1
set protocols l2circuit neighbor 203.0.113.0 interface ge-1/1/9.0 oam bfd-liveness-detection
minimum-interval 300
set protocols l2circuit neighbor 203.0.113.0 interface ge-1/1/9.0 oam bfd-liveness-detection
minimum-receive-interval 10
set protocols l2circuit neighbor 203.0.113.0 interface ge-1/1/9.0 oam bfd-liveness-detection
```

```
multiplier 3
set protocols l2circuit neighbor 203.0.113.0 interface ge-1/1/9.0 oam bfd-liveness-detection
transmit-interval minimum-interval 5
set protocols l2circuit neighbor 203.0.113.0 interface ge-1/1/9.0 oam bfd-liveness-detection
transmit-interval threshold 30
set protocols l2circuit neighbor 203.0.113.0 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 192.0.2.2/24
set interfaces ge-2/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 203.0.113.0/24
set routing-options static route 198.51.100.0/24 next-hop 192.0.2.1
set routing-options router-id 203.0.113.0
set protocols rsvp interface ge-2/0/1.0
set protocols mpls label-switched-path lsp2 to 198.51.100.0
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 198.51.100.0 interface ge-1/1/9.0 virtual-circuit-id 1
set protocols l2circuit neighbor 198.51.100.0 interface ge-1/1/9.0 oam bfd-liveness-detection
minimum-interval 300
set protocols l2circuit neighbor 198.51.100.0 interface ge-1/1/9.0 oam bfd-liveness-detection
minimum-receive-interval 10
set protocols l2circuit neighbor 198.51.100.0 interface ge-1/1/9.0 oam bfd-liveness-detection
multiplier 3
set protocols l2circuit neighbor 198.51.100.0 interface ge-1/1/9.0 oam bfd-liveness-detection
transmit-interval minimum-interval 5
set protocols l2circuit neighbor 198.51.100.0 interface ge-1/1/9.0 oam bfd-liveness-detection
transmit-interval threshold 30
set protocols l2circuit neighbor 198.51.100.0 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:

> ℹ **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 192.0.2.1/24
user@R0# set ge-2/0/0 unit 0 family mpls
user@R0# set lo0 unit 0 family inet address 198.51.100.0/24
```

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 203.0.113.0/24 next-hop 192.0.2.2
user@R0# set router-id 198.51.100.0
```

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 203.0.113.0
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 203.0.113.0 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 203.0.113.0 interface ge-1/1/9.0 oam bfd-liveness-detection minimum-
interval 300
user@R0# set neighbor 203.0.113.0 interface ge-1/1/9.0 oam bfd-liveness-detection minimum-
receive-interval 10
user@R0# set neighbor 203.0.113.0 interface ge-1/1/9.0 oam bfd-liveness-detection multiplier 3
user@R0# set neighbor 203.0.113.0 interface ge-1/1/9.0 oam bfd-liveness-detection transmit-
interval minimum-interval 5
```

```
user@R0# set neighbor 203.0.113.0 interface ge-1/1/9.0 oam bfd-liveness-detection transmit-
interval threshold 30
user@R0# set neighbor 203.0.113.0 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 192.0.2.1/24;
        }
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 198.51.100.0/24;
        }
```

```
    }
}
```

```
user@R0# show protocols
rsvp {
    interface ge-2/0/0.0;
}
mpls {
    label-switched-path lsp1 {
        to 203.0.113.0;
    }
    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 203.0.113.0 {
        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 203.0.113.0/24 next-hop 192.0.2.2;
}
router-id 198.51.100.0;
```

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

Verification

IN THIS SECTION

Verify that the configuration is working properly.

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: 203.0.113.0
    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: 203.0.113.0, 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

203.0.113.7             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

203.0.113.7                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 203.0.113.0, 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

CHAPTER 12

Troubleshooting Layer 2 Circuits

IN THIS CHAPTER

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

3
PART

Configuration Statements and Operational Commands

-

Junos CLI Reference Overview

We've consolidated all Junos CLI commands and configuration statements in one place. Read this guide to learn about the syntax and options that make up the statements and commands. Also understand the contexts in which you'll use these CLI elements in your network configurations and operations.

- Junos CLI Reference

Click the links to access Junos OS and Junos OS Evolved configuration statement and command summary topics.

- Configuration Statements

- Operational Commands