JUNIPER NETWORKS

Junos® OS

Layer 2 VPNs Configuration Guide

Release
12.1

Published: 2012-03-13

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

Junos® OS Layer 2 VPNs Configuration Guide
12.1

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

Table of Contents

List of Figures

List of Tables

About the Documentation

Documentation and Release Notes

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

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

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

Using the Examples in This Manual

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

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

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

Merging a Full Example

To merge a full example, follow these steps:

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

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

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

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

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

Merging a Snippet

To merge a snippet, follow these steps:

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

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

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

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

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

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

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

For more information about the **load** command, see the *Junos OS CLI User Guide*.

Documentation Conventions

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

Table 1: Notice Icons

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

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

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the **configure** command: user@host> **configure**
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host>` **show chassis alarms** `No alarms currently active`
Italic text like this	• Introduces important new terms. • Identifies book names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS System Basics Configuration Guide* • RFC 1997, *BGP Communities Attribute*

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

Convention	Description	Examples
Italic text like this	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*
Text like this	Represents names of configuration statements, commands, files, and directories; interface names; configuration hierarchy levels; or labels on routing platform components.	• To configure a stub area, include the **stub** statement at the **[edit protocols ospf area area-id]** hierarchy level. • The console port is labeled **CONSOLE**.
< > (angle brackets)	Enclose optional keywords or variables.	**stub** <**default-metric** *metric*>**;**
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	**broadcast | multicast** (***string1*** **|** ***string2*** **|** ***string3***)
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	**rsvp { # Required for dynamic MPLS only**
[ ] (square brackets)	Enclose a variable for which you can substitute one or more values.	**community name members [** ***community-ids*** **]**

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

; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
J-Web GUI Conventions		
Bold text like this	Represents J-Web graphical user interface (GUI) items you click or select.	• In the Logical Interfaces box, select **All Interfaces**. • To cancel the configuration, click **Cancel**.
> (bold right angle bracket)	Separates levels in a hierarchy of J-Web selections.	In the configuration editor hierarchy, select **Protocols>Ospf**.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can send your comments to **techpubs-comments@juniper.net**, or fill out the documentation feedback form at

https://www.juniper.net/cgi-bin/docbugreport/ . If you are using e-mail, be sure to include the following information with your comments:

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

Requesting Technical Support

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

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

Self-Help Online Tools and Resources

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

- Find CSC offerings: http://www.juniper.net/customers/support/
- Search for known bugs: http://www2.juniper.net/kb/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/
- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: https://www.juniper.net/alerts/
- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

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

Opening a Case with JTAC

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

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

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

PART 1

Overview

CHAPTER 1

Introduction to Layer 2 VPNs

Layer 2 VPN Overview

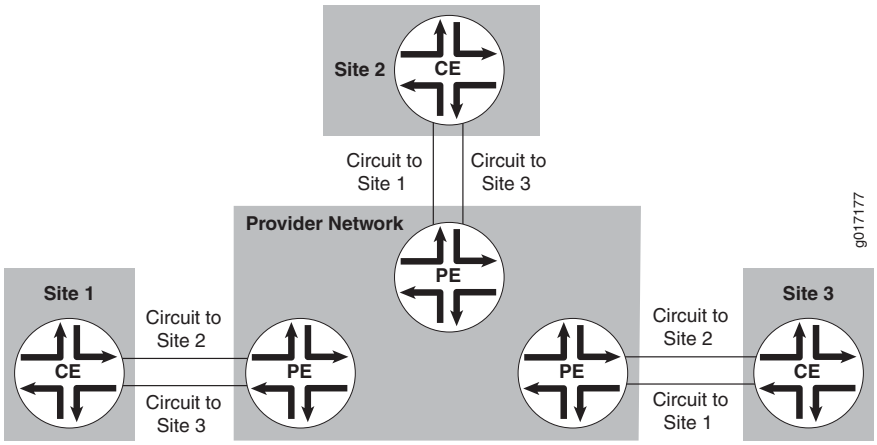
Implementing a Layer 2 VPN on a router is similar to implementing a VPN using a Layer 2 technology such as Asynchronous Transfer Mode (ATM) or Frame Relay. However, for a Layer 2 VPN on a router, traffic is forwarded to the router in a Layer 2 format. It is carried by MPLS over the service provider's 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 security and privacy of an MPLS Layer 2 VPN are equal to those of an ATM or Frame Relay VPN.

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. Figure 1 on page 4 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.

CHAPTER 2

Introduction to Layer 2 VPNs Configuration Example

Layer 2 VPN to Layer 2 VPN Connections

As the need to link different Layer 2 services to one another for expanded service offerings grows, Layer 2 MPLS VPN services are increasingly in demand. Junos OS enables you to terminate Layer 2 VPN into Layer 2 VPN (also known as Layer 2 VPN stitching) using the Layer 2 interworking (iw0) interface.

Another way to do this is to use a Tunnel Services PIC to loop packets out and back from the Packet Forwarding Engine (PFE), to link together Layer 2 networks. The Layer 2 interworking software interface avoids the need for the Tunnel Services PIC and overcomes the limitation of bandwidth constraints imposed by the Tunnel Services PIC.

Related Documentation

- Example: Interconnecting a Layer 2 VPN with a Layer 2 VPN on page 37

PART 2

Configuration

CHAPTER 3

Configuring Layer 2 VPNs

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 data-link connection identifier [DLCI], 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 the *Junos OS Routing Protocols Configuration Guide*.

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:

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:

```
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]**

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:

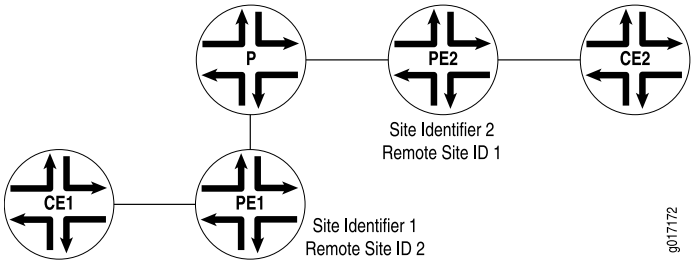
- ***site-name***—Name of the site.
- **site-identifier *identifier***—Unsigned 16-bit number greater than zero that uniquely identifies the local Layer 2 VPN site. The site identifier corresponds to the remote site ID configured on another site within the same VPN.
- **interface *interface-name***—The name of the interface and, optionally, a remote site ID for remote site connections. See "Configuring the Remote Site ID" on page 12.

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" on page 17.

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

```
encapsulation-type 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 *Junos OS MPLS Applications Configuration Guide*.

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" on page 14.

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 ATM CCC encapsulation type on the logical interface, you should configure a compatible ATM 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.

For more information about how to configure interfaces and interface encapsulations, see the *Junos OS Network Interfaces Configuration Guide*.

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.

For information about how to configure encapsulations for Layer 2 circuits, see Configuring Interfaces 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" on page 14.

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*;

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. Ethernet TCC encapsulation is supported on 1-port Gigabit Ethernet, 2-port Gigabit Ethernet, 4-port Gigabit Ethernet, and 4-port Fast Ethernet Physical Interface Cards (PICs) only.

For more information about how to configure interfaces and interface encapsulations, see the *Junos OS Network Interfaces Configuration Guide*.

Configuring Traffic Policing in Layer 2 VPNs

You can use policing to control the amount of traffic flowing over the interfaces servicing a Layer 2 VPN. If policing is disabled on an interface, all the available bandwidth on a Layer 2 VPN tunnel can be used by a single CCC or TCC interface.

For more information about the **policer** statement, see the *Junos OS Policy Framework Configuration Guide*.

To enable Layer 2 VPN policing on an interface, include the **policer** statement:

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

If you configure CCC encapsulation, you can include the **policer** statement at the following hierarchy levels:

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

If you configure TCC encapsulation, you can include the **policer** statement at the following hierarchy levels:

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

For information about how to configure the encapsulation type, see "Configuring the Encapsulation Type" on page 14.

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 (ATM, Frame Relay, Ethernet, and so on) 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 Interfaces for Layer 2 Circuits and the *Layer 2 Circuits Feature Guide*.

NOTE: Use the **no-control-word** statement to disable the control word when the topology uses generic routing encapsulation (GRE) as the connection mechanism between PEs, and one of the PEs is an M Series router.

CHAPTER 4

Layer 2 VPNs Configuration Example

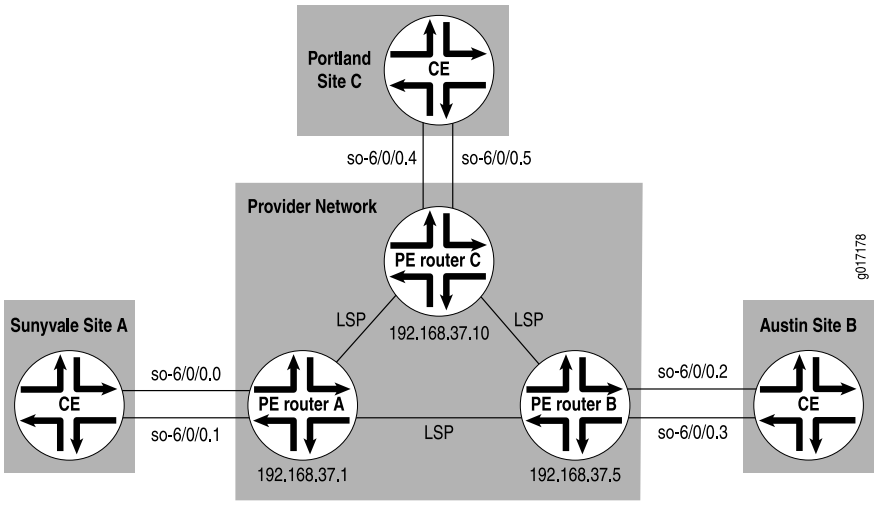
Layer 2 VPN Configuration Example

The following sections explain how to configure Layer 2 VPN functionality on the provider edge (PE) routers connected to each site:

Simple Full-Mesh Layer 2 VPN Overview

In the sections that follow, you configure a simple full-mesh Layer 2 VPN spanning three sites: Sunnyvale, Austin, and Portland. Each site connects to a PE router. The customer edge (CE) routers at each site use Frame Relay to carry Layer 2 traffic to the PE routers. Since this example uses a full-mesh topology between all three sites, each site requires two logical interfaces (one for each of the other CE routers), although only one physical link is needed to connect each PE router to each CE router. Figure 3 on page 22 illustrates the topology of this Layer 2 VPN.

Figure 3: Example of a Simple Full-Mesh Layer 2 VPN Topology



Enabling an IGP on the PE Routers

To allow the PE routers to exchange routing information among themselves, you must configure an interior gateway protocol (IGP) or static routes on these routers. You configure the IGP on the master instance of the routing protocol process (**rpd**) (that is, at the **[edit protocols]** hierarchy level), not within the Layer 2 VPN routing instance (that is, not at the **[edit routing-instances]** hierarchy level). Turn on traffic engineering on the IGP.

You configure the IGP in the standard way. This example does not include this portion of the configuration.

Configuring MPLS LSP Tunnels Between the PE Routers

In this configuration example, RSVP is used for MPLS signaling. Therefore, in addition to configuring RSVP, you must create an MPLS label-switched path (LSP) to tunnel the VPN traffic.

On Router A, enable RSVP and configure one end of the MPLS LSP tunnel to Router B. When configuring the MPLS LSP, include all interfaces using the **interface all** statement.

```
[edit]
protocols {
   rsvp {
      interface all;
   }
   mpls {
      interface all;
      label-switched-path RouterA-to-RouterB {
         to 192.168.37.5;
         primary Path-to-RouterB;
      }
      label-switched-path RouterA-to-RouterC {
         to 192.168.37.10;
         primary Path-to-RouterC;
```

```
    }
  }
}
```

On Router B, enable RSVP and configure the other end of the MPLS LSP tunnel. Again, configure the interfaces by using the **interface all** statement.

```
[edit]
protocols {
  rsvp {
    interface all;
  }
  mpls {
    interface all;
    label-switched-path RouterB-to-RouterA {
      to 192.168.37.1;
      primary Path-to-RouterA;
    }
    label-switched-path RouterB-to-RouterC {
      to 192.168.37.10;
      primary Path-to-RouterC;
    }
  }
}
```

On Router C, enable RSVP and configure the other end of the MPLS LSP tunnel. Again, configure all interfaces using the **interface all** statement.

```
[edit]
protocols {
  rsvp {
    interface all;
  }
  mpls {
    interface all;
    label-switched-path RouterC-to-RouterA {
      to 192.168.37.1;
      primary Path-to-RouterA;
    }
    label-switched-path RouterC-to-RouterB {
      to 192.168.37.5;
      primary Path-to-RouterB;
    }
  }
}
```

Configuring IBGP on the PE Routers

On the PE routers, configure an IBGP session with the following parameters:

- Layer 2 VPN—To indicate that the IBGP session is for a Layer 2 VPN, include the **family l2vpn** statement.
- Local address—The IP address in the **local-address** statement is the same as the address configured in the **to** statement at the **[edit protocols mpls label-switched-path**

lsp-path-name] hierarchy level on the remote PE router. The IBGP session for Layer 2 VPNs runs through this address.

- Neighbor address—Include the **neighbor** statement, specifying the IP address of the neighboring PE router.

On Router A, configure IBGP:

```
[edit]
protocols {
  bgp {
    import match-all;
    export match-all;
    group pe-pe {
      type internal;
      neighbor 192.168.37.5 {
        local-address 192.168.37.1;
        family l2vpn {
          signaling;
        }
      }
      neighbor 192.168.37.10 {
        local-address 192.168.37.1;
        family l2vpn {
          signaling;
        }
      }
    }
  }
}
```

On Router B, configure IBGP:

```
[edit]
protocols {
  bgp {
    local-address 192.168.37.5;
    import match-all;
    export match-all;
    group pe-pe {
      type internal;
      neighbor 192.168.37.1 {
        local-address 192.168.37.5;
        family l2vpn {
          signaling;
        }
      }
      neighbor 192.168.37.10 {
        local-address 192.168.37.5;
        family l2vpn {
          signaling;
        }
      }
    }
  }
}
```

On Router C, configure IBGP:

```
[edit]
protocols {
    bgp {
        local-address 192.168.37.10;
        import match-all;
        export match-all;
        group pe-pe {
            type internal;
            neighbor 192.168.37.1 {
                local-address 192.168.37.10;
                family l2vpn {
                    signaling;
                }
            }
            neighbor 192.168.37.5 {
                local-address 192.168.37.10;
                family l2vpn {
                    signaling;
                }
            }
        }
    }
}
```

Configuring Routing Instances for Layer 2 VPNs on the PE Routers

The three PE routers service the Layer 2 VPN, so you need to configure a routing instance on each router. For the VPN, you must define the following in each routing instance:

- Route distinguisher, which must be unique for each routing instance on the PE router. It is used to distinguish the addresses in one VPN from those in another VPN.
- Instance type of **l2vpn**, which configures the router to run a Layer 2 VPN.
- Interfaces connected to the CE routers.
- VPN routing and forwarding (VRF) import and export policies, which must be the same on each PE router that services the same VPN and are used to control the network topology. Unless the import policy contains only a **then reject** statement, it must include a reference to a community. Otherwise, when you attempt to commit the configuration, the commit operation fails.

On Router A, configure the following routing instance for the Layer 2 VPN:

```
[edit]
routing-instances {
    VPN-Sunnyvale-Portland-Austin {
        instance-type l2vpn;
        interface so-6/0/0.0;
        interface so-6/0/0.1;
        route-distinguisher 100:1;
        vrf-import vpn-SPA-import;
        vrf-export vpn-SPA-export;
        protocols {
```

```
        l2vpn {
          encapsulation-type frame-relay;
          site Sunnyvale {
            site-identifier 1;
            interface so-6/0/0.0 {
              remote-site-id 2;
            }
            interface so-6/0/0.1 {
              remote-site-id 3;
            }
          }
        }
      }
    }
  }
```

On Router B, configure the following routing instance for the Layer 2 VPN:

```
[edit]
routing-instances {
  VPN-Sunnyvale-Portland-Austin {
    instance-type l2vpn;
    interface so-6/0/0.2;
    interface so-6/0/0.3;
    route-distinguisher 100:1;
    vrf-import vpn-SPA-import;
    vrf-export vpn-SPA-export;
    protocols {
      l2vpn {
        encapsulation-type frame-relay;
        site Austin {
          site-identifier 2;
          interface so-6/0/0.2 {
            remote-site-id 1;
          }
          interface so-6/0/0.3 {
            remote-site-id 3;
          }
        }
      }
    }
  }
}
```

On Router C, configure the following routing instance for the Layer 2 VPN:

```
[edit]
routing-instances {
  VPN-Sunnyvale-Portland-Austin {
    instance-type l2vpn;
    interface so-6/0/0.4;
    interface so-6/0/0.5;
    route-distinguisher 100:1;
    vrf-import vpn-SPA-import;
    vrf-export vpn-SPA-export;
    protocols {
      l2vpn {
```

```
            encapsulation-type frame-relay;
            site Portland {
               site-identifier 3;
               interface so-6/0/0.4 {
                  remote-site-id 1;
               }
               interface so-6/0/0.5 {
                  remote-site-id 2;
               }
            }
         }
      }
   }
}
```

Configuring CCC Encapsulation on the Interfaces

You need to specify a circuit cross-connect (CCC) encapsulation type for each PE-router-to-CE-router interface running in the Layer 2 VPN. This encapsulation type should match the encapsulation type configured under the routing instance.

Configure the following CCC encapsulation types for the interfaces on Router A:

```
[edit]
interfaces so-6/0/0 {
   encapsulation frame-relay-ccc;
   unit 0 {
      encapsulation frame-relay-ccc;
   }
}
interfaces so-6/0/0 {
   encapsulation frame-relay-ccc;
   unit 1 {
      encapsulation frame-relay-ccc;
   }
}
```

Configure the following CCC encapsulation types for the interfaces on Router B:

```
[edit]
interfaces so-6/0/0 {
   encapsulation frame-relay-ccc;
   unit 2 {
      encapsulation frame-relay-ccc;
   }
}
interfaces so-6/0/0 {
   encapsulation frame-relay-ccc;
   unit 3 {
      encapsulation frame-relay-ccc;
   }
}
```

Configure the following CCC encapsulation types for the interfaces on Router C:

```
[edit]
interface so-6/0/0 {
```

```
    encapsulation frame-relay-ccc;
    unit 4 {
      encapsulation frame-relay-ccc;
    }
  }
  interface so-6/0/0 {
    encapsulation frame-relay-ccc;
    unit 5 {
      encapsulation frame-relay-ccc;
    }
  }
```

Configuring VPN Policy on the PE Routers

You must configure VPN import and export policies on each of the PE routers so that they install the appropriate routes in their VRF tables, which the routers use to forward packets within the VPN.

NOTE: Use the **community add** *community-name* statement at the **[edit policy-options policy-statement** *policy-statement-name* **term** *term-name* **then]** hierarchy level to facilitate Layer 2 VPN VRF export policies.

On Router A, configure the following VPN import and export policies:

```
[edit]
policy-options {
  policy-statement match-all {
    term acceptable {
      then accept;
    }
  }
  policy-statement vpn-SPA-export {
    term a {
      then {
        community add SPA-com;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  policy-statement vpn-SPA-import {
    term a {
      from {
        protocol bgp;
        community SPA-com;
      }
      then accept;
    }
    term b {
      then reject;
    }
```

```
    }
    community SPA-com members target:69:100;
}
```

On Router B, configure the following VPN import and export policies:

```
[edit]
policy-options {
    policy-statement match-all {
        term acceptable {
            then accept;
        }
    }
    policy-statement vpn-SPA-import {
        term a {
            from {
                protocol bgp;
                community SPA-com;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    policy-statement vpn-SPA-export {
        term a {
            then {
                community add SPA-com;
                accept;
            }
        }
        term b {
            then reject;
        }
    }
    community SPA-com members target:69:100;
}
```

On Router C, configure the following VPN import and export policies:

```
[edit]
policy-options {
    policy-statement match-all {
        term acceptable {
            then accept;
        }
    }
    policy-statement vpn-SPA-import {
        term a {
            from {
                protocol bgp;
                community SPA-com;
            }
            then accept;
        }
        term b {
```

```
        then reject;
      }
    }
    policy-statement vpn-SPA-export {
      term a {
        then {
          community add SPA-com;
          accept;
        }
      }
      term b {
        then reject;
      }
    }
    community SPA-com members target:69:100;
}
```

To apply the VPN policies on the routers, include the **vrf-export** and **vrf-import** statements when you configure the routing instance. The VRF import and export policies handle the route distribution across the IBGP session running between the PE routers.

To apply the VPN policies on Router A, include the following statements:

```
[edit]
routing-instances {
  VPN-Sunnyvale-Portland-Austin {
    vrf-import vpn-SPA-import;
    vrf-export vpn-SPA-export;
  }
}
```

To apply the VPN policies on Router B, include the following statements:

```
[edit]
routing-instances {
  VPN-Sunnyvale-Portland-Austin {
    vrf-import vpn-SPA-import;
    vrf-export vpn-SPA-export;
  }
}
```

To apply the VPN policies on Router C, include the following statements:

```
[edit]
routing-instances {
  VPN-Sunnyvale-Portland-Austin {
    vrf-import vpn-SPA-import;
    vrf-export vpn-SPA-export;
  }
}
```


Layer 2 VPN Configuration Summarized by Router

For a summary of the configuration on each router in the examples in this chapter, see the following sections:

Summary for Router A (PE Router for Sunnyvale)

Routing Instance for Layer 2 VPN

```
[edit]
routing-instances {
   VPN-Sunnyvale-Portland-Austin {
      instance-type l2vpn;
      interface so-6/0/0.0;
      interface so-6/0/0.1;
      route-distinguisher 100:1;
      vrf-import vpn-SPA-import;
      vrf-export vpn-SPA-export;
      protocols {
         l2vpn {
            encapsulation-type frame-relay;
            site Sunnyvale {
               site-identifier 1;
               interface so-6/0/0.0 {
                  remote-site-id 2;
               }
               interface so-6/0/0.1 {
                  remote-site-id 3;
               }
            }
         }
      }
   }
}
```

Configure CCC Encapsulation Types for Interfaces

```
interfaces {
   interface so-6/0/0 {
      encapsulation frame-relay-ccc;
      unit 0 {
         encapsulation frame-relay-ccc;
      }
   }
   interface so-6/0/0 {
      encapsulation frame-relay-ccc;
      unit 1 {
         encapsulation frame-relay-ccc;
      }
   }
}
```

Master Protocol Instance

```
protocols {
}
```

Enable RSVP

```
rsvp {
  interface all;
}
```

Configure MPLS LSPs

```
mpls {
  label-switched-path RouterA-to-RouterB {
    to 192.168.37.5;
    primary Path-to-RouterB {
      cspf;
    }
  }
  label-switched-path RouterA-to-RouterC {
    to 192.168.37.10;
    primary Path-to-RouterC {
      cspf;
    }
  }
  interface all;
}
```

Configure IBGP

```
bgp {
  import match-all;
  export match-all;
  group pe-pe {
    type internal;
    neighbor 192.168.37.5 {
      local-address 192.168.37.1;
      family l2vpn {
        signaling;
      }
    }
    neighbor 192.168.37.10 {
      local-address 192.168.37.1;
      family l2vpn {
        signaling;
      }
    }
  }
}
```

Configure VPN Policy

```
policy-options {
  policy-statement match-all {
    term acceptable {
      then accept;
    }
  }
  policy-statement vpn-SPA-export {
    term a {
      then {
        community add SPA-com;
        accept;
      }
```

```
    }
    term b {
      then reject;
    }
  }
  policy-statement vpn-SPA-import {
    term a {
      from {
        protocol bgp;
        community SPA-com;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  community SPA-com members target:69:100;
}
```

Summary for Router B (PE Router for Austin)

Routing Instance for VPN

```
[edit]
routing-instances {
  VPN-Sunnyvale-Portland-Austin {
    instance-type l2vpn;
    interface so-6/0/0.2;
    interface so-6/0/0.3;
    route-distinguisher 100:1;
    vrf-import vpn-SPA-import;
    vrf-export vpn-SPA-export;
  }
}
```

Configure Layer 2 VPN

```
protocols {
  l2vpn {
    encapsulation-type frame-relay;
    site Austin {
      site-identifier 2;
      interface so-6/0/0.2 {
        remote-site-id 1;
      }
      interface so-6/0/0.3 {
        remote-site-id 3;
      }
    }
  }
}
```

Configure CCC Encapsulation Types for Interfaces

```
[edit]
interfaces {
  interface so-6/0/0 {
    encapsulation frame-relay-ccc;
    unit 2 {
      encapsulation frame-relay-ccc;
```

```
    }
  }
  interface so-6/0/0 {
    encapsulation frame-relay-ccc;
    unit 3 {
      encapsulation frame-relay-ccc;
    }
  }
}
```

Master Protocol Instance

```
protocols {
}
```

Enable RSVP

```
rsvp {
  interface all;
}
```

Configure MPLS LSPs

```
mpls {
  label-switched-path RouterB-to-RouterA {
    to 192.168.37.1;
    primary Path-to-RouterA {
      cspf;
    }
  }
  label-switched-path RouterB-to-RouterC {
    to 192.168.37.10;
    primary Path-to-RouterC {
      cspf;
    }
  }
  interface all;
}
```

Configure IBGP

```
bgp {
  local-address 192.168.37.5;
  import match-all;
  export match-all;
  group pe-pe {
    type internal;
    neighbor 192.168.37.1 {
      local-address 192.168.37.5;
      family l2vpn {
        signaling;
      }
    }
    neighbor 192.168.37.10 {
      local-address 192.168.37.5;
      family l2vpn {
        signaling;
      }
    }
  }
}
```

Configure VPN Policy

```
policy-options {
```

```
    policy-statement match-all {
      term acceptable {
        then accept;
      }
    }
    policy-statement vpn-SPA-import {
      term a {
        from {
          protocol bgp;
          community SPA-com;
        }
        then accept;
      }
      term b {
        then reject;
      }
    }
    policy-statement vpn-SPA-export {
      term a {
        then {
          community add SPA-com;
          accept;
        }
      }
      term b {
        then reject;
      }
    }
    community SPA-com members target:69:100;
}
```

Summary for Router C (PE Router for Portland)

Routing Instance for VPN

```
[edit]
routing-instances {
  VPN-Sunnyvale-Portland-Austin {
    instance-type l2vpn;
    interface so-6/0/0.3;
    interface so-6/0/0.4;
    route-distinguisher 100:1;
    vrf-import vpn-SPA-import;
    vrf-export vpn-SPA-export;
  }
}
```

Configure Layer 2 VPN

```
protocols {
  l2vpn {
    encapsulation-type frame-relay;
    site Portland {
      site-identifier 3;
      interface so-6/0/0.4 {
        remote-site-id 1;
      }
      interface so-6/0/0.5 {
        remote-site-id 2;
```

```
        }
      }
    }
}
```

Configure CCC Encapsulation Types for Interfaces

```
[edit]
interfaces {
  interface so-6/0/0 {
    encapsulation frame-relay-ccc;
    unit 4 {
      encapsulation frame-relay-ccc;
    }
  }
  interface so-6/0/0 {
    encapsulation frame-relay-ccc;
    unit 5 {
      encapsulation frame-relay-ccc;
    }
  }
}
```

Master Protocol Instance

```
protocols {
}
```

Enable RSVP

```
rsvp {
  interface all;
}
```

Configure MPLS LSPs

```
mpls {
  label-switched-path RouterC-to-RouterA {
    to 192.168.37.1;
    primary Path-to-RouterA {
      cspf;
    }
  }
  label-switched-path RouterC-to-RouterB {
    to 192.168.37.5;
    primary Path-to-RouterB {
      cspf;
    }
  }
  interface all;
}
```

Configure IBGP

```
bgp {
  local-address 192.168.37.10;
  import match-all;
  export match-all;
  group pe-pe {
    type internal;
    neighbor 192.168.37.1 {
      local-address 192.168.37.10;
      family l2vpn {
        signaling;
      }
```

```
    }
    neighbor 192.168.37.5 {
      local-address 192.168.37.10;
      family l2vpn {
        signaling;
      }
    }
  }
}
```

Configure VPN Policy

```
policy-options {
  policy-statement match-all {
    term acceptable {
      then accept;
    }
  }
  policy-statement vpn-SPA-import {
    term a {
      from {
        protocol bgp;
        community SPA-com;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement vpn-SPA-export {
    term a {
      then {
        community add SPA-com;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  community SPA-com members target:69:100;
}
```

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

This example provides a step-by-step procedure for interconnecting and verifying a Layer 2 VPN with a Layer 2 VPN. It contains the following sections:

Requirements

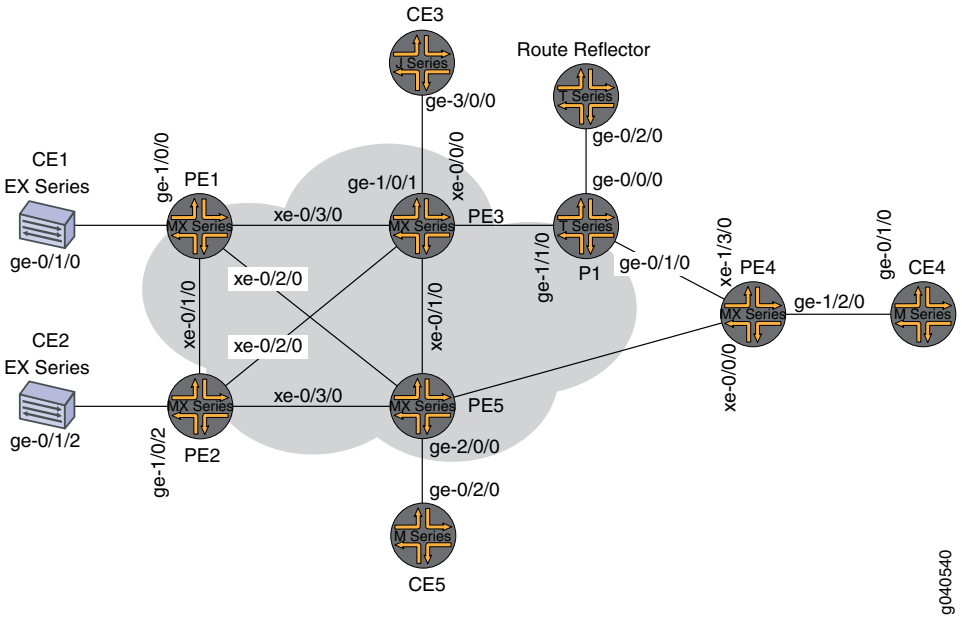
This example uses the following hardware and software components:

- Junos OS Release 9.3 or later
- 2 MX Series 3D Universal Edge Routers
- 2 M Series Multiservice Edge Routers
- 1 T Series Core Routers
- 1 EX Series Ethernet Switches
- 1 J Series Services Routers

Overview and Topology

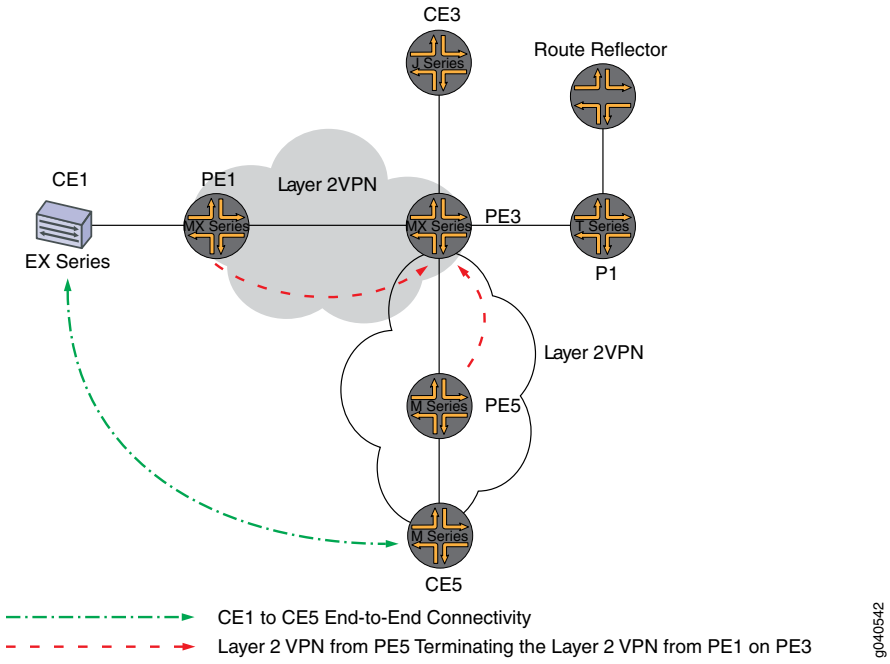
The physical topology of the Layer 2 VPN to Layer 2 VPN connection example is shown in Figure 4 on page 38.

Figure 4: Physical Topology of a Layer 2 VPN to Layer 2 VPN Connection



The logical topology of a Layer 2 VPN to Layer 2 VPN connection is shown in Figure 5 on page 39.

Figure 5: Logical Topology of a Layer 2 VPN to Layer 2 VPN Connection



Configuration

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

In this example, the router being configured is identified using the following command prompts:

- **CE1** identifies the customer edge 1 (CE1) router
- **PE1** identifies the provider edge 1 (PE1) router
- **CE3** identifies the customer edge 3 (CE3) router
- **PE3** identifies the provider edge 3 (PE3) router
- **CE5** identifies the customer edge 5 (CE5) router
- **PE5** identifies the provider edge 5 (PE5) router

This example is organized in the following sections:

Configuring Protocols on the PE and P Routers

Step-by-Step Procedure

All of the PE routers and P routers are configured with OSPF as the IGP protocol. The MPLS, LDP, and BGP protocols are enabled on all of the interfaces except **fxp.0**. Core-facing interfaces are enabled with the MPLS address and inet address.

1. Configure all the PE and P routers with OSPF as the IGP. Enable the MPLS, LDP, and BGP protocols on all interfaces except **fxp.0**. The following configuration snippet shows the protocol configuration for Router PE1:

```
[edit]
protocols {
    mpls {
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
    bgp {
        group RR {
            type internal;
            local-address 1.1.1.1;
            family l2vpn {
                signaling;
            }
            neighbor 7.7.7.7;
        }
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface all;
            interface fxp0.0 {
                disable;
            }
        }
    }
    ldp {
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
}
```

2. Configure the PE and P routers with OSPF as the IGP. Enable the MPLS, LDP, and BGP protocols on all interfaces except **fxp.0**. The following configuration snippet shows the protocol configuration for Router PE3:

```
[edit]
protocols {
    mpls {
        interface all;
        interface fxp0.0 {
            disable;
```

```
    }
  }
  bgp {
    group RR {
      type internal;
      local-address 3.3.3.3;
      family l2vpn {
        signaling;
      }
      neighbor 7.7.7.7;
    }
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface all;
      interface fxp0.0 {
        disable;
      }
    }
  }
  ldp {
    interface all;
    interface fxp0.0 {
      disable;
    }
  }
}
```

Step-by-Step Procedure

Configuring the Layer 2 VPN Protocol and Interfaces

1. On Router PE1, configure the **ge-1/0/0** interface encapsulation. To configure the interface encapsulation, include the **encapsulation** statement and specify the **ethernet-ccc** option (vlan-ccc encapsulation is also supported). Configure the **ge-1/0/0.0** logical interface family for circuit cross-connect functionality. To configure the logical interface family, include the **family** statement and specify the **ccc** option. The encapsulation should be configured the same way for all routers in the Layer 2 VPN domain.

```
[edit interfaces]
ge-1/0/0 {
  encapsulation ethernet-ccc;
  unit 0 {
    family ccc;
  }
}
lo0 {
  unit 0 {
    family inet {
      address 1.1.1.1/32;
    }
  }
}
```

2. On Router PE1, configure the Layer 2 VPN protocols. Configure the remote site ID as 3. Site ID 3 represents Router PE3 (Hub-PE). To configure the Layer 2 VPN protocols, include the **l2vpn** statement at the **[edit routing-instances *routing-instances-name* protocols]** hierarchy level. Layer 2 VPNs use BGP as the signaling protocol.

```
[edit routing-instances]
L2VPN {
   instance-type l2vpn;
   interface ge-1/0/0.0;
   route-distinguisher 65000:1;
   vrf-target target:65000:2;
   protocols {
      l2vpn {
         encapsulation-type ethernet;
         site CE1 {
            site-identifier 1;
            interface ge-1/0/0.0 {
               remote-site-id 3;
            }
         }
      }
   }
}
```

3. On Router PE5, configure the **ge-2/0/0** interface encapsulation by including the **encapsulation** statement and specify the **ethernet-ccc** option. Configure the ge-1/0/0.0 logical interface family for circuit cross-connect functionality by including the **family** statement and specifying the **ccc** option.

```
[edit interfaces]
ge-2/0/0 {
   encapsulation ethernet-ccc;
   unit 0 {
      family ccc;
   }
}
lo0 {
   unit 0 {
      family inet {
         address 5.5.5.5/32;
      }
   }
}
```

4. On Router PE5, configure the Layer 2 VPN protocols by including the **l2vpn** statement at the **[edit routing-instances *routing-instances-name* protocols]** hierarchy level. Configure the remote site ID as **3**.

```
[edit routing-instances]
L2VPN {
   instance-type l2vpn;
   interface ge-2/0/0.0;
   route-distinguisher 65000:5;
   vrf-target target:65000:2;
   protocols {
```



```
    l2vpn {
      encapsulation-type ethernet;
      site CE5 {
        site-identifier 5;
        interface ge-2/0/0.0 {
          remote-site-id 3;
        }
      }
    }
  }
}
```

5. On Router PE3, configure the **iw0** interface with two logical interfaces. To configure the **iw0** interface, include the **interfaces** statement and specify **iw0** as the interface name. For the unit 0 logical interface, include the **peer-unit** statement and specify the logical interface **unit 1** as the peer interface. For the unit 1 logical interface, include the **peer-unit** statement and specify the logical interface **unit 0** as the peer interface.

```
[edit interfaces]
iw0 {
  unit 0 {
    encapsulation ethernet-ccc;
    peer-unit 1;
  }
  unit 1 {
    encapsulation ethernet-ccc;
    peer-unit 0;
  }
}
```

6. On Router PE3, configure the edge-facing **ge-1/0/1** interface encapsulation by including the **encapsulation** statement and specifying the **ethernet-ccc** option.

```
[edit interfaces]
ge-1/0/1 {
  encapsulation ethernet-ccc;
  unit 0 {
    family ccc;
  }
}
```

7. On Router PE3, configure the logical loopback interface. The loopback interface is used to establish the targeted LDP sessions to Routers PE1 and Router PE5.

```
[edit interfaces]
lo0 {
  unit 0 {
    family inet {
      address 3.3.3.3/32;
    }
  }
}
```

8. On Router PE3, enable the Layer 2 interworking protocol. To enable the Layer 2 interworking protocol, include the **l2iw** statement at the **[edit protocols]** hierarchy level.

```
[edit protocols]
l2iw;
```

9. On Router PE3, configure two Layer 2 VPN routing instances to terminate the Layer 2 VPN virtual circuits from Router PE1 and Router PE5, as shown.

```
[edit routing-instances]
L2VPN-PE1 {
    instance-type l2vpn;
    interface iw0.0;
    route-distinguisher 65000:3;
    vrf-target target:65000:2;
    protocols {
        l2vpn {
            encapsulation-type ethernet;
            site CE3 {
                site-identifier 3;
                interface iw0.0 {
                    remote-site-id 1;
                }
            }
        }
    }
}
L2VPN-PE5 {
    instance-type l2vpn;
    interface iw0.1;
    route-distinguisher 65000:33;
    vrf-target target:65000:2;
    protocols {
        l2vpn {
            encapsulation-type ethernet;
            site CE3 {
                site-identifier 3;
                interface iw0.1 {
                    remote-site-id 5;
                }
            }
        }
    }
}
```

Verifying the Layer 2 VPN to Layer 2 VPN Connection on Router PE3

Step-by-Step Procedure

1. BGP is used for control plane signaling in a Layer 2 VPN. On Router PE1, use the **show bgp** command to verify that the BGP control plane for the Layer 2 VPN, has established a neighbor relationship with the route reflector that has IP address **7.7.7.7**.

   Three Layer 2 VPN routes are received from the route reflector for each PE router in the topology.

```
user@PE1> show bgp summary

Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l2vpn.0            3          3          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
7.7.7.7                65000        190        192       0       0     1:24:40 Establ
  bgp.l2vpn.0: 3/3/3/0
  L2VPN.l2vpn.0: 3/3/3/0
```

2. On Router PE1, use the **show route** command to verify that the BGP Layer 2 VPN routes are stored in the **L2VPN.l2vpn.0** routing table for each PE router.

```
user@PE1> show route table L2VPN.l2vpn.0

L2VPN.l2vpn.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
65000:1:1:3/96
                   *[L2VPN/170/-101] 01:31:53, metric2 1
                      Indirect
65000:3:3:1/96
                   *[BGP/170] 01:24:58, localpref 100, from 7.7.7.7
                      AS path: I
                    > to 10.10.1.2 via xe-0/3/0.0
65000:5:5:3/96
                   *[BGP/170] 01:24:58, localpref 100, from 7.7.7.7
                      AS path: I
                    > to 10.10.3.2 via xe-0/2/0.0
65000:33:3:5/96
                   *[BGP/170] 01:24:58, localpref 100, from 7.7.7.7
                      AS path: I
                    > to 10.10.1.2 via xe-0/3/0.0
```

3. On Router PE1, use the **show ldp session** command to verify that targeted LDP sessions are established to the PE routers in the network and that the state is **Operational**.

```
user@PE1> show ldp session

  Address           State       Connection     Hold time
2.2.2.2             Operational Open             24
3.3.3.3             Operational Open             22
5.5.5.5             Operational Open             28
```

4. On Router PE1, use the **show l2vpn connections** command to verify that the Layer 2 VPN to site 3 on Router PE3 (Hub-PE) is **Up**.

```
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 availble
BK -- Backup connection          ST -- Standby connection
PF -- Profile parse failure      PB -- Profile busy

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

Instance: L2VPN
  Local site: CE1 (1)
    connection-site           Type  St     Time last up          # Up trans
    3                         rmt   Up     Jan  5 18:08:25 2010           1
      Remote PE: 3.3.3.3, Negotiated control-word: Yes (Null)
      Incoming label: 800000, Outgoing label: 800000
      Local interface: ge-1/0/0.0, Status: Up, Encapsulation: ETHERNET
    5                         rmt   OR
```

5. On Router PE1, use the **show route** command to verify that the **mpls.0** routing table is populated with the Layer 2 VPN routes used to forward the traffic using an LDP label. Notice that in this example, the router is pushing label **8000000**.

```
user@PE1> show route table mpls.0

[edit]
mpls.0: 13 destinations, 13 routes (13 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 1w1d 11:36:44, metric 1
                      Receive
1                  *[MPLS/0] 1w1d 11:36:44, metric 1
                      Receive
2                  *[MPLS/0] 1w1d 11:36:44, metric 1
                      Receive
300432             *[LDP/9] 3d 04:25:02, metric 1
                    > to 10.10.2.2 via xe-0/1/0.0, Pop
300432(S=0)        *[LDP/9] 3d 04:25:02, metric 1
                    > to 10.10.2.2 via xe-0/1/0.0, Pop
300768             *[LDP/9] 3d 04:25:02, metric 1
                    > to 10.10.3.2 via xe-0/2/0.0, Pop
300768(S=0)        *[LDP/9] 3d 04:25:02, metric 1
                    > to 10.10.3.2 via xe-0/2/0.0, Pop
300912             *[LDP/9] 3d 04:25:02, metric 1
                    > to 10.10.3.2 via xe-0/2/0.0, Swap 299856
301264             *[LDP/9] 3d 04:24:58, metric 1
```

```
                    > to 10.10.1.2 via xe-0/3/0.0, Swap 308224
301312              *[LDP/9] 3d 04:25:01, metric 1
                    > to 10.10.1.2 via xe-0/3/0.0, Pop
301312(S=0)         *[LDP/9] 3d 04:25:01, metric 1
                    > to 10.10.1.2 via xe-0/3/0.0, Pop
800000      *[L2VPN/7] 01:25:28
        > via ge-1/0/0.0, Pop     Offset: 4
ge-1/0/0.0      *[L2VPN/7] 01:25:28, metric2 1
        > to 10.10.1.2 via xe-0/3/0.0, Push 800000 Offset: -4
```

Verifying the Layer 2 VPN to Layer 2 VPN Connection on Router PE3

Step-by-Step Procedure

1. On Router PE3, use the **show l2vpn connections** command to verify that the Layer 2 VPN connections from Router PE1 and Router PE5 are **Up** and are using the **iw0** interface.

```
user@PE3> show l2vpn connections

Instance: L2VPN-PE1
  Local site: CE3 (3)
    connection-site           Type  St     Time last up          # Up trans
    1                         rmt   Up     Jan  5 18:08:22 2010           1
      Remote PE: 1.1.1.1, Negotiated control-word: Yes (Null)
      Incoming label: 800000, Outgoing label: 800000
      Local interface: iw0.0, Status: Up, Encapsulation: ETHERNET
    5                         rmt   OR

Instance: L2VPN-PE5
  Local site: CE3 (3)
    connection-site           Type  St     Time last up          # Up trans
    1                         rmt   CN
    5                         rmt   Up     Jan  5 18:08:22 2010           1
      Remote PE: 5.5.5.5, Negotiated control-word: Yes (Null)
      Incoming label: 800002, Outgoing label: 800000
      Local interface: iw0.1, Status: Up, Encapsulation: ETHERNET
```

2. On Router PE3, use the **show ldp neighbor** command to verify that the targeted LDP session neighbor IP addresses are shown.

```
user@PE3> show ldp neighbor

Address            Interface          Label space ID         Hold time
1.1.1.1            lo0.0              1.1.1.1:0                 44
2.2.2.2            lo0.0              2.2.2.2:0                 42
4.4.4.4            lo0.0              4.4.4.4:0                 31
5.5.5.5            lo0.0              5.5.5.5:0                 44
```

3. On Router PE3, use the **show bgp summary** command to verify that the BGP control plane for the Layer 2 VPN, has established a neighbor relationship with the route reflector that has IP address **7.7.7.7**.

```
user@PE3> show bgp summary

Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l2vpn.0            2          2          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
7.7.7.7              65000      10092      10195       0       0  3d 4:23:27 Establ
  bgp.l2vpn.0: 2/2/2/0
```

```
L2VPN-PE1.l2vpn.0: 2/2/2/0
L2VPN-PE5.l2vpn.0: 2/2/2/0
```

4. On Router PE3, use the **show ldp session** command to verify that targeted LDP sessions are established to all of the PE routers in the network and that the state is **Operational**.

```
user@PE3> show ldp session

Address           State        Connection     Hold time
1.1.1.1             Operational  Open             24
2.2.2.2             Operational  Open             22
4.4.4.4             Operational  Open             20
5.5.5.5             Operational  Open             24
```

5. On Router PE3, use the **show route** command to verify that the **mpls.0** routing table is populated with the Layer 2 VPN routes used to forward the traffic using an LDP label. Notice that in this example, the router is swapping label **800000**. Also notice the two **iw0** interfaces that are used for the Layer 2 interworking routes.

```
user@PE3>show route table mpls.0

mpls.0: 16 destinations, 18 routes (16 active, 2 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 1w1d 11:50:14, metric 1
                      Receive
1                  *[MPLS/0] 1w1d 11:50:14, metric 1
                      Receive
2                  *[MPLS/0] 1w1d 11:50:14, metric 1
                      Receive
308160             *[LDP/9] 3d 04:38:45, metric 1
                    > to 10.10.1.1 via xe-0/3/0.0, Pop
308160(S=0)        *[LDP/9] 3d 04:38:45, metric 1
                    > to 10.10.1.1 via xe-0/3/0.0, Pop
308176             *[LDP/9] 3d 04:38:44, metric 1
                    > to 10.10.6.2 via xe-0/1/0.0, Pop
308176(S=0)        *[LDP/9] 3d 04:38:44, metric 1
                    > to 10.10.6.2 via xe-0/1/0.0, Pop
308192             *[LDP/9] 00:07:18, metric 1
                    > to 10.10.20.1 via xe-0/0/0.0, Swap 601649
                      to 10.10.6.2 via xe-0/1/0.0, Swap 299856
308208             *[LDP/9] 3d 04:38:44, metric 1
                    > to 10.10.5.1 via xe-0/2/0.0, Pop
308208(S=0)        *[LDP/9] 3d 04:38:44, metric 1
                    > to 10.10.5.1 via xe-0/2/0.0, Pop
308224             *[LDP/9] 3d 04:38:42, metric 1
                    > to 10.10.20.1 via xe-0/0/0.0, Pop
308224(S=0)        *[LDP/9] 3d 04:38:42, metric 1
                    > to 10.10.20.1 via xe-0/0/0.0, Pop
800000       *[L2IW/6] 01:39:13, metric2 1
         > to 10.10.6.2 via xe-0/1/0.0, Swap 800000
         [L2VPN/7] 01:39:13
         > via iw0.0, Pop     Offset: 4
800002       *[L2IW/6] 01:39:13, metric2 1
         > to 10.10.1.1 via xe-0/3/0.0, Swap 800000
         [L2VPN/7] 01:39:13
         > via iw0.1, Pop     Offset: 4
iw0.0        *[L2VPN/7] 01:39:13, metric2 1
         > to 10.10.1.1 via xe-0/3/0.0, Push 800000 Offset: -4
```

```
iw0.1          *[L2VPN/7] 01:39:13, metric2 1
                > to 10.10.6.2 via xe-0/1/0.0, Push 800000 Offset: -4
```

Step-by-Step Procedure

Testing Layer 2 VPN to Layer 2 VPN Connectivity (CE1 to CE5)

1. On Router CE1, use the **ping** command to test connectivity to Router CE5. Notice that the response time is in milliseconds, confirming that the ping response is returned.

```
user@CE1>ping 40.40.40.11

PING 40.40.40.11 (40.40.40.11): 56 data bytes
64 bytes from 40.40.40.11: icmp_seq=1 ttl=64 time=22.425 ms
64 bytes from 40.40.40.11: icmp_seq=2 ttl=64 time=1.299 ms
64 bytes from 40.40.40.11: icmp_seq=3 ttl=64 time=1.032 ms
64 bytes from 40.40.40.11: icmp_seq=4 ttl=64 time=1.029 ms
```

2. On Router CE5, use the **ping** command to test connectivity to Router CE1. Notice that the response time is in milliseconds, confirming that the ping response is returned.

```
user@CE5>ping 40.40.40.1

PING 40.40.40.1 (40.40.40.1): 56 data bytes
64 bytes from 40.40.40.1: icmp_seq=0 ttl=64 time=1.077 ms
64 bytes from 40.40.40.1: icmp_seq=1 ttl=64 time=0.957 ms
64 bytes from 40.40.40.1: icmp_seq=2 ttl=64 time=1.057 ms 1.017 ms
```

Results

The configuration and verification of this example have been completed. The following section is for your reference.

The relevant sample configuration for Router PE1 follows.

Router PE1

```
chassis {
    dump-on-panic;
    fpc 1 {
        pic 3 {
            tunnel-services {
                bandwidth 1g;
            }
        }
    }
    network-services ethernet;
}
    interfaces {
        xe-0/1/0 {
            unit 0 {
                family inet {
                    address 10.10.2.1/30;
                }
                family mpls;
            }
        }
        xe-0/2/0 {
            unit 0 {
                family inet {
                    address 10.10.3.1/30;
```

```
        }
        family mpls;
      }
    }
    xe-0/3/0 {
      unit 0 {
        family inet {
          address 10.10.1.1/30;
        }
        family mpls;
      }
    }
    ge-1/0/0 {
      encapsulation ethernet-ccc;
      unit 0 {
        family ccc;
      }
    }
    lo0 {
      unit 0 {
        family inet {
          address 1.1.1.1/32;
        }
      }
    }
}
routing-options {
    static {
      route 172.0.0.0/8 next-hop 172.19.59.1;
    }
    autonomous-system 65000;
}
protocols {
    mpls {
      interface all;
      interface fxp0.0 {
        disable;
      }
    }
    bgp {
      group RR {
        type internal;
        local-address 1.1.1.1;
        family l2vpn {
          signaling;
        }
        neighbor 7.7.7.7;
      }
    }
    ospf {
      traffic-engineering;
      area 0.0.0.0 {
        interface all;
        interface fxp0.0 {
          disable;
        }
```

```
        }
      }
      ldp {
        interface all;
        interface fxp0.0 {
          disable;
        }
      }
    }
    routing-instances {
      L2VPN {
        instance-type l2vpn;
        interface ge-1/0/0.0;
        route-distinguisher 65000:1;
        vrf-target target:65000:2;
        protocols {
          l2vpn {
            encapsulation-type ethernet;
            site CE1 {
              site-identifier 1;
              interface ge-1/0/0.0 {
                remote-site-id 3;
              }
            }
          }
        }
      }
    }
```

The relevant sample configuration for Router PE3 follows.

Router PE3

```
chassis {
  dump-on-panic;
  fpc 1 {
    pic 3 {
      tunnel-services {
        bandwidth 1g;
      }
    }
  }
  network-services ethernet;
}
interfaces {
  xe-0/0/0 {
    unit 0 {
      family inet {
        address 10.10.20.2/30;
      }
      family mpls;
    }
  }
  xe-0/1/0 {
    unit 0 {
      family inet {
        address 10.10.6.1/30;
      }
```

```
        family mpls;
      }
    }
    xe-0/2/0 {
      unit 0 {
        family inet {
          address 10.10.5.2/30;
        }
        family mpls;
      }
    }
    xe-0/3/0 {
      unit 0 {
        family inet {
          address 10.10.1.2/30;
        }
        family mpls;
      }
    }
    ge-1/0/1 {
      encapsulation ethernet-ccc;
      unit 0 {
        family ccc;
      }
    }
    iw0 {
      unit 0 {
        encapsulation ethernet-ccc;
        peer-unit 1;
      }
      unit 1 {
        encapsulation ethernet-ccc;
        peer-unit 0;
      }
    }
    lo0 {
      unit 0 {
        family inet {
          address 3.3.3.3/32;
        }
      }
    }
}
routing-options {
    static {
      route 172.0.0.0/8 next-hop 172.19.59.1;
    }
    autonomous-system 65000;
}
protocols {
    l2iw;
    mpls {
      interface all;
      interface fxp0.0 {
        disable;
      }
```

```
    }
    bgp {
      group RR {
        type internal;
        local-address 3.3.3.3;
        family l2vpn {
          signaling;
        }
        neighbor 7.7.7.7;
      }
    }
    ospf {
      area 0.0.0.0 {
        interface all;
        interface fxp0.0 {
          disable;
        }
      }
    }
    ldp {
      interface all;
      interface fxp0.0 {
        disable;
      }
    }
  }
  routing-instances {
    L2VPN-PE1 {
      instance-type l2vpn;
      interface iw0.0;
      route-distinguisher 65000:3;
      vrf-target target:65000:2;
      protocols {
        l2vpn {
          encapsulation-type ethernet;
          site CE3 {
            site-identifier 3;
            interface iw0.0 {
              remote-site-id 1;
            }
          }
        }
      }
    }
    L2VPN-PE5 {
      instance-type l2vpn;
      interface iw0.1;
      route-distinguisher 65000:33;
      vrf-target target:65000:2;
      protocols {
        l2vpn {
          encapsulation-type ethernet;
          site CE3 {
            site-identifier 3;
            interface iw0.1 {
              remote-site-id 5;
```

```
            }
          }
        }
      }
    }
  }
```

Related Documentation

- Layer 2 VPN Overview on page 55
- Layer 2 VPN Applications on page 56
- Using the Layer 2 Interworking Interface to Interconnect a Layer 2 VPN to a Layer 2 VPN

CHAPTER 5

Additional Examples

Layer 2 VPN Overview

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

Implementing a Layer 2 VPN on a router is similar to implementing a VPN using a Layer 2 technology, such as Asynchronous Transfer Mode (ATM). However, for a Layer 2 VPN on a router, traffic is forwarded to the router in a Layer 2 format. It is carried by Multiprotocol Label Switching (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 security and privacy of an MPLS Layer 2 VPN are equal to those of an ATM or Frame Relay VPN. The service provisioned with Layer 2 VPNs is also known as Virtual Private Wire Service (VPWS).

On a Layer 2 VPN, routing typically occurs 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 the traffic 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.

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.

Related Documentation

- Layer 2 VPN Applications on page 56
- 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
- Interconnecting Layer 2 VPNs with Layer 3 VPNs Overview on page 87
- Example: Interconnecting a Layer 2 VPN with a Layer 2 VPN on page 37
- Example: Interconnecting a Layer 2 Circuit with a Layer 2 VPN
- Example: Interconnecting a Layer 2 VPN with a Layer 3 VPN on page 88

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

- Layer 2 VPN Overview on page 55
- 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 on page 37

Example: Configuring MPLS-Based Layer 2 VPNs

You can implement an MPLS-based Layer 2 virtual private network (VPN) using Junos OS routing devices to interconnect customer sites with Layer 2 technology. Layer 2 VPNs give customers complete control of their own routing. To support an MPLS-based Layer 2 VPN, you need to add components to the configuration of the two provider edge (PE) routing devices. You do not need to change the configuration of the provider devices.

This example shows how to configure an MPLS-based Layer 2 VPN.

NOTE: You can configure both an MPLS-based Layer 2 VPN and an MPLS-based Layer 3 VPN on the same device. However, you cannot configure the same customer edge interface to support both a Layer 2 VPN and a Layer 3 VPN. The core interfaces and the loopback interfaces are configured in the same way for Layer 2 VPNs and Layer 3 VPNs.

Requirements

This example uses the following hardware and software components:

- Junos OS Release 11.1 or later if you are using EX Series switches
- Two PE routing devices

Before you configure the Layer 2 VPN components, configure the basic components for an MPLS network:

- Configure two PE routing devices. See Configuring MPLS on Provider Edge Switches Using Circuit Cross-Connect (CLI Procedure).
- Configure one or more provider devices. See Configuring MPLS on Provider Switches (CLI Procedure).

NOTE: A Layer 2 VPN requires that the PE routing devices be configured using circuit cross-connect (CCC). The provider routing devices are configured in the same way for MPLS using CCC and for IP over MPLS.

Overview and Topology

A Layer 2 VPN provides complete separation between the provider's network and the customer's network—that is, the PE devices and the CE devices do not exchange routing information. Some benefits of a Layer 2 VPN are that it is private, secure, and flexible.

This example shows how to configure Layer 2 VPN components on the local and remote PE devices. This example does not include configuring a provider device, because there are no specific Layer 2 VPN components on the provider devices.

In the basic MPLS configuration of the PE devices using a circuit cross-connect (CCC), the PE devices are configured to use an interior gateway protocol (IGP), such as OSPF or IS-IS, as the routing protocol between the MPLS devices and LDP or RSVP as the signaling protocol. Traffic engineering is enabled. A label-switched path (LSP) is configured within the **[edit protocols]** hierarchy. However, unlike the basic MPLS configuration using a CCC, you do not need to associate the LSP with the customer edge interface. When you are configuring a Layer 2 VPN, you must use BGP signaling. The BGP signaling automates the connections, so manual configuration of the association between the LSP and the customer edge interface is not required.

The following components must be added to the PE routing devices for an MPLS-based Layer 2 VPN:

- BGP group with **family l2vpn signaling**
- Routing instance using instance type **l2vpn**
- The physical layer encapsulation type (**ethernet**) must be specified on the customer edge interface and the encapsulation type must also be specified in the configuration of the routing instance.

Figure 6 on page 59 illustrates the topology of this MPLS-based Layer 2 VPN.

Figure 6: MPLS-Based Layer 2 VPN

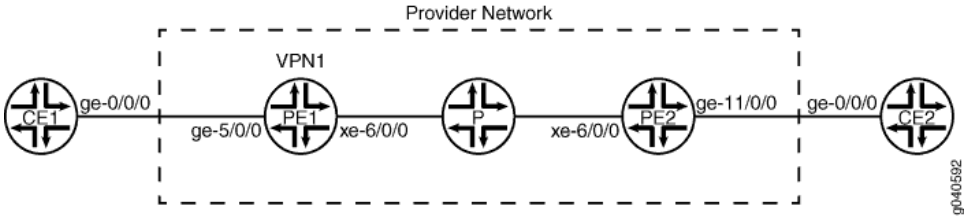


Table 3 on page 59 shows the settings of the customer edge interface on the local CE device.

Table 3: Local CE Routing Device in the MPLS-Based Layer 2 VPN Topology

Property	Settings	Description
Local CE routing device hardware	Routing device	CE1
Customer edge interface	**ge-0/0/0 unit 0** **family inet** **address 11.0.0.2/16**	Interface that connects CE1 to PE1.

Table 4 on page 59 shows the settings of the customer edge interface on the remote CE routing device.

Table 4: Remote CE Routing Device in the MPLS-Based Layer 2 VPN Topology

Property	Settings	Description
Remote CE routing device hardware	Routing device	CE2
Customer edge interface	**ge-0/0/0 unit 0** **family inet** **address 11.0.0.1/16**	Interface that connects CE2 to PE2.

Table 5 on page 60 shows the Layer 2 VPN components of the local PE routing device.

Table 5: Layer 2 VPN Components of the Local PE Routing Device

Property	Settings	Description
Local PE routing device hardware	Routing device	PE1
Customer edge interface	**ge-5/0/0** **encapsulation ethernet-ccc** **unit 0** **family ccc**	Connects PE1 to CE1. For the Layer 2 VPN, add **ethernet-ccc** as the physical layer encapsulation type. NOTE: The **family ccc** should already have been completed as part of the basic MPLS configuration of a PE routing device for circuit cross-connect. It is included here to show what was specified for that portion of the configuration.
Core interface	**xe-6/0/0 unit 0** **family inet address 60.0.0.60/16** **family iso** **family mpls**	Connects PE1 to P. NOTE: This portion of the configuration should already have been completed as part of the basic MPLS configuration. It is included here to show what was specified for that portion of the configuration.
Loopback interface	**lo0 unit 0** **family inet address 21.21.21.21/32** **family iso address** **49.0001.2102.2021.0210.00**	NOTE: This portion of the configuration should already have been completed as part of the basic MPLS configuration. It is included here to show what was specified for that portion of the configuration.
BGP	**bgp**	Added for the Layer 2 VPN configuration.
Routing instance	**vpn1**	Added for the Layer 2 VPN configuration.

Table 6 on page 61 shows the Layer 2 VPN components of the remote PE routing device.

Table 6: Layer 2 VPN Components of the Remote PE Routing Device

Property	Settings	Description
PE routing device hardware	Routing device	PE2
Customer edge interface	**ge-11/0/0** **encapsulation ethernet-ccc unit 0** **family ccc**	Connects PE2 to CE2. For the Layer 2 VPN, add **ethernet-ccc** as the physical layer encapsulation type. NOTE: The **family ccc** should already have been completed as part of the basic MPLS configuration of a PE routing device for circuit cross-connect. It is included here to show what was specified for that portion of the configuration.
Core interface	**xe-6/0/0** **unit 0** **family inet** **address 60.2.0.61/16 family iso** **family mpls**	Connects PE2 to P. NOTE: This portion of the configuration should already have been completed as part of the basic MPLS configuration. It is included here to show what was specified for that portion of the configuration.
Loopback interface	**lo0 unit 0** **family inet address 22.22.22.22/32** **family iso address** **49.0001.2202.2022.0220.00**	NOTE: This portion of the configuration should already have been completed as part of the basic MPLS configuration. It is included here to show what was specified for that portion of the configuration.
BGP	**bgp**	Added for the Layer 2 VPN configuration.
Routing instance	**vpn1**	Added for the Layer 2 VPN configuration.

Configuring the Local PE Routing Device

CLI Quick Configuration

To quickly configure the Layer 2 VPN components on the local PE routing device, copy the following commands and paste them into the routing device terminal window:

```
[edit]
set interfaces ge-5/0/0 encapsulation ethernet-ccc
set protocols bgp local-address 21.21.21.21 family l2vpn signaling
set protocols bgp group ibgp type internal
set protocols bgp neighbor 22.22.22.22
set routing-instances vpn1 instance-type l2vpn
set routing-instances vpn1 interface ge-5/0/0
set routing-instances vpn1 route-distinguisher 21.21.21.21:21
set routing-instances vpn1 vrf-target target:21:21
set routing-instances vpn1 protocols l2vpn encapsulation-type ethernet
set routing-instances vpn1 protocols l2vpn interface ge-5/0/0.0 description "BETWEEN PE1 AND PE2"
```

```
set routing-instances vpn1 protocols l2vpn site JE-V21 site-identifier 21 remote-site-id 26
```

Step-by-Step Procedure

To configure the Layer 2 VPN components on the local PE routing device:

1. Configure the customer edge interface to use the physical encapsulation type **ethernet-ccc**:

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

2. Configure BGP, specifying the loopback address as the local address and enabling **family l2vpn signaling**:

   ```
   [edit protocols bgp]
   user@PE1# set local-address 21.21.21.21 family l2vpn signaling
   ```

3. Configure the BGP group, specifying the group name and type:

   ```
   [edit protocols bgp]
   user@PE1# set group ibgp type internal
   ```

4. Configure the BGP neighbor, specifying the loopback address of the remote PE routing device as the neighbor's address:

   ```
   [edit protocols bgp]
   user@PE1# set neighbor 22.22.22.22
   ```

5. Configure the routing instance, specifying the routing-instance name and using **l2vpn** as the instance type:

   ```
   [edit routing-instances]
   user@PE1# set vpn1 instance-type l2vpn
   ```

6. Configure the routing instance to apply to the customer edge interface:

   ```
   [edit routing-instances]
   user@PE1# set vpn1 interface ge-5/0/0
   ```

7. Configure the routing instance to use a route distinguisher:

   ```
   [edit routing-instances]
   user@PE1# set vpn1 route-distinguisher 21.21.21.21:21
   ```

8. Configure the VPN routing and forwarding (VRF) target of the routing instance:

   ```
   [edit routing-instances]
   user@PE1# set vpn1 vrf-target target:21:21
   ```

   NOTE: You can create more complex policies by explicitly configuring VRF import and export policies using the import and export options. See the *Junos OS VPNs Configuration Guide*.

9. Configure the protocols and encapsulation type used by the routing instance:

   ```
   [edit routing-instances]
   user@PE1# set vpn1 protocols l2vpn encapsulation-type ethernet
   ```

10. Apply the routing instance to a customer edge interface and specify a description for it:

```
[edit routing-instances]
user@PE1# set vpn1 protocols interface ge-5/0/0.0 description "BETWEEN PE1 AND PE2"
```

11. Configure the routing-instance protocols site:

```
[edit routing-instances]
user@PE1# set vpn1 protocols l2vpn site JE-V21 site-identifier 21remote-site-id 26
```

NOTE: The remote site ID (configured with the remote-site-id statement) corresponds to the site ID (configured with the site-identifier statement) configured on the other PE routing device.

Results Display the results of the configuration:

```
user@PE1# show
  interfaces {
    ge-5/0/0 {
      encapsulation ethernet-ccc;
      unit 0 {
        family ccc;
      }
    }
    xe-6/0/0 {
      unit 0 {
        family inet {
          address 60.0.0.60/16;
        }
        family mpls;
      }
    }
    lo0 {
      unit 0 {
        family inet {
          address 21.21.21.21/32;
        }
        family iso {
          49.0001.2102.2021.0210.00;
        }
      }
    }
  protocols {
    rsvp {
      interface lo0.0;
      interface xe-0/0/6.0;
    }
    mpls {
      label-switched-path lsp_to_pe2 {
        to 22.22.22.22;
```

```
        }
        interface xe-0/0/6.0;
      }
      bgp {
        local-address 21.21.21.21;
        family l2vpn {
          signaling;
        }
        group ibgp {
          type internal;
          neighbor 22.22.22.22;
        }
      }
    routing-instances {
      vpn1 {
        instance-type l2vpn;
        interface ge-5/0/0.0;
        route-distinguisher 21.21.21.21:21;
        vrf-target target:21:21;
        protocols {
          l2vpn {
            encapsulation-type ethernet;
            interface ge-5/0/0.0 {
              description "BETWEEN PE1 AND PE2";
            }
            site JE-V21 {
            site-identifier 21;
              interface ge-5/0/0.0 {
                remote-site-id 26;
              }
          }
        }
      }
    }
```

Configuring the Remote PE Routing Device

CLI Quick Configuration

To quickly configure the Layer 2 VPN components on the remote PE routing device, copy the following commands and paste them into the routing device terminal window:

```
[edit]
set interfaces ge-11/0/0 encapsulation ethernet-ccc
set protocols bgp local-address 22.22.22.22 family l2vpn signaling
set protocols bgp group ibgp type internal
set protocols bgp neighbor 21.21.21.21
set routing-instances vpn1 instance-type l2vpn
set routing-instances vpn1 interface ge-11/0/0
set routing-instances vpn1 route-distinguisher 21.21.21.21:21
set routing-instances vpn1 vrf-target target:21:21
set routing-instances vpn1 protocols l2vpn encapsulation-type ethernet
set routing-instances vpn1 protocols l2vpn interface ge-11/0/0.0 description "BETWEEN PE1 AND
PE2"
set routing-instances vpn1 protocols l2vpn site T26-VPN1 site-identifier 26 remote-site-id 21
```

Step-by-Step Procedure

To configure the Layer 2 VPN components on the remote PE routing device:

1. Configure the customer edge interface to use the physical encapsulation type **ethernet-ccc**:

```
[edit]
user@PE1# set interfaces ge-11/0/0 encapsulation ethernet-ccc
```

2. Configure BGP, specifying the loopback address as the **local-address** and specifying **family l2vpn signaling**:

```
[edit protocols bgp]
user@PE2# set local-address 22.22.22.22 family l2vpn signaling
```

3. Configure the BGP group, specifying the group name and type:

```
[edit protocols bgp]
user@PE2# set group ibgp type internal
```

4. Configure the BGP neighbor, specifying the loopback address of the remote PE routing device as the neighbor's address:

```
[edit protocols bgp]
user@PE2# set neighbor 21.21.21.21
```

5. Configure the routing instance, specifying the routing-instance name and using **l2vpn** as the **instance-type**:

```
[edit routing-instances]
user@PE2# set vpn1 instance-type l2vpn
```

6. Configure the routing instance to apply to the customer edge interface:

```
[edit routing-instances]
user@PE2# set vpn1 interface ge-11/0/0.0
```

7. Configure the routing instance to use a route distinguisher, using the format *ip-address:number*:

```
[edit routing-instances]
user@PE2# set vpn1 route-distinguisher 21.21.21.21:21
```

8. Configure the VPN routing and forwarding (VRF) target of the routing instance:

```
[edit routing-instances]
user@PE2# set vpn1 vrf-target target:21:21
```

9. Configure the protocols and encapsulation type used by the routing instance:

```
[edit routing-instances]
user@PE2# set vpn1 protocols l2vpn encapsulation-type ethernet
```

10. Apply the routing instance to a customer edge interface and specify a description for it:

```
[edit routing-instances]
user@PE1# set vpn1 protocols interface ge-11/0/0.0 description "BETWEEN PE1 AND
PE2"
```

11. Configure the routing-instance protocols site:

```
[edit routing-instances]
user@PE2# set vpn1 protocols l2vpn site T26-VPN1 site-identifier 26 remote-site-id 21
```

NOTE: The remote site ID (configured with the **remote-site-id** statement) corresponds to the site ID (configured with the **site-identifier** statement) configured on the other PE routing device.

Results Display the results of the configuration:

```
user@PE2# show

  interfaces {
    ge-11/0/0 {
      encapsulation ethernet-ccc;
      unit 0 {
        family ccc;
      }
    }
    xe-6/0/0 {
      unit 0 {
        family inet {
          address 60.2.0.61/16;
        }
        family mpls;
      }
    }
    lo0 {
      unit 0 {
        family inet {
          address 22.22.22.22/32;
        }
        family iso {
          address 49.0001.2202.2022.0220.00;
        }
      }
    }
  protocols {
    rsvp {
      interface lo0.0;
      interface xe-0/0/6.0;
    mpls {
      label-switched-path lsp_to_pe1 {
        to 21.21.21.21;
      }
    interface xe-0/0/6.0;
    bgp {
      local-address 22.22.22.22;
      family l2vpn {
        signaling;
      }
      group ibgp {
        type internal;
        neighbor 21.21.21.21;
      }
    }
```

```
routing-instances {
   vpn1 {
      instance-type l2vpn;
      interface ge-11/0/0.0;
      route-distinguisher 21.21.21.21:21;
      vrf-target target:21:21;
      protocols {
         l2vpn {
            encapsulation-type ethernet;
            interface ge-11/0/0.0 {
               description "BETWEEN PE1 AND PE2";
            }
            site T26-VPN1 {
               site-identifier 26;
               interface ge-11/0/0.0 {
                  remote-site-id 21;
               }
            }
      }
   }
}
```

Verification

To confirm that the MPLS-based Layer 2 VPN is working properly, perform these tasks:

- Verifying the Layer 2 VPN Connection on page 67
- Verifying the Status of MPLS Label-Switched Paths on page 68
- Verifying BGP Status on page 68
- Verifying the Status of the RSVP Sessions on page 69
- Verifying the Routes in the Routing Table on page 69
- Pinging the Layer 2 VPN Connections on page 70

Verifying the Layer 2 VPN Connection

Purpose Verify that the Layer 2 VPN connection is up.

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 availble
BK -- Backup connection           ST -- Standby connection
PF -- Profile parse failure       PB -- Profile busy
RS -- remote site standby         SN -- Static Neighbor

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

Instance: vpn1
  Local site: JE-V21 (21)
    connection-site           Type  St     Time last up          # Up trans
    26                        rmt   Up     Apr 16 05:53:21 2010           1
      Remote PE: 22.22.22.22, Negotiated control-word: Yes (Null)
      Incoming label: 800000, Outgoing label: 800001
      Local interface: ge-5/0/0.0, Status: Up, Encapsulation: ETHERNET
```

Meaning The **St** field in the output shows that the Layer 2 VPN connection to **Remote PE** (**22.22.22.22**) is up.

Verifying the Status of MPLS Label-Switched Paths

Purpose Verify that the MPLS label-switched paths (ingress and egress) are up.

Action

```
user@PE1> show mpls lsp
Ingress LSP: 1 sessions

To              From            State Rt P     ActivePath       LSPname
22.22.22.22     21.21.21.21     Up     0 *                      lsp_to_pe2

Total 1 displayed, Up 1, Down 0


Egress LSP: 1 sessions

To              From            State   Rt Style Labelin Labelout LSPname

21.21.21.21     22.22.22.22     Up       0  1 FF       3        - lsp_to_pe1

Total 1 displayed, Up 1, Down 0


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

Meaning The **State** field in the output shows that the Ingress LSP to **Remote PE** (**22.22.22.22**) is up, and the Egress LSP from the remote PE routing device to this PE routing device (**21.21.21.21**) is also up.

Verifying BGP Status

Purpose Verify that BGP is up.

Action

```
user@PE1> show bgp summary
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...
22.22.22.22              10         33         34       0       1       13:24
Establ
  bgp.l2vpn.0: 1/1/1/0
  vpn2.l2vpn.0: 1/1/1/0
```

Meaning The output shows that the remote PE routing device (**22.22.22.22**) is listed as the BGP peer and that a protocol session has been established. It also shows the number of packets received from the remote PE routing device (**33**) and the number of packets sent (**34**) to the remote PE routing device.

Verifying the Status of the RSVP Sessions

Purpose Verify that the RSVP sessions (ingress and egress) are up.

Action

```
user@PE1> show rsvp session
Ingress RSVP: 1 sessions
To              From            State   Rt Style Labelin Labelout LSPname
22.22.22.22     21.21.21.21     Up       0  1 FF       -   462880 lsp_to_pe2
Total 1 displayed, Up 1, Down 0


Egress RSVP: 1 sessions
To              From            State   Rt Style Labelin Labelout LSPname
21.21.21.21     22.22.22.22     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 RSVP session and the egress RSVP session are up.

Verifying the Routes in the Routing Table

Purpose On routing device PE 1, use the **show route table** command to verify that the routing table is populated with the Layer 2 VPN routes used to forward the traffic.

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


2:2:27:27/96
                   *[BGP/170] 00:13:55, localpref 100, from 22.22.22.22
                      AS path: I
                    > to 60.2.0.24 via ge-6/0/46.0, label-switched-path lsp_to_pe2
```

```
user@PE1> show route table vpn1.l2vpn.0

vpn1.l2vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both


2:2:27:27/96

                   *[BGP/170] 00:14:00, localpref 100, from 22.22.22.22
                      AS path: I

                   > to 60.2.0.24 via ge-6/0/46.0, label-switched-path lsp_to_pe2
2:2:28:27/96

                   *[L2VPN/170/-101] 00:15:55, metric2 1
                      Indirect
```

Meaning The command **show route table bgp.l2vpn.0** displays all Layer 2 VPN routes that have been created on this routing device. The command **show route table vpn1.l2vpn.0** shows the Layer 2 VPN routes that have been created for the routing instance **vpn1**.

Pinging the Layer 2 VPN Connections

Purpose Verify connectivity.

Action

```
user@PE1> ping mpls l2vpn interface xe-6/0/0.0 reply-mode ip-udp
!!!!!
--- lsping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss

user@PE1> ping mpls l2vpn instance vpn1 remote-site-id 26 local-site-id 21 detail
Request for seq 1, to interface 68, labels <800001, 100176>
Reply for seq 1, return code: Egress-ok
Request for seq 2, to interface 68, labels <800001, 100176>
Reply for seq 2, return code: Egress-ok
Request for seq 3, to interface 68, labels <800001, 100176>
Reply for seq 3, return code: Egress-ok
Request for seq 4, to interface 68, labels <800001, 100176>
Reply for seq 4, return code: Egress-ok
Request for seq 5, to interface 68, labels <800001, 100176>
Reply for seq 5, return code: Egress-ok

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

Meaning The output shows that connectivity is established.

Related Documentation

- Example: Configuring MPLS on EX Series Switches
- Example: Configuring MPLS-Based Layer 3 VPNs on EX Series Switches
- Configuring an MPLS-Based Layer 2 VPN (CLI Procedure)

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

This example provides a step-by-step procedure for interconnecting and verifying a Layer 2 VPN with a Layer 2 VPN. It contains the following sections:

Requirements

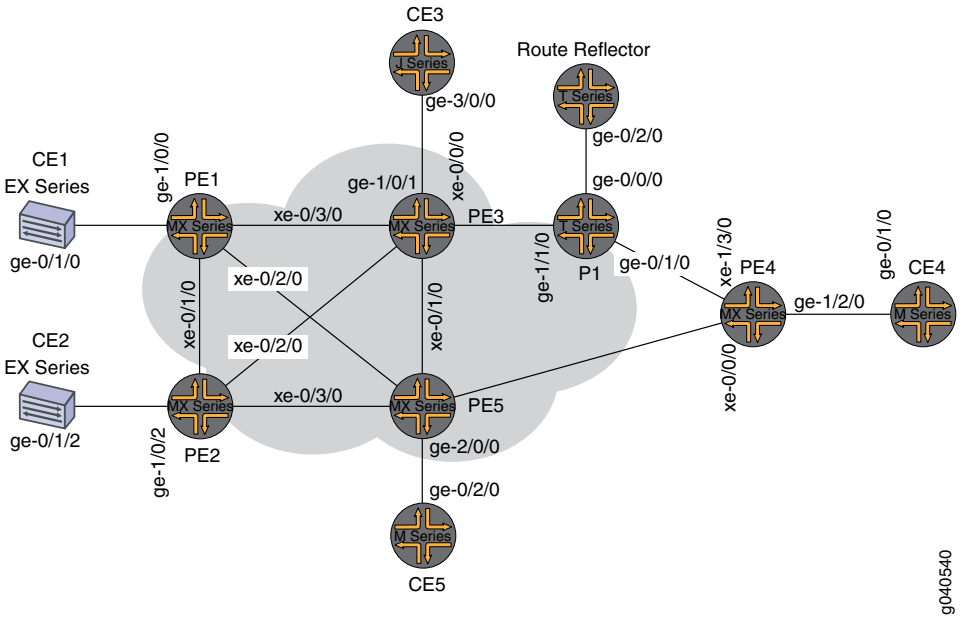
This example uses the following hardware and software components:

- Junos OS Release 9.3 or later
- 2 MX Series 3D Universal Edge Routers
- 2 M Series Multiservice Edge Routers
- 1 T Series Core Routers
- 1 EX Series Ethernet Switches
- 1 J Series Services Routers

Overview and Topology

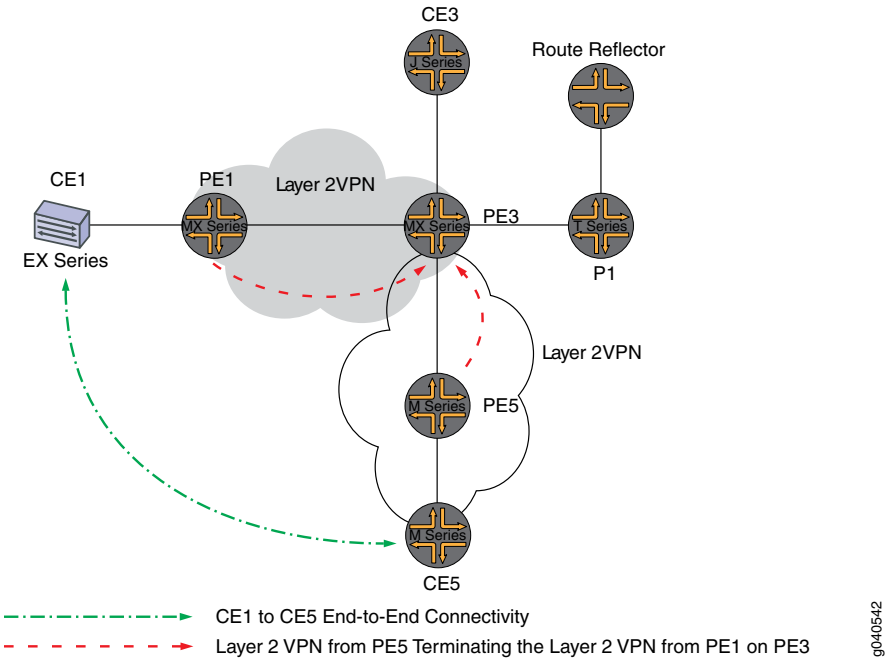
The physical topology of the Layer 2 VPN to Layer 2 VPN connection example is shown in Figure 4 on page 38.

Figure 7: Physical Topology of a Layer 2 VPN to Layer 2 VPN Connection



The logical topology of a Layer 2 VPN to Layer 2 VPN connection is shown in Figure 5 on page 39.

Figure 8: Logical Topology of a Layer 2 VPN to Layer 2 VPN Connection



Configuration

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

In this example, the router being configured is identified using the following command prompts:

- **CE1** identifies the customer edge 1 (CE1) router
- **PE1** identifies the provider edge 1 (PE1) router
- **CE3** identifies the customer edge 3 (CE3) router
- **PE3** identifies the provider edge 3 (PE3) router
- **CE5** identifies the customer edge 5 (CE5) router
- **PE5** identifies the provider edge 5 (PE5) router

This example is organized in the following sections:

- Configuring Protocols on the PE and P Routers on page 73
- Verifying the Layer 2 VPN to Layer 2 VPN Connection on Router PE3 on page 78
- Verifying the Layer 2 VPN to Layer 2 VPN Connection on Router PE3 on page 80

Configuring Protocols on the PE and P Routers

Step-by-Step Procedure

All of the PE routers and P routers are configured with OSPF as the IGP protocol. The MPLS, LDP, and BGP protocols are enabled on all of the interfaces except **fxp.0**. Core-facing interfaces are enabled with the MPLS address and inet address.

1. Configure all the PE and P routers with OSPF as the IGP. Enable the MPLS, LDP, and BGP protocols on all interfaces except **fxp.0**. The following configuration snippet shows the protocol configuration for Router PE1:

```
[edit]
protocols {
    mpls {
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
    bgp {
        group RR {
            type internal;
            local-address 1.1.1.1;
            family l2vpn {
                signaling;
            }
            neighbor 7.7.7.7;
        }
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface all;
            interface fxp0.0 {
                disable;
            }
        }
    }
    ldp {
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
}
```

2. Configure the PE and P routers with OSPF as the IGP. Enable the MPLS, LDP, and BGP protocols on all interfaces except **fxp.0**. The following configuration snippet shows the protocol configuration for Router PE3:

```
[edit]
protocols {
    mpls {
        interface all;
        interface fxp0.0 {
            disable;
```

```
    }
  }
  bgp {
    group RR {
      type internal;
      local-address 3.3.3.3;
      family l2vpn {
        signaling;
      }
      neighbor 7.7.7.7;
    }
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface all;
      interface fxp0.0 {
        disable;
      }
    }
  }
  ldp {
    interface all;
    interface fxp0.0 {
      disable;
    }
  }
}
```

Step-by-Step Procedure

Configuring the Layer 2 VPN Protocol and Interfaces

1. On Router PE1, configure the **ge-1/0/0** interface encapsulation. To configure the interface encapsulation, include the **encapsulation** statement and specify the **ethernet-ccc** option (vlan-ccc encapsulation is also supported). Configure the **ge-1/0/0.0** logical interface family for circuit cross-connect functionality. To configure the logical interface family, include the **family** statement and specify the **ccc** option. The encapsulation should be configured the same way for all routers in the Layer 2 VPN domain.

```
[edit interfaces]
ge-1/0/0 {
  encapsulation ethernet-ccc;
  unit 0 {
    family ccc;
  }
}
lo0 {
  unit 0 {
    family inet {
      address 1.1.1.1/32;
    }
  }
}
```

2. On Router PE1, configure the Layer 2 VPN protocols. Configure the remote site ID as 3. Site ID 3 represents Router PE3 (Hub-PE). To configure the Layer 2 VPN protocols, include the **l2vpn** statement at the **[edit routing-instances *routing-instances-name* protocols]** hierarchy level. Layer 2 VPNs use BGP as the signaling protocol.

```
[edit routing-instances]
L2VPN {
   instance-type l2vpn;
   interface ge-1/0/0.0;
   route-distinguisher 65000:1;
   vrf-target target:65000:2;
   protocols {
      l2vpn {
         encapsulation-type ethernet;
         site CE1 {
            site-identifier 1;
            interface ge-1/0/0.0 {
               remote-site-id 3;
            }
         }
      }
   }
}
```

3. On Router PE5, configure the **ge-2/0/0** interface encapsulation by including the **encapsulation** statement and specify the **ethernet-ccc** option. Configure the ge-1/0/0.0 logical interface family for circuit cross-connect functionality by including the **family** statement and specifying the **ccc** option.

```
[edit interfaces]
ge-2/0/0 {
   encapsulation ethernet-ccc;
   unit 0 {
      family ccc;
   }
}
lo0 {
   unit 0 {
      family inet {
         address 5.5.5.5/32;
      }
   }
}
```

4. On Router PE5, configure the Layer 2 VPN protocols by including the **l2vpn** statement at the **[edit routing-instances *routing-instances-name* protocols]** hierarchy level. Configure the remote site ID as **3**.

```
[edit routing-instances]
L2VPN {
   instance-type l2vpn;
   interface ge-2/0/0.0;
   route-distinguisher 65000:5;
   vrf-target target:65000:2;
   protocols {
```

```
    l2vpn {
      encapsulation-type ethernet;
      site CE5 {
        site-identifier 5;
        interface ge-2/0/0.0 {
          remote-site-id 3;
        }
      }
    }
  }
}
```

5. On Router PE3, configure the **iw0** interface with two logical interfaces. To configure the **iw0** interface, include the **interfaces** statement and specify **iw0** as the interface name. For the unit 0 logical interface, include the **peer-unit** statement and specify the logical interface **unit 1** as the peer interface. For the unit 1 logical interface, include the **peer-unit** statement and specify the logical interface **unit 0** as the peer interface.

```
[edit interfaces]
iw0 {
  unit 0 {
    encapsulation ethernet-ccc;
    peer-unit 1;
  }
  unit 1 {
    encapsulation ethernet-ccc;
    peer-unit 0;
  }
}
```

6. On Router PE3, configure the edge-facing **ge-1/0/1** interface encapsulation by including the **encapsulation** statement and specifying the **ethernet-ccc** option.

```
[edit interfaces]
ge-1/0/1 {
  encapsulation ethernet-ccc;
  unit 0 {
    family ccc;
  }
}
```

7. On Router PE3, configure the logical loopback interface. The loopback interface is used to establish the targeted LDP sessions to Routers PE1 and Router PE5.

```
[edit interfaces]
lo0 {
  unit 0 {
    family inet {
      address 3.3.3.3/32;
    }
  }
}
```

8. On Router PE3, enable the Layer 2 interworking protocol. To enable the Layer 2 interworking protocol, include the **l2iw** statement at the **[edit protocols]** hierarchy level.

```
[edit protocols]
l2iw;
```

9. On Router PE3, configure two Layer 2 VPN routing instances to terminate the Layer 2 VPN virtual circuits from Router PE1 and Router PE5, as shown.

```
[edit routing-instances]
L2VPN-PE1 {
    instance-type l2vpn;
    interface iw0.0;
    route-distinguisher 65000:3;
    vrf-target target:65000:2;
    protocols {
        l2vpn {
            encapsulation-type ethernet;
            site CE3 {
                site-identifier 3;
                interface iw0.0 {
                    remote-site-id 1;
                }
            }
        }
    }
}
L2VPN-PE5 {
    instance-type l2vpn;
    interface iw0.1;
    route-distinguisher 65000:33;
    vrf-target target:65000:2;
    protocols {
        l2vpn {
            encapsulation-type ethernet;
            site CE3 {
                site-identifier 3;
                interface iw0.1 {
                    remote-site-id 5;
                }
            }
        }
    }
}
```

Verifying the Layer 2 VPN to Layer 2 VPN Connection on Router PE3

Step-by-Step Procedure

1. BGP is used for control plane signaling in a Layer 2 VPN. On Router PE1, use the **show bgp** command to verify that the BGP control plane for the Layer 2 VPN, has established a neighbor relationship with the route reflector that has IP address **7.7.7.7**.

   Three Layer 2 VPN routes are received from the route reflector for each PE router in the topology.

```
user@PE1> show bgp summary

Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l2vpn.0            3          3          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
7.7.7.7                65000        190        192       0       0     1:24:40 Establ
  bgp.l2vpn.0: 3/3/3/0
  L2VPN.l2vpn.0: 3/3/3/0
```

2. On Router PE1, use the **show route** command to verify that the BGP Layer 2 VPN routes are stored in the **L2VPN.l2vpn.0** routing table for each PE router.

```
user@PE1> show route table L2VPN.l2vpn.0

L2VPN.l2vpn.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
65000:1:1:3/96
                   *[L2VPN/170/-101] 01:31:53, metric2 1
                      Indirect
65000:3:3:1/96
                   *[BGP/170] 01:24:58, localpref 100, from 7.7.7.7
                      AS path: I
                    > to 10.10.1.2 via xe-0/3/0.0
65000:5:5:3/96
                   *[BGP/170] 01:24:58, localpref 100, from 7.7.7.7
                      AS path: I
                    > to 10.10.3.2 via xe-0/2/0.0
65000:33:3:5/96
                   *[BGP/170] 01:24:58, localpref 100, from 7.7.7.7
                      AS path: I
                    > to 10.10.1.2 via xe-0/3/0.0
```

3. On Router PE1, use the **show ldp session** command to verify that targeted LDP sessions are established to the PE routers in the network and that the state is **Operational**.

```
user@PE1> show ldp session

  Address           State       Connection     Hold time
2.2.2.2             Operational Open             24
3.3.3.3             Operational Open             22
5.5.5.5             Operational Open             28
```

4. On Router PE1, use the **show l2vpn connections** command to verify that the Layer 2 VPN to site 3 on Router PE3 (Hub-PE) is **Up**.

```
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 availble
BK -- Backup connection          ST -- Standby connection
PF -- Profile parse failure      PB -- Profile busy

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

Instance: L2VPN
  Local site: CE1 (1)
    connection-site           Type  St     Time last up          # Up trans
    3                         rmt   Up     Jan  5 18:08:25 2010           1
      Remote PE: 3.3.3.3, Negotiated control-word: Yes (Null)
      Incoming label: 800000, Outgoing label: 800000
      Local interface: ge-1/0/0.0, Status: Up, Encapsulation: ETHERNET
    5                         rmt   OR
```

5. On Router PE1, use the **show route** command to verify that the **mpls.0** routing table is populated with the Layer 2 VPN routes used to forward the traffic using an LDP label. Notice that in this example, the router is pushing label **8000000**.

```
user@PE1> show route table mpls.0

[edit]
mpls.0: 13 destinations, 13 routes (13 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 1w1d 11:36:44, metric 1
                      Receive
1                  *[MPLS/0] 1w1d 11:36:44, metric 1
                      Receive
2                  *[MPLS/0] 1w1d 11:36:44, metric 1
                      Receive
300432             *[LDP/9] 3d 04:25:02, metric 1
                    > to 10.10.2.2 via xe-0/1/0.0, Pop
300432(S=0)        *[LDP/9] 3d 04:25:02, metric 1
                    > to 10.10.2.2 via xe-0/1/0.0, Pop
300768             *[LDP/9] 3d 04:25:02, metric 1
                    > to 10.10.3.2 via xe-0/2/0.0, Pop
300768(S=0)        *[LDP/9] 3d 04:25:02, metric 1
                    > to 10.10.3.2 via xe-0/2/0.0, Pop
300912             *[LDP/9] 3d 04:25:02, metric 1
                    > to 10.10.3.2 via xe-0/2/0.0, Swap 299856
301264             *[LDP/9] 3d 04:24:58, metric 1
```

```
                    > to 10.10.1.2 via xe-0/3/0.0, Swap 308224
301312              *[LDP/9] 3d 04:25:01, metric 1
                    > to 10.10.1.2 via xe-0/3/0.0, Pop
301312(S=0)         *[LDP/9] 3d 04:25:01, metric 1
                    > to 10.10.1.2 via xe-0/3/0.0, Pop
800000      *[L2VPN/7] 01:25:28
        > via ge-1/0/0.0, Pop     Offset: 4
ge-1/0/0.0      *[L2VPN/7] 01:25:28, metric2 1
        > to 10.10.1.2 via xe-0/3/0.0, Push 800000 Offset: -4
```

Verifying the Layer 2 VPN to Layer 2 VPN Connection on Router PE3

Step-by-Step Procedure

1. On Router PE3, use the **show l2vpn connections** command to verify that the Layer 2 VPN connections from Router PE1 and Router PE5 are **Up** and are using the **iw0** interface.

```
user@PE3> show l2vpn connections

Instance: L2VPN-PE1
  Local site: CE3 (3)
    connection-site           Type  St     Time last up          # Up trans
    1                         rmt   Up     Jan  5 18:08:22 2010           1
      Remote PE: 1.1.1.1, Negotiated control-word: Yes (Null)
      Incoming label: 800000, Outgoing label: 800000
      Local interface: iw0.0, Status: Up, Encapsulation: ETHERNET
    5                         rmt   OR

Instance: L2VPN-PE5
  Local site: CE3 (3)
    connection-site           Type  St     Time last up          # Up trans
    1                         rmt   CN
    5                         rmt   Up     Jan  5 18:08:22 2010           1
      Remote PE: 5.5.5.5, Negotiated control-word: Yes (Null)
      Incoming label: 800002, Outgoing label: 800000
      Local interface: iw0.1, Status: Up, Encapsulation: ETHERNET
```

2. On Router PE3, use the **show ldp neighbor** command to verify that the targeted LDP session neighbor IP addresses are shown.

```
user@PE3> show ldp neighbor

Address            Interface          Label space ID         Hold time
1.1.1.1            lo0.0              1.1.1.1:0                 44
2.2.2.2            lo0.0              2.2.2.2:0                 42
4.4.4.4            lo0.0              4.4.4.4:0                 31
5.5.5.5            lo0.0              5.5.5.5:0                 44
```

3. On Router PE3, use the **show bgp summary** command to verify that the BGP control plane for the Layer 2 VPN, has established a neighbor relationship with the route reflector that has IP address **7.7.7.7**.

```
user@PE3> show bgp summary

Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l2vpn.0            2          2          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
7.7.7.7              65000      10092      10195       0       0  3d 4:23:27 Establ
  bgp.l2vpn.0: 2/2/2/0
```

```
L2VPN-PE1.l2vpn.0: 2/2/2/0
L2VPN-PE5.l2vpn.0: 2/2/2/0
```

4. On Router PE3, use the **show ldp session** command to verify that targeted LDP sessions are established to all of the PE routers in the network and that the state is **Operational**.

```
user@PE3> show ldp session

Address           State        Connection     Hold time
1.1.1.1             Operational  Open             24
2.2.2.2             Operational  Open             22
4.4.4.4             Operational  Open             20
5.5.5.5             Operational  Open             24
```

5. On Router PE3, use the **show route** command to verify that the **mpls.0** routing table is populated with the Layer 2 VPN routes used to forward the traffic using an LDP label. Notice that in this example, the router is swapping label **800000**. Also notice the two **iw0** interfaces that are used for the Layer 2 interworking routes.

```
user@PE3>show route table mpls.0

mpls.0: 16 destinations, 18 routes (16 active, 2 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 1w1d 11:50:14, metric 1
                      Receive
1                  *[MPLS/0] 1w1d 11:50:14, metric 1
                      Receive
2                  *[MPLS/0] 1w1d 11:50:14, metric 1
                      Receive
308160             *[LDP/9] 3d 04:38:45, metric 1
                    > to 10.10.1.1 via xe-0/3/0.0, Pop
308160(S=0)        *[LDP/9] 3d 04:38:45, metric 1
                    > to 10.10.1.1 via xe-0/3/0.0, Pop
308176             *[LDP/9] 3d 04:38:44, metric 1
                    > to 10.10.6.2 via xe-0/1/0.0, Pop
308176(S=0)        *[LDP/9] 3d 04:38:44, metric 1
                    > to 10.10.6.2 via xe-0/1/0.0, Pop
308192             *[LDP/9] 00:07:18, metric 1
                    > to 10.10.20.1 via xe-0/0/0.0, Swap 601649
                      to 10.10.6.2 via xe-0/1/0.0, Swap 299856
308208             *[LDP/9] 3d 04:38:44, metric 1
                    > to 10.10.5.1 via xe-0/2/0.0, Pop
308208(S=0)        *[LDP/9] 3d 04:38:44, metric 1
                    > to 10.10.5.1 via xe-0/2/0.0, Pop
308224             *[LDP/9] 3d 04:38:42, metric 1
                    > to 10.10.20.1 via xe-0/0/0.0, Pop
308224(S=0)        *[LDP/9] 3d 04:38:42, metric 1
                    > to 10.10.20.1 via xe-0/0/0.0, Pop
800000       *[L2IW/6] 01:39:13, metric2 1
         > to 10.10.6.2 via xe-0/1/0.0, Swap 800000
         [L2VPN/7] 01:39:13
         > via iw0.0, Pop     Offset: 4
800002       *[L2IW/6] 01:39:13, metric2 1
         > to 10.10.1.1 via xe-0/3/0.0, Swap 800000
         [L2VPN/7] 01:39:13
         > via iw0.1, Pop     Offset: 4
iw0.0        *[L2VPN/7] 01:39:13, metric2 1
         > to 10.10.1.1 via xe-0/3/0.0, Push 800000 Offset: -4
```

```
iw0.1          *[L2VPN/7] 01:39:13, metric2 1
                > to 10.10.6.2 via xe-0/1/0.0, Push 800000 Offset: -4
```

Step-by-Step Procedure

Testing Layer 2 VPN to Layer 2 VPN Connectivity (CE1 to CE5)

1. On Router CE1, use the **ping** command to test connectivity to Router CE5. Notice that the response time is in milliseconds, confirming that the ping response is returned.

```
user@CE1>ping 40.40.40.11

PING 40.40.40.11 (40.40.40.11): 56 data bytes
64 bytes from 40.40.40.11: icmp_seq=1 ttl=64 time=22.425 ms
64 bytes from 40.40.40.11: icmp_seq=2 ttl=64 time=1.299 ms
64 bytes from 40.40.40.11: icmp_seq=3 ttl=64 time=1.032 ms
64 bytes from 40.40.40.11: icmp_seq=4 ttl=64 time=1.029 ms
```

2. On Router CE5, use the **ping** command to test connectivity to Router CE1. Notice that the response time is in milliseconds, confirming that the ping response is returned.

```
user@CE5>ping 40.40.40.1

PING 40.40.40.1 (40.40.40.1): 56 data bytes
64 bytes from 40.40.40.1: icmp_seq=0 ttl=64 time=1.077 ms
64 bytes from 40.40.40.1: icmp_seq=1 ttl=64 time=0.957 ms
64 bytes from 40.40.40.1: icmp_seq=2 ttl=64 time=1.057 ms 1.017 ms
```

Results

The configuration and verification of this example have been completed. The following section is for your reference.

The relevant sample configuration for Router PE1 follows.

Router PE1

```
chassis {
    dump-on-panic;
    fpc 1 {
        pic 3 {
            tunnel-services {
                bandwidth 1g;
            }
        }
    }
    network-services ethernet;
}
    interfaces {
        xe-0/1/0 {
            unit 0 {
                family inet {
                    address 10.10.2.1/30;
                }
                family mpls;
            }
        }
        xe-0/2/0 {
            unit 0 {
                family inet {
                    address 10.10.3.1/30;
```

```
        }
        family mpls;
      }
    }
    xe-0/3/0 {
      unit 0 {
        family inet {
          address 10.10.1.1/30;
        }
        family mpls;
      }
    }
    ge-1/0/0 {
      encapsulation ethernet-ccc;
      unit 0 {
        family ccc;
      }
    }
    lo0 {
      unit 0 {
        family inet {
          address 1.1.1.1/32;
        }
      }
    }
}
routing-options {
    static {
      route 172.0.0.0/8 next-hop 172.19.59.1;
    }
    autonomous-system 65000;
}
protocols {
    mpls {
      interface all;
      interface fxp0.0 {
        disable;
      }
    }
    bgp {
      group RR {
        type internal;
        local-address 1.1.1.1;
        family l2vpn {
          signaling;
        }
        neighbor 7.7.7.7;
      }
    }
    ospf {
      traffic-engineering;
      area 0.0.0.0 {
        interface all;
        interface fxp0.0 {
          disable;
        }
```

```
        }
      }
      ldp {
        interface all;
        interface fxp0.0 {
          disable;
        }
      }
    }
    routing-instances {
      L2VPN {
        instance-type l2vpn;
        interface ge-1/0/0.0;
        route-distinguisher 65000:1;
        vrf-target target:65000:2;
        protocols {
          l2vpn {
            encapsulation-type ethernet;
            site CE1 {
              site-identifier 1;
              interface ge-1/0/0.0 {
                remote-site-id 3;
              }
            }
          }
        }
      }
    }
```

The relevant sample configuration for Router PE3 follows.

Router PE3

```
chassis {
  dump-on-panic;
  fpc 1 {
    pic 3 {
      tunnel-services {
        bandwidth 1g;
      }
    }
  }
  network-services ethernet;
}
interfaces {
  xe-0/0/0 {
    unit 0 {
      family inet {
        address 10.10.20.2/30;
      }
      family mpls;
    }
  }
  xe-0/1/0 {
    unit 0 {
      family inet {
        address 10.10.6.1/30;
      }
```

```
        family mpls;
      }
    }
    xe-0/2/0 {
      unit 0 {
        family inet {
          address 10.10.5.2/30;
        }
        family mpls;
      }
    }
    xe-0/3/0 {
      unit 0 {
        family inet {
          address 10.10.1.2/30;
        }
        family mpls;
      }
    }
    ge-1/0/1 {
      encapsulation ethernet-ccc;
      unit 0 {
        family ccc;
      }
    }
    iw0 {
      unit 0 {
        encapsulation ethernet-ccc;
        peer-unit 1;
      }
      unit 1 {
        encapsulation ethernet-ccc;
        peer-unit 0;
      }
    }
    lo0 {
      unit 0 {
        family inet {
          address 3.3.3.3/32;
        }
      }
    }
}
routing-options {
    static {
      route 172.0.0.0/8 next-hop 172.19.59.1;
    }
    autonomous-system 65000;
}
protocols {
    l2iw;
    mpls {
      interface all;
      interface fxp0.0 {
        disable;
      }
```

```
    }
    bgp {
      group RR {
        type internal;
        local-address 3.3.3.3;
        family l2vpn {
          signaling;
        }
        neighbor 7.7.7.7;
      }
    }
    ospf {
      area 0.0.0.0 {
        interface all;
        interface fxp0.0 {
          disable;
        }
      }
    }
    ldp {
      interface all;
      interface fxp0.0 {
        disable;
      }
    }
  }
  routing-instances {
    L2VPN-PE1 {
      instance-type l2vpn;
      interface iw0.0;
      route-distinguisher 65000:3;
      vrf-target target:65000:2;
      protocols {
        l2vpn {
          encapsulation-type ethernet;
          site CE3 {
            site-identifier 3;
            interface iw0.0 {
              remote-site-id 1;
            }
          }
        }
      }
    }
    L2VPN-PE5 {
      instance-type l2vpn;
      interface iw0.1;
      route-distinguisher 65000:33;
      vrf-target target:65000:2;
      protocols {
        l2vpn {
          encapsulation-type ethernet;
          site CE3 {
            site-identifier 3;
            interface iw0.1 {
              remote-site-id 5;
```

```
            }
          }
        }
      }
    }
  }
```

Related Documentation

- Layer 2 VPN Overview on page 55
- Layer 2 VPN Applications on page 56
- Using the Layer 2 Interworking Interface to Interconnect a Layer 2 VPN to a Layer 2 VPN

Interconnecting Layer 2 VPNs with Layer 3 VPNs Overview

As MPLS-based Layer 2 services grow in demand, new challenges arise for service providers to be able to interoperate with Layer 2 and Layer 3 services and give their customers value-added services. Junos OS has various features to address the needs of service providers. One of these features is the use of a logical tunnel interface. This Junos OS functionality makes use of a tunnel PIC to loop packets out and back from the Packet Forwarding Engine to link the Layer 2 network with the Layer 3 network. The solution is limited by the logical tunnel bandwidth constraints imposed by the tunnel PIC.

Interconnecting Layer 2 VPNs with Layer 3 VPNs Applications

Interconnecting a Layer 2 VPN with a Layer 3 VPN provides the following benefits:

- A single access line to provide multiple services—Traditional VPNs over Layer 2 circuits require the provisioning and maintenance of separate networks for IP and for VPN services. In contrast, Layer 2 VPNs enable the sharing of a provider's core network infrastructure between IP and Layer 2 VPN services, thereby reducing the cost of providing those services.
- Flexibility—Many different types of networks can be accommodated by the service provider. If all sites in a VPN are owned by the same enterprise, this is an intranet. If various sites are owned by different enterprises, the VPN is an extranet. A site can be located in more than one VPN.
- Wide range of possible policies—You can give every site in a VPN a different route to every other site, or you can force traffic between certain pairs of sites routed via a third site and so pass certain traffic through a firewall.
- Scalable network—This design enhances the scalability because it eliminates the need for provider edge (PE) routers to maintain all of the service provider's VPN routes. Each PE router maintains a VRF table for each of its directly connected sites. Each customer connection (such as a Frame Relay PVC, an ATM PVC, or a VLAN) is mapped to a specific VRF table. Thus, it is a port on the PE router and not a site that is associated with a VRF table. Multiple ports on a PE router can be associated with a single VRF table. It is the ability of PE routers to maintain multiple forwarding tables that supports the per-VPN segregation of routing information.

- Use of route reflectors—Provider edge routers can maintain IBGP sessions to route reflectors as an alternative to a full mesh of IBGP sessions. Deploying multiple route reflectors enhances the scalability of the RFC 2547bis model because it eliminates the need for any single network component to maintain all VPN routes.
- Multiple VPNs are kept separate and distinct from each other—The customer edge routers do not peer with each other. Two sites have IP connectivity over the common backbone only, and only if there is a VPN which contains both sites. This feature keeps the VPNs separate and distinct from each other, even if two VPNs have an overlapping address space.
- Simple for customers to use—Customers can obtain IP backbone services from a service provider, and they do not need to maintain their own backbones.

Related Documentation

- Layer 2 VPN Overview on page 55
- Layer 3 VPN Overview
- Example: Interconnecting a Layer 2 VPN with a Layer 3 VPN on page 88

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

This example provides a step-by-step procedure and commands for interconnecting and verifying a Layer 2 VPN with a Layer 3 VPN. It contains the following sections:

Requirements

This example uses the following hardware and software components:

- Junos OS Release 9.3 or later
- Five MX Series routers
- Three M Series routers
- Two T Series routers

Overview and Topology

A Layer 2 VPN is a type of virtual private network (VPN) that uses MPLS labels to transport data. The communication occurs between the provider edge (PE) routers.

Layer 2 VPNs use BGP as the signaling protocol and, consequently, have a simpler design and require less provisioning overhead than traditional VPNs over Layer 2 circuits. BGP signaling also enables autodiscovery of Layer 2 VPN peers. Layer 2 VPNs can have either a full-mesh or a hub-and-spoke topology. The tunneling mechanism in the core network

is, typically, MPLS. However, Layer 2 VPNs can also use other tunneling protocols, such as GRE.

Layer 3 VPNs are based on RFC 2547bis, *BGP/MPLS IP VPNs*. RFC 2547bis defines a mechanism by which service providers can use their IP backbones to provide VPN services to their customers. A Layer 3 VPN is a set of sites that share common routing information and whose connectivity is controlled by a collection of policies. The sites that make up a Layer 3 VPN are connected over a provider's existing public Internet backbone. RFC 2547bis VPNs are also known as BGP/MPLS VPNs because BGP is used to distribute VPN routing information across the provider's backbone, and MPLS is used to forward VPN traffic across the backbone to remote VPN sites.

Customer networks, because they are private, can use either public addresses or private addresses, as defined in RFC 1918, *Address Allocation for Private Internets*. When customer networks that use private addresses connect to the public Internet infrastructure, the private addresses might overlap with the same private addresses used by other network users. MPLS/BGP VPNs solve this problem by adding a *distinguisher*, a VPN identifier prefix to each address from a particular VPN site, thereby creating an address that is unique both within the VPN and within the public Internet.

In addition, each VPN has its own VPN-specific routing table that contains the routing information for that VPN only. To separate a VPN's routes from routes in the public Internet or those in other VPNs, the PE router creates a separate routing table for each VPN called a VPN routing and forwarding (VRF) table. The PE router creates one VRF table for each VPN that has a connection to a customer edge (CE) router. Any customer or site that belongs to the VPN can access only the routes in the VRF tables for that VPN. Every VRF table has one or more extended community attributes associated with it that identify the route as belonging to a specific collection of routers. One of these, the *route target* attribute, identifies a collection of sites (VRF tables) to which a PE router distributes routes. The PE router uses the route target to constrain the import of remote routes into its VRF tables.

When an ingress PE router receives routes advertised from a directly connected CE router, it checks the received route against the VRF export policy for that VPN.

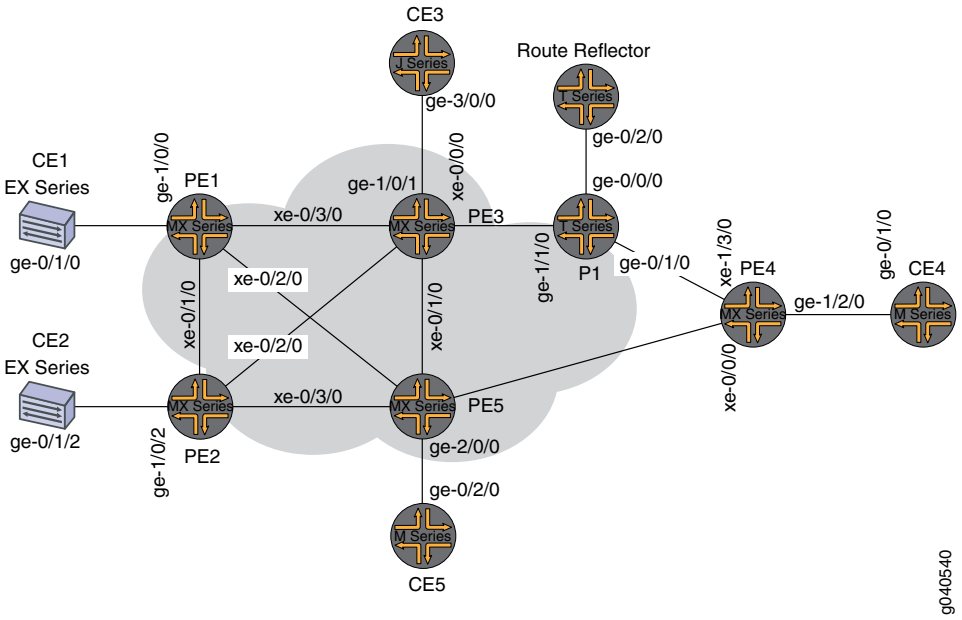
- If it matches, the route is converted to VPN-IPv4 format—that is, the route distinguisher is added to the route. The PE router then announces the route in VPN-IPv4 format to the remote PE routers. It also attaches a route target to each route learned from the directly connected sites. The route target attached to the route is based on the value of the VRF table's configured export target policy. The routes are then distributed using IBGP sessions, which are configured in the provider's core network.
- If the route from the CE router does not match, it is not exported to other PE routers, but it can still be used locally for routing, for example, if two CE routers in the same VPN are directly connected to the same PE router.

When an egress PE router receives a route, it checks it against the import policy on the IBGP session between the PE routers. If it passes, the router places the route into its bgp.l3vpn.0 table. At the same time, the router checks the route against the VRF import policy for the VPN. If it matches, the route distinguisher is removed from the route and

the route is placed into the VRF table (the *routing-instance-name*.inet.0 table) in IPv4 format.

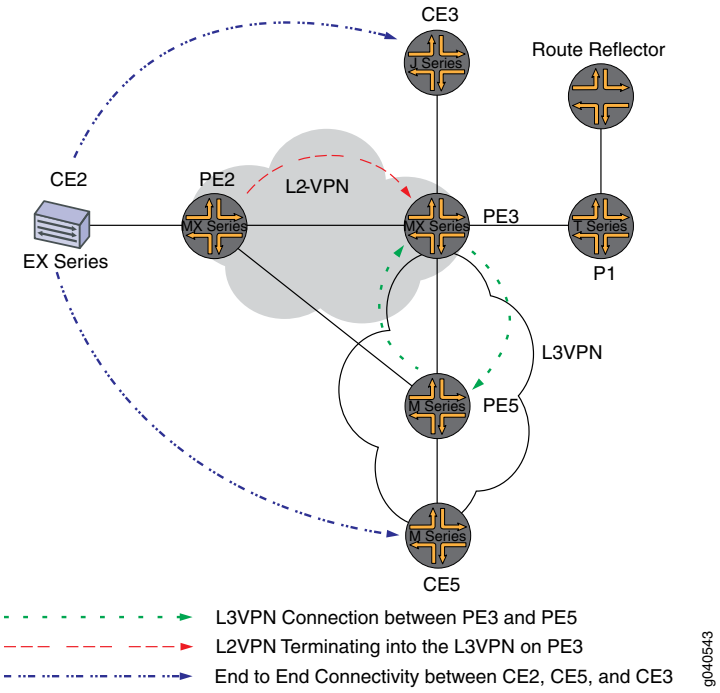
Figure 9 on page 90 shows the physical topology of a Layer 2 VPN-to-Layer 3 VPN interconnection.

Figure 9: Physical Topology of a Layer 2 VPN Terminating into a Layer 3 VPN



The logical topology of a Layer 2 VPN-to-Layer 3 VPN interconnection is shown in Figure 10 on page 91.

Figure 10: Logical Topology of a Layer 2 VPN Terminating into a Layer 3 VPN



The following definitions describe the meaning of the device abbreviations used in Figure 9 on page 90 and Figure 10 on page 91.

- Customer edge (CE) device—A device at the customer premises that provides access to the service provider's VPN over a data link to one or more provider edge (PE) routers.

  Typically the CE device is an IP router that establishes an adjacency with its directly connected PE routers. After the adjacency is established, the CE router advertises the site's local VPN routes to the PE router and learns remote VPN routes from the PE router.

- Provider edge (PE) device—A device, or set of devices, at the edge of the provider network that presents the provider's view of the customer site.

  PE routers exchange routing information with CE routers. PE routers are aware of the VPNs that connect through them, and PE routers maintain VPN state. A PE router is only required to maintain VPN routes for those VPNs to which it is directly attached. After learning local VPN routes from CE routers, a PE router exchanges VPN routing information with other PE routers using IBGP. Finally, when using MPLS to forward VPN data traffic across the provider's backbone, the ingress PE router functions as the ingress label-switching router (LSR) and the egress PE router functions as the egress LSR.

- Provider (P) device—A device that operates inside the provider's core network and does not directly interface to any CE.

  Although the P device is a key part of implementing VPNs for the service provider's customers and may provide routing for many provider-operated tunnels that belong

to different VPNs, it is not itself VPN-aware and does not maintain VPN state. Its principal role is allowing the service provider to scale its VPN offerings, for example, by acting as an aggregation point for multiple PE routers.

P routers function as MPLS transit LSRs when forwarding VPN data traffic between PE routers. P routers are required only to maintain routes to the provider's PE routers; they are not required to maintain specific VPN routing information for each customer site.

Configuration

To interconnect a Layer 2 VPN with a Layer 3 VPN, perform these tasks:

- Configuring the Base Protocols and Interfaces on page 92
- Configuring the VPN Interfaces on page 95

Configuring the Base Protocols and Interfaces

Step-by-Step Procedure

1. On each PE and P router, configure OSPF with traffic engineering extensions on all interfaces. Disable OSPF on the **fxp0.0** interface.

```
[edit protocols]
ospf {
  traffic-engineering;
  area 0.0.0.0 {
    interface all;
    interface fxp0.0 {
      disable;
    }
  }
}
```

2. On all the core routers, enable MPLS on all interfaces. Disable MPLS on the **fxp0.0** interface.

```
[edit protocols]
mpls {
  interface all;
  interface fxp0.0 {
    disable;
  }
}
```

3. On all the core routers, create an internal BGP peer group and specify the route reflector address (7.7.7.7) as the neighbor. Also enable BGP to carry Layer 2 VPLS network layer reachability information (NLRI) messages for this peer group by including the **signaling** statement at the **[edit protocols bgp group *group-name* family l2vpn]** hierarchy level.

```
[edit protocols]
bgp {
  group RR {
    type internal;
    local-address 2.2.2.2;
    family l2vpn {
      signaling;
```

```
    }
    neighbor 7.7.7.7;
  }
}
```

4. On Router PE3, create an internal BGP peer group and specify the route reflector IP address (7.7.7.7) as the neighbor. Enable BGP to carry Layer 2 VPLS NLRI messages for this peer group and enable the processing of VPN-IPv4 addresses by including the **unicast** statement at the **[edit protocols bgp group *group-name* family inet-vpn]** hierarchy level.

```
[edit protocols]
bgp {
  group RR {
    type internal;
    local-address 3.3.3.3;
    family inet-vpn {
      unicast;
    }
    family l2vpn {
      signaling;
    }
    neighbor 7.7.7.7;
  }
}
```

5. For the Layer 3 VPN domain on Router PE3 and Router PE5, enable RSVP on all interfaces. Disable RSVP on the **fxp0.0** interface.

```
[edit protocols]
rsvp {
  interface all;
  interface fxp0.0 {
    disable;
  }
}
```

6. On Router PE3 and Router PE5, create label-switched paths (LSPs) to the route reflector and the other PE routers. The following example shows the configuration on Router PE5.

```
[edit protocols]
mpls {
  label-switched-path to-RR {
    to 7.7.7.7;
  }
  label-switched-path to-PE2 {
    to 2.2.2.2;
  }
  label-switched-path to-PE3 {
    to 3.3.3.3;
  }
  label-switched-path to-PE4 {
    to 4.4.4.4;
  }
  label-switched-path to-PE1 {
    to 1.1.1.1;
```

```
    }
  }
```

7. On Routers PE1, PE2, PE3, and PE5, configure the core interfaces with an IPv4 address and enable the MPLS address family. The following example shows the configuration of the **xe-0/1/0** interface on Router PE2.

```
[edit]
interfaces {
  xe-0/1/0 {
    unit 0 {
      family inet {
        address 10.10.2.2/30;
      }
      family mpls;
    }
  }
}
```

8. On Router PE2 and Router PE3, configure LDP for the Layer 2 VPN MPLS signaling protocol for all interfaces. Disable LDP on the **fxp0.0** interface. (RSVP can also be used.)

```
[edit protocols]
ldp {
  interface all;
  interface fxp0.0 {
    disable;
  }
}
```

9. On the route reflector, create an internal BGP peer group and specify the PE routers IP addresses as the neighbors.

```
[edit]
protocols {
  bgp {
    group RR {
      type internal;
      local-address 7.7.7.7;
      family inet {
        unicast;
      }
      family inet-vpn {
        unicast;
      }
      family l2vpn {
        signaling;
      }
      cluster 7.7.7.7;
      neighbor 1.1.1.1;
      neighbor 2.2.2.2;
      neighbor 4.4.4.4;
      neighbor 5.5.5.5;
      neighbor 3.3.3.3;
    }
  }
```

```
}
```

10. On the route reflector, configure MPLS LSPs towards Routers PE3 and PE5 to resolve the BGP next hops from inet.3 routing table.

```
[edit]
protocols {
  mpls {
    label-switched-path to-pe3 {
      to 3.3.3.3;
    }
    label-switched-path to-pe5 {
      to 5.5.5.5;
    }
    interface all;
  }
}
```

Configuring the VPN Interfaces

Step-by-Step Procedure

Router PE2 is one end of the Layer 2 VPN. Router PE3 is performing the Layer 2 VPN stitching between the Layer 2 VPN and the Layer 3 VPN. Router PE3 uses the logical tunnel interface (**lt** interface) configured with different logical interface units applied under two different Layer 2 VPN instances. The packet is looped though the **lt** interface configured on Router PE3. The configuration of Router PE5 contains the PE-CE interface.

1. On Router PE2, configure the **ge-1/0/2** interface encapsulation. Include the encapsulation statement and specify the **ethernet-ccc** option (**vlan-ccc** encapsulation is also supported) at the **[edit interfaces ge-1/0/2]** hierarchy level. The encapsulation should be the same in a whole Layer 2 VPN domain (Routers PE2 and PE3). Also, configure interface **lo0**.

```
[edit]
interfaces {
  ge-1/0/2 {
    encapsulation ethernet-ccc;
    unit 0;
  }
  lo0 {
    unit 0 {
      family inet {
        address 2.2.2.2/32;
      }
    }
  }
}
```

2. On Router PE2, configure the routing instance at the [**edit routing-instances**] hierarchy level. Also, configure the Layer 2 VPN protocol at the [**edit routing-instances *routing-instances-name* protocols**] hierarchy level. Configure the remote site ID as 3. Site ID 3 represents Router PE3 (Hub-PE). The Layer 2 VPN is using LDP as the signaling protocol. Be aware that in the following example, both the routing instance and the protocol are named **l2vpn**.

```
[edit]
```

```
routing-instances {
  l2vpn { # routing instance
    instance-type l2vpn;
    interface ge-1/0/2.0;
    route-distinguisher 65000:2;
    vrf-target target:65000:2;
    protocols {
      l2vpn { # protocol
        encapsulation-type ethernet;
        site CE2 {
          site-identifier 2;
          interface ge-1/0/2.0 {
            remote-site-id 3;
          }
        }
      }
    }
  }
}
```

3. On Router PE5, configure the Gigabit Ethernet interface for the PE-CE link **ge-2/0/0** and configure the **lo0** interface.

```
[edit interfaces]
ge-2/0/0 {
  unit 0 {
    family inet {
      address 80.80.80.1/24;
    }
  }
}
lo0 {
  unit 0 {
  }
}
```

4. On Router PE5, configure the Layer 3 VPN routing instance (**L3VPN**) at the **[edit routing-instances]** hierarchy level. Also configure BGP at the **[edit routing-instances L3VPN protocols]** hierarchy level.

```
[edit]
routing-instances {
  L3VPN {
    instance-type vrf;
    interface ge-2/0/0.0;
    route-distinguisher 65000:5;
    vrf-target target:65000:2;
    vrf-table-label;
    protocols {
      bgp {
        group ce5 {
          neighbor 80.80.80.2 {
            peer-as 200;
          }
        }
      }
    }
```

```
    }
  }
```

5. In an MX Series router, such as Router PE3, you must create the tunnel services interface to be used for tunnel services. To create the tunnel service interface, include the **bandwidth** statement and specify the amount of bandwidth to reserve for tunnel services in gigabits per second at the **[edit chassis fpc *slot-number* pic *slot-number* tunnel-services]** hierarchy level.

```
[edit]
chassis {
  dump-on-panic;
  fpc 1 {
    pic 1 {
      tunnel-services {
        bandwidth 1g;
      }
    }
  }
}
```

6. On Router PE3, configure the Gigabit Ethernet interface.

   Include the **address** statement at the **[edit interfaces ge-1/0/1.0 family inet]** hierarchy level and specify **90.90.90.1/24** as the IP address.

```
[edit]
interfaces {
  ge-1/0/1 {
    unit 0 {
      family inet {
        address 90.90.90.1/24;
      }
    }
  }
}
```

7. On Router PE3, configure the **lt-1/1/10.0** logical tunnel interface at the **[edit interfaces lt-1/1/10 unit 0]** hierarchy level. Router PE3 is the router that is *stitching* the Layer 2 VPN to the Layer 3 VPN using the logical tunnel interface. The configuration of the peer unit interfaces is what makes the interconnection.

   To configure the interface, include the **encapsulation** statement and specify the **ethernet-ccc** option. Include the **peer-unit** statement and specify the logical interface unit **1** as the peer tunnel interface. Include the **family** statement and specify the **ccc** option.

```
[edit]
interfaces {
  lt-1/1/10 {
    unit 0 {
      encapsulation ethernet-ccc;
      peer-unit 1;
      family ccc;
    }
  }
}
```

8. On Router PE3, configure the **lt-1/1/10.1** logical tunnel interface at the **[edit interfaces lt-1/1/10 unit 1]** hierarchy level.

   To configure the interface, include the **encapsulation** statement and specify the **ethernet** option. Include the **peer-unit** statement and specify the logical interface unit **0** as the peer tunnel interface. Include the **family** statement and specify the **inet** option. Include the **address** statement at the **[edit interfaces lt-1/1/10 unit 0]** hierarchy level and specify **70.70.70.1/24** as the IPv4 address.

   ```
   [edit]
   interfaces {
     lt-1/1/10 {
       unit 1 {
         encapsulation ethernet;
         peer-unit 0;
         family inet {
           address 70.70.70.1/24;
         }
       }
     }
   }
   ```

9. On Router PE3, add the **lt** interface unit 1 to the routing instance at the **[edit routing-instances L3VPN]** hierarchy level. Configure the instance type as **vrf** with **lt** peer-unit 1 as a PE-CE interface to terminate the Layer 2 VPN on Router PE2 into the Layer 3 VPN on Router PE3.

   ```
   [edit]
   routing-instances {
     L3VPN {
       instance-type vrf;
       interface ge-1/0/1.0;
       interface lt-1/1/10.1;
       route-distinguisher 65000:33;
       vrf-target target:65000:2;
       vrf-table-label;
       protocols {
         bgp {
           export direct;
           group ce3 {
             neighbor 90.90.90.2 {
               peer-as 100;
             }
           }
         }
       }
     }
   }
   ```

10. On Router PE3, add the **lt** interface unit 0 to the routing instance at the **[edit routing-instances protocols l2vpn]** hierarchy level. Also configure the same vrf target for the Layer 2 VPN and Layer 3 VPN routing instances, so that the routes can be leaked between the instances. The example configuration in the previous step shows the vrf target for the **L3VPN** routing instance. The following example shows the vrf target for the **l2vpn** routing instance.

```
[edit]
routing-instances {
  l2vpn {
    instance-type l2vpn;
    interface lt-1/1/10.0;
    route-distinguisher 65000:3;
    vrf-target target:65000:2;
    protocols {
      l2vpn {
        encapsulation-type ethernet;
        site CE3 {
          site-identifier 3;
          interface lt-1/1/10.0 {
            remote-site-id 2;
          }
        }
      }
    }
  }
}
```

11. On Router PE3, configure the **policy-statement** statement to export the routes learned from the directly connected **lt** interface unit 1 to all the CE routers for connectivity, if needed.

```
[edit]
policy-options {
  policy-statement direct {
    term 1 {
      from protocol direct;
      then accept;
    }
  }
}
```

Results The following output shows the full configuration of Router PE2:

Router PE2

```
interfaces {
  xe-0/1/0 {
    unit 0 {
      family inet {
        address 10.10.2.2/30;
      }
      family mpls;
    }
  }
  xe-0/2/0 {
    unit 0 {
      family inet {
        address 10.10.5.1/30;
      }
      family mpls;
    }
  }
  xe-0/3/0 {
    unit 0 {
```

```
        family inet {
          address 10.10.4.1/30;
        }
        family mpls;
      }
    }
    ge-1/0/2 {
      encapsulation ethernet-ccc;
      unit 0;
    }
    fxp0 {
      apply-groups [ re0 re1 ];
    }
    lo0 {
      unit 0 {
        family inet {
          address 2.2.2.2/32;
        }
      }
    }
}
routing-options {
    static {
      route 172.0.0.0/8 next-hop 172.19.59.1;
    }
    autonomous-system 65000;
}
protocols {
    mpls {
      interface all;
      interface fxp0.0 {
        disable;
      }
    }
    bgp {
      group RR {
        type internal;
        local-address 2.2.2.2;
        family l2vpn {
          signaling;
        }
        neighbor 7.7.7.7;
      }
    }
    ospf {
      traffic-engineering;
      area 0.0.0.0 {
        interface all;
        interface fxp0.0 {
          disable;
        }
      }
    }
    ldp {
      interface all;
      interface fxp0.0 {
```

```
        disable;
      }
    }
  }
  routing-instances {
    l2vpn {
      instance-type l2vpn;
      interface ge-1/0/2.0;
      route-distinguisher 65000:2;
      vrf-target target:65000:2;
      protocols {
        l2vpn {
          encapsulation-type ethernet;
          site CE2 {
            site-identifier 2;
            interface ge-1/0/2.0 {
              remote-site-id 3;
            }
          }
        }
      }
    }
  }
```

The following output shows the final configuration of Router PE5:

Router PE5

```
interfaces {
  ge-0/0/0 {
    unit 0 {
      family inet {
        address 10.10.4.2/30;
      }
      family mpls;
    }
  }
  xe-0/1/0 {
    unit 0 {
      family inet {
        address 10.10.6.2/30;
      }
      family mpls;
    }
  }
  ge-1/0/0 {
    unit 0 {
      family inet {
        address 10.10.9.1/30;
      }
      family mpls;
    }
  }
  xe-1/1/0 {
    unit 0 {
      family inet {
        address 10.10.3.2/30;
      }
```

```
            family mpls;
        }
    }
    ge-2/0/0 {
        unit 0 {
            family inet {
                address 80.80.80.1/24;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 5.5.5.5/32;
            }
        }
    }
}
routing-options {
    static {
        route 172.0.0.0/8 next-hop 172.19.59.1;
    }
    autonomous-system 65000;
}
protocols {
    rsvp {
        interface all {
            link-protection;
        }
        interface fxp0.0 {
            disable;
        }
    }
    mpls {
        label-switched-path to-RR {
            to 7.7.7.7;
        }
        label-switched-path to-PE2 {
            to 2.2.2.2;
        }
        label-switched-path to-PE3 {
            to 3.3.3.3;
        }
        label-switched-path to-PE4 {
            to 4.4.4.4;
        }
        label-switched-path to-PE1 {
            to 1.1.1.1;
        }
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
    bgp {
        group to-rr {
```

```
        type internal;
        local-address 5.5.5.5;
        family inet-vpn {
          unicast;
        }
        family l2vpn {
          signaling;
        }
        neighbor 7.7.7.7;
      }
    }
    ospf {
      traffic-engineering;
      area 0.0.0.0 {
        interface all;
        interface fxp0.0 {
          disable;
        }
      }
    }
    ldp {
      interface all;
      interface fxp0.0 {
        disable;
      }
    }
  }
  routing-instances {
    L3VPN {
      instance-type vrf;
      interface ge-2/0/0.0;
      route-distinguisher 65000:5;
      vrf-target target:65000:2;
      vrf-table-label;
      protocols {
        bgp {
          group ce5 {
            neighbor 80.80.80.2 {
              peer-as 200;
            }
          }
        }
      }
    }
  }
```

The following output shows the final configuration of Router PE3:

Router PE3

```
chassis {
  dump-on-panic;
  fpc 1 {
    pic 1 {
      tunnel-services {
        bandwidth 1g;
      }
    }
```

```
    }
    network-services ip;
}
interfaces {
    ge-1/0/1 {
        unit 0 {
            family inet {
                address 90.90.90.1/24;
            }
        }
    }
    lt-1/1/10 {
        unit 0 {
            encapsulation ethernet-ccc;
            peer-unit 1;
            family ccc;
        }
        unit 1 {
            encapsulation ethernet;
            peer-unit 0;
            family inet {
                address 70.70.70.1/24;
            }
        }
    }
    xe-2/0/0 {
        unit 0 {
            family inet {
                address 10.10.20.2/30;
            }
            family mpls;
        }
    }
    xe-2/1/0 {
        unit 0 {
            family inet {
                address 10.10.6.1/30;
            }
            family mpls;
        }
    }
    xe-2/2/0 {
        unit 0 {
            family inet {
                address 10.10.5.2/30;
            }
            family mpls;
        }
    }
    xe-2/3/0 {
        unit 0 {
            family inet {
                address 10.10.1.2/30;
            }
            family mpls;
        }
```

```
    }
    lo0 {
      unit 0 {
        family inet {
          address 3.3.3.3/32;
        }
      }
    }
}
routing-options {
    static {
      route 172.0.0.0/8 next-hop 172.19.59.1;
    }
    autonomous-system 65000;
}
protocols {
    rsvp {
      interface all;
      interface fxp0.0 {
        disable;
      }
    }
    mpls {
      label-switched-path to-RR {
        to 7.7.7.7;
      }
      }
      label-switched-path to-PE2 {
        to 2.2.2.2;
      }
      label-switched-path to-PE5 {
        to 5.5.5.5;
      }
      label-switched-path to-PE4 {
        to 4.4.4.4;
      }
      label-switched-path to-PE1 {
        to 1.1.1.1;
      }
      interface all;
      interface fxp0.0 {
        disable;
      }
    }
    bgp {
      group RR {
        type internal;
        local-address 3.3.3.3;
        family inet-vpn {
          unicast;
        }
        family l2vpn {
          signaling;
        }
        neighbor 7.7.7.7;
      }
```

```
    }
    ospf {
      traffic-engineering;
      area 0.0.0.0 {
        interface all;
        interface fxp0.0 {
          disable;
        }
      }
    }
    ldp {
      interface all;
      interface fxp0.0 {
        disable;
      }
    }
  }
  policy-options {
    policy-statement direct {
      term 1 {
        from protocol direct;
        then accept;
      }
    }
  }
  routing-instances {
    L3VPN {
      instance-type vrf;
      interface ge-1/0/1.0;
      interface lt-1/1/10.1;
      route-distinguisher 65000:33;
      vrf-target target:65000:2;
      vrf-table-label;
      protocols {
        bgp {
          export direct;
          group ce3 {
            neighbor 90.90.90.2 {
              peer-as 100;
            }
          }
        }
      }
    }
    l2vpn {
      instance-type l2vpn;
      interface lt-1/1/10.0;
      route-distinguisher 65000:3;
      vrf-target target:65000:2;
      protocols {
        l2vpn {
          encapsulation-type ethernet;
          site CE3 {
            site-identifier 3;
            interface lt-1/1/10.0 {
              remote-site-id 2;
```

```
            }
          }
        }
      }
    }
  }
```

Verification

Verify the Layer 2 VPN-to-Layer 3 VPN interconnection:

- Verifying Router PE2 VPN Interface on page 107
- Verifying Router PE3 VPN Interface on page 109
- Verifying End-to-End connectivity from Router CE2 to Router CE5 and Router CE3 on page 111

Verifying Router PE2 VPN Interface

Purpose Check that the Layer 2 VPN is up and working at the Router PE2 interface and that all the routes are there.

Action 1. Use the **show l2vpn connections** command to verify that the connection site ID is 3 for Router PE3 and that the status is **Up**.

```
user@PE2> 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

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

Instance: l2vpn
  Local site: CE2 (2)
    connection-site     Type  St     Time last up        # Up trans
    3                   rmt   Up     Jan  7 14:14:37 2010         1
      Remote PE: 3.3.3.3, Negotiated control-word: Yes (Null)
```

```
        Incoming label: 800000, Outgoing label: 800001
        Local interface: ge-1/0/2.0, Status: Up, Encapsulation: ETHERNET
```

2. Use the **show route table** command to verify that the Layer 2 VPN route is present and that there is a next hop of **10.10.5.2** through the **xe-0/2/0.0** interface. The following output verifies that the Layer 2 VPN routes are present in the l2vpn.l2vpn.0 table. Similar output should be displayed for Router PE3.

```
user@PE2> show route table l2vpn.l2vpn.0

l2vpn.l2vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

65000:2:2:3/96
                   *[L2VPN/170/-101] 02:40:35, metric2 1
                      Indirect
65000:3:3:1/96
                   *[BGP/170] 02:40:35, localpref 100, from 7.7.7.7
                      AS path: I
                    > to 10.10.5.2 via xe-0/2/0.0
```

3. Verify that Router PE2 has a Layer 2 VPN MPLS label pointing to the LDP label to Router PE3 in both directions (PUSH and POP).

```
user@PE2> show route table mpls.0

 mpls.0: 13 destinations, 13 routes (13 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 1w3d 08:57:41, metric 1
                      Receive
1                  *[MPLS/0] 1w3d 08:57:41, metric 1
                      Receive
2                  *[MPLS/0] 1w3d 08:57:41, metric 1
                      Receive
300560             *[LDP/9] 19:45:53, metric 1
                    > to 10.10.2.1 via xe-0/1/0.0, Pop
300560(S=0)        *[LDP/9] 19:45:53, metric 1
                    > to 10.10.2.1 via xe-0/1/0.0, Pop
301008             *[LDP/9] 19:45:53, metric 1
                    > to 10.10.4.2 via xe-0/3/0.0, Swap 299856
301536             *[LDP/9] 19:45:53, metric 1
                    > to 10.10.4.2 via xe-0/3/0.0, Pop
301536(S=0)        *[LDP/9] 19:45:53, metric 1
                    > to 10.10.4.2 via xe-0/3/0.0, Pop
301712             *[LDP/9] 16:14:52, metric 1
                    > to 10.10.5.2 via xe-0/2/0.0, Swap 315184
301728             *[LDP/9] 16:14:52, metric 1
                    > to 10.10.5.2 via xe-0/2/0.0, Pop
301728(S=0)        *[LDP/9] 16:14:52, metric 1
                    > to 10.10.5.2 via xe-0/2/0.0, Pop
800000       *[L2VPN/7] 02:40:35
         > via ge-1/0/2.0, Pop     Offset: 4
ge-1/0/2.0     *[L2VPN/7] 02:40:35, metric2 1
         > to 10.10.5.2 via xe-0/2/0.0, Push 800001 Offset: -4
```

Meaning The **l2vpn** routing instance is up at interface **ge-1/0/2** and the Layer 2 VPN route is shown in table l2vpn.l2vpn.0. Table **mpls.0** shows the Layer 2 VPN routes used to forward the traffic using an LDP label.

Verifying Router PE3 VPN Interface

Purpose Check that the Layer 2 VPN connection from Router PE2 and Router PE3 is **Up** and working.

Action 1. Verify that the BGP session with the route reflector for the family **l2vpn-signaling** and the family **inet-vpn** is established.

```
user@PE3> show bgp summary

Groups: 2 Peers: 2 Down peers: 0
Table        Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l2vpn.0          1          1          0          0          0          0
bgp.L3VPN.0          1          1          0          0          0          0
Peer        AS   InPkt    OutPkt   OutQ   Flaps Last Up/Dwn   State|#Active /Received/Accepted/Damped...
7.7.7.7  65000    2063      2084      0       1    15:35:16   Establ
  bgp.l2vpn.0: 1/1/1/0
  bgp.L3VPN.0: 1/1/1/0
  L3VPN.inet.0: 1/1/1/0
  l2vpn.l2vpn.0: 1/1/1/0
```

2. The following output shows the L3VPN.inet.0 routing table, which has Routers CE1, CE3, and CE5 listed.

```
user@PE3> show route table L3VPN.inet.0

L3VPN.inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

70.70.70.0/24      *[Direct/0] 02:45:16
                    > via lt-1/1/10.1
70.70.70.1/32      *[Local/0] 14:45:42
                       Local via lt-1/1/10.1
80.80.80.0/24      *[BGP/170] 02:47:51, localpref 100, from 7.7.7.7
                       AS path: I
                    > to 10.10.6.2 via xe-2/1/0.0, Push 16
90.90.90.0/24      *[Direct/0] 15:26:24
                    > via ge-1/0/1.0
90.90.90.1/32      *[Local/0] 15:26:24
                       Local via ge-1/0/1.0
```

3. The following output verifies the Layer 2 VPN route and the label associated with it.

```
user@PE3> show route table l2vpn.l2vpn.0 detail

l2vpn.l2vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
65000:2:2:3/96 (1 entry, 1 announced)
        *BGP    Preference: 170/-101
                Route Distinguisher: 65000:2
                Next hop type: Indirect
                Next-hop reference count: 4
                Source: 7.7.7.7
                Protocol next hop: 2.2.2.2
                Indirect next hop: 2 no-forward
                State: <Secondary Active Int Ext>
                Local AS: 65000 Peer AS: 65000
                Age: 2:45:52    Metric2: 1
                Task: BGP_65000.7.7.7.7+60585
                Announcement bits (1): 0-l2vpn-l2vpn
                AS path: I (Originator) Cluster list:  7.7.7.7
                AS path:  Originator ID: 2.2.2.2
```

```
                Communities: target:65000:2 Layer2-info: encaps:ETHERNET,
 control flags:Control-Word, mtu: 0, site preference: 100  Accepted
                Label-base: 800000, range: 2, status-vector: 0x0
                Localpref: 100
                Router ID: 7.7.7.7
                Primary Routing Table bgp.l2vpn.0
```

4. The following output show the L2VPN MPLS.0 route in the mpls.0 route table.

```
user@PE3> show route table mpls.0

mpls.0: 21 destinations, 21 routes (21 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 1w3d 09:05:41, metric 1
                      Receive
1                  *[MPLS/0] 1w3d 09:05:41, metric 1
                      Receive
2                  *[MPLS/0] 1w3d 09:05:41, metric 1
                      Receive
16                 *[VPN/0] 15:59:24
                      to table L3VPN.inet.0, Pop
315184             *[LDP/9] 16:21:53, metric 1
                    > to 10.10.20.1 via xe-2/0/0.0, Pop
315184(S=0)        *[LDP/9] 16:21:53, metric 1
                    > to 10.10.20.1 via xe-2/0/0.0, Pop
315200             *[LDP/9] 01:13:44, metric 1
                      to 10.10.20.1 via xe-2/0/0.0, Swap 625297
                    > to 10.10.6.2 via xe-2/1/0.0, Swap 299856
315216             *[LDP/9] 16:21:53, metric 1
                    > to 10.10.6.2 via xe-2/1/0.0, Pop
315216(S=0)        *[LDP/9] 16:21:53, metric 1
                    > to 10.10.6.2 via xe-2/1/0.0, Pop
315232             *[LDP/9] 16:21:45, metric 1
                    > to 10.10.1.1 via xe-2/3/0.0, Pop
315232(S=0)        *[LDP/9] 16:21:45, metric 1
                    > to 10.10.1.1 via xe-2/3/0.0, Pop
315248             *[LDP/9] 16:21:53, metric 1
                    > to 10.10.5.1 via xe-2/2/0.0, Pop
315248(S=0)        *[LDP/9] 16:21:53, metric 1
                    > to 10.10.5.1 via xe-2/2/0.0, Pop
315312             *[RSVP/7] 15:02:40, metric 1
                    > to 10.10.6.2 via xe-2/1/0.0, label-switched-path
to-pe5
315312(S=0)        *[RSVP/7] 15:02:40, metric 1
                    > to 10.10.6.2 via xe-2/1/0.0, label-switched-path
to-pe5
315328             *[RSVP/7] 15:02:40, metric 1
                    > to 10.10.20.1 via xe-2/0/0.0, label-switched-path
to-RR
315360             *[RSVP/7] 15:02:40, metric 1
                    > to 10.10.20.1 via xe-2/0/0.0, label-switched-path
to-RR
316272             *[RSVP/7] 01:13:27, metric 1
                    > to 10.10.6.2 via xe-2/1/0.0, label-switched-path
Bypass->10.10.9.1
316272(S=0)        *[RSVP/7] 01:13:27, metric 1
                    > to 10.10.6.2 via xe-2/1/0.0, label-switched-path
Bypass->10.10.9.1
800001             *[L2VPN/7] 02:47:33
                    > via lt-1/1/10.0, Pop       Offset: 4
```

```
lt-1/1/10.0        *[L2VPN/7] 02:47:33, metric2 1
                    > to 10.10.5.1 via xe-2/2/0.0, Push 800000 Offset: -4
```

5. Use the **show route table mpls.0** command with the **detail** option to see the BGP attributes of the route such as next-hop type and label operations.

```
user@PE5> show route table mpls.0 detail

lt-1/1/10.0 (1 entry, 1 announced)
        *L2VPN  Preference: 7
                Next hop type: Indirect
                Next-hop reference count: 2
                Next hop type: Router, Next hop index: 607
                Next hop: 10.10.5.1 via xe-2/2/0.0, selected
                Label operation: Push 800000 Offset: -4
                Protocol next hop: 2.2.2.2
                Push 800000 Offset: -4
                Indirect next hop: 8cae0a0 1048574
                State:  <Active Int>
                Age: 2:46:34    Metric2: 1
                Task: Common L2 VC
                Announcement bits (2): 0-KRT 2-Common L2 VC
                AS path: I
                Communities: target:65000:2 Layer2-info: encaps:ETHERNET,
 control flags:Control-Word, mtu: 0, site preference: 100
```

Verifying End-to-End connectivity from Router CE2 to Router CE5 and Router CE3

Purpose Check the connectivity between Routers CE2, CE3, and CE5.

Action

1. Ping the Router CE3 IP address from Router CE2.

```
user@CE2> ping 90.90.90.2 # CE3 IP address

PING 90.90.90.2 (90.90.90.2): 56 data bytes
64 bytes from 90.90.90.2: icmp_seq=0 ttl=63 time=0.708 ms
64 bytes from 90.90.90.2: icmp_seq=1 ttl=63 time=0.610 ms
```

2. Ping the Router CE5 IP address from Router CE2.

```
user@CE2> ping 80.80.80.2 # CE5 IP address

PING 80.80.80.2 (80.80.80.2): 56 data bytes
64 bytes from 80.80.80.2: icmp_seq=0 ttl=62 time=0.995 ms
64 bytes from 80.80.80.2: icmp_seq=1 ttl=62 time=1.005 ms
```

Related Documentation

- Layer 2 VPN Overview on page 55
- Layer 3 VPN Overview
- Interconnecting Layer 2 VPNs with Layer 3 VPNs Overview on page 87

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

This example provides a step-by-step procedure and commands for configuring and verifying a Layer 2 circuit to Layer 3 VPN interconnection. It contains the following sections:

Requirements

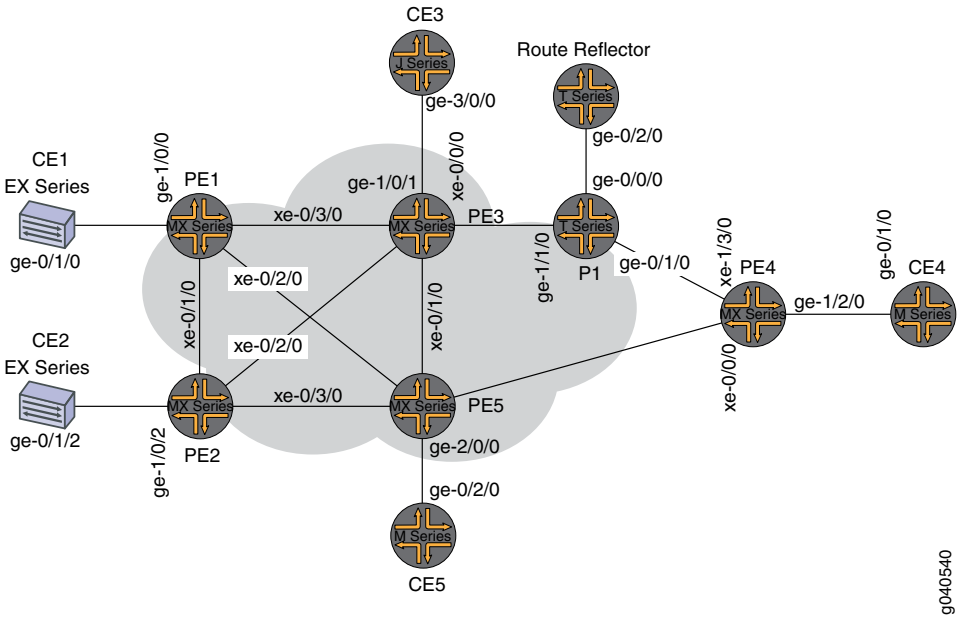
This example uses the following hardware and software components:

- Junos OS Release 9.3 or later
- 3 MX Series routers
- 1 M Series routers
- 1 T Series router
- 1 EX Series router
- 1 J Series router

Overview and Topology

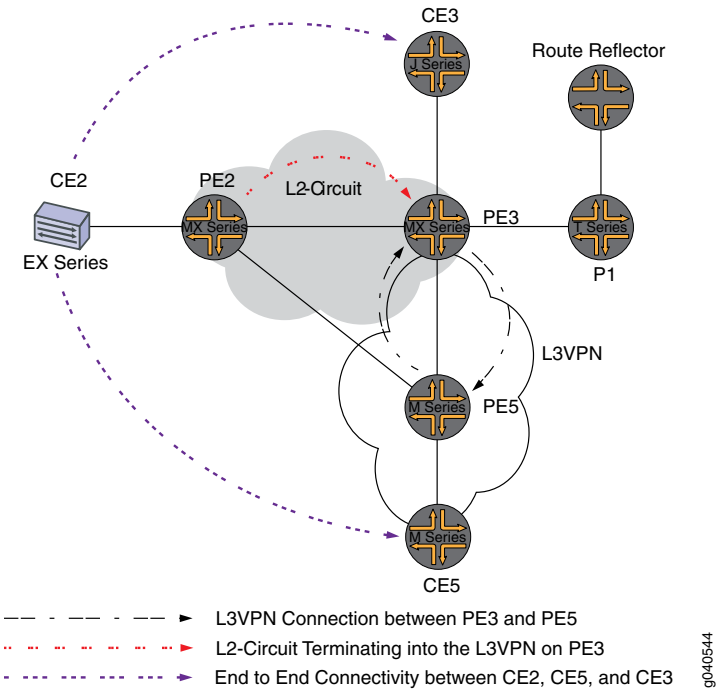
The physical topology of a Layer 2 circuit to Layer 3 VPN interconnection is shown in Figure 11 on page 113.

Figure 11: Physical Topology of a Layer 2 Circuit to Layer 3 VPN Interconnection



The logical topology of a Layer 2 circuit to Layer 3 VPN interconnection is shown in Figure 12 on page 113.

Figure 12: Logical Topology of a Layer 2 Circuit to Layer 3 VPN Interconnection



Configuration

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

In this example, the router being configured is identified using the following command prompts:

- **CE2** identifies the customer edge 2 (CE2) router
- **PE1** identifies the provider edge 1 (PE1) router
- **CE3** identifies the customer edge 3 (CE3) router
- **PE3** identifies the provider edge 3 (PE3) router
- **CE5** identifies the customer edge 5 (CE5) router
- **PE5** identifies the provider edge 5 (PE5) router

This example contains the following procedures:

- Configuring PE Router Customer-facing and Loopback Interfaces on page 114
- Configuring Core-facing Interfaces on page 116
- Configuring Protocols on page 117
- Configuring Routing Instances and Layer 2 Circuits on page 120
- Configuring the Route Reflector on page 122
- Interconnecting the Layer 2 Circuit with the Layer 3 VPN on page 123

Configuring PE Router Customer-facing and Loopback Interfaces

Step-by-Step Procedure

To begin building the interconnection, configure the interfaces on the PE routers. If your network contains provider (P) routers, configure the interfaces on the P routers also. This example shows the configuration for Router PE2, Router PE3, and Router PE5.

1. On Router PE2, configure the **ge-1/0/2** interface encapsulation. To configure the interface encapsulation, include the **encapsulation** statement and specify the **ethernet-ccc** option (**vlan-ccc** encapsulation is also supported). Configure the **ge-1/0/2.0** logical interface family for circuit cross-connect functionality. To configure the logical interface family, include the **family** statement and specify the **ccc** option. The encapsulation should be configured the same way for all routers in the Layer 2 circuit domain.

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

2. On Router PE2, configure the **lo0.0** interface. Include the **family** statement and specify the **inet** option. Include the **address** statement and specify **2.2.2.2/32** as the loopback IPv4 address.

```
[edit interfaces]
lo0 {
  unit 0 {
    family inet {
      address 2.2.2.2/32;
    }
  }
}
```

3. On Router PE3, configure the **ge-1/0/1** interface. Include the **family** statement and specify the **inet** option. Include the **address** statement and specify **90.90.90.1/24** as the interface address for this device.

```
[edit interfaces]
ge-1/0/1 {
  unit 0 {
    family inet {
      address 90.90.90.1/24;
    }
  }
}
```

4. On Router PE3, configure the **lo0.0** loopback interface. Include the **family** statement and specify the **inet** option. Include the **address** statement and specify **3.3.3.3/32** as the loopback IPv4 address for this router.

```
[edit interfaces]
lo0 {
  unit 0 {
    family inet {
      address 3.3.3.3/32;
    }
  }
}
```

5. On Router PE5, configure the **ge-2/0/0** interface. Include the **family** statement and specify the **inet** option. Include the **address** statement and specify **80.80.80.1/24** as the interface address.

```
[edit interfaces]
ge-2/0/0 {
  unit 0 {
    family inet {
      address 80.80.80.1/24;
    }
  }
}
```

6. On Router PE5, configure the **lo0.0** interface. Include the **family** statement and specify the **inet** option. Include the **address** statement and specify **5.5.5.5/32** as the loopback IPv4 address for this router.

```
[edit interfaces]
lo0 {
```

```
    unit 0 {
      family inet {
        address 5.5.5.5/32;
      }
    }
  }
```

Configuring Core-facing Interfaces

Step-by-Step Procedure

This procedure describes how to configure the core-facing interfaces on the PE routers. This example does not include all the core-facing interfaces shown in the physical topology illustration. Enable the **mpls** and **inet** address families on the core-facing interfaces.

1. On Router PE2, configure the **xe-0/2/0** interface. Include the **family** statement and specify the **inet** address family. Include the **address** statement and specify **10.10.5.1/30** as the interface address. Include the **family** statement and specify the **mpls** address family.

```
[edit interfaces]
xe-0/2/0 {
  unit 0 {
    family inet {
      address 10.10.5.1/30;
    }
    family mpls;
  }
}
```

2. On Router PE3, configure the core-facing interfaces. Include the **family** statement and specify the **inet** address family. Include the **address** statement and specify the IPv4 addresses shown in the example as the interface addresses. Include the **family** statement and specify the **mpls** address family. In the example, the **xe-2/1/0** interface is connected to Router PE5, and the **xe-2/2/0** interface is connected to Router PE2.

```
[edit interfaces]
xe-2/0/0 {
  unit 0 {
    family inet {
      address 10.10.20.2/30;
    }
    family mpls;
  }
}
xe-2/1/0 {
  unit 0 {
    family inet {
      address 10.10.6.1/30;
    }
    family mpls;
  }
}
xe-2/2/0 {
  unit 0 {
    family inet {
```

```
        address 10.10.5.2/30;
      }
      family mpls;
    }
  }
  xe-2/3/0 {
    unit 0 {
      family inet {
        address 10.10.1.2/30;
      }
      family mpls;
    }
  }
```

3. On Router PE5, configure the **xe-0/1/0** interface. Include the **family** statement and specify the **inet** address family. Include the **address** statement and specify **10.10.6.2/30** as the interface address. Include the **family** statement and specify the **mpls** address family.

```
[edit interfaces]
xe-0/1/0 {
  unit 0 {
    family inet {
      address 10.10.6.2/30;
    }
    family mpls;
  }
}
```

Configuring Protocols

Step-by-Step Procedure

This procedure describes how to configure the protocols used in this example. If your network contains P routers, configure the interfaces on the P routers also.

1. On Router PE3, enable OSPF as the IGP. Enable the MPLS, LDP, and BGP protocols on all interfaces except **fxp.0**. LDP is used as the signaling protocol for the Layer 2 circuit to Router PE2 . The following configuration snippet shows the protocol configuration for Router PE3:

```
[edit]
protocols {
  rsvp {
    interface all;
    interface fxp0.0 {
      disable;
    }
  }
  mpls {
    label-switched-path to-RR {
      to 7.7.7.7;
    }
    label-switched-path to-PE2 {
      to 2.2.2.2;
    }
    label-switched-path to-PE5 {
```

```
        to 5.5.5.5;
      }
      label-switched-path to-PE4 {
        to 4.4.4.4;
      }
      label-switched-path to-PE1 {
        to 1.1.1.1;
      }
      interface all;
      interface fxp0.0 {
        disable;
      }
    }
    bgp {
      group RR {
        type internal;
        local-address 3.3.3.3;
        family inet-vpn {
          unicast;
        }
        family l2vpn {
          signaling;
        }
        neighbor 7.7.7.7;
      }
    }
    ospf {
      traffic-engineering;
      area 0.0.0.0 {
        interface all;
        interface fxp0.0 {
          disable;
        }
      }
    }
    ldp {
      interface all;
      interface fxp0.0 {
        disable;
      }
    }
  }
```

2. On Router PE2, configure the MPLS, OSPF, and LDP protocols.

```
[edit ]
protocols {
  mpls {
    interface all;
    interface fxp0.0 {
      disable;
    }
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
```

```
        interface all;
        interface fxp0.0 {
          disable;
        }
      }
    }
    ldp {
      interface all;
      interface fxp0.0 {
        disable;
      }
    }
  }
```

3. On Router PE5, enable OSPF as the IGP. Enable the MPLS, RSVP, and BGP protocols on all interfaces except **fxp.0**. Enable core-facing interfaces with the **mpls** and **inet** address families.

```
[edit]
protocols {
  rsvp {
    interface all {
      link-protection;
    }
    interface fxp0.0 {
      disable;
    }
  }
  mpls {
    label-switched-path to-RR {
      to 7.7.7.7;
    }
    label-switched-path to-PE2 {
      to 2.2.2.2;
    }
    label-switched-path to-PE3 {
      to 3.3.3.3;
    }
    label-switched-path to-PE4 {
      to 4.4.4.4;
    }
    label-switched-path to-PE1 {
      to 1.1.1.1;
    }
    interface all;
    interface fxp0.0 {
      disable;
    }
  }
  bgp {
    group to-rr {
      type internal;
      local-address 5.5.5.5;
      family inet-vpn {
        unicast;
      }
```

```
        family l2vpn {
          signaling;
        }
        neighbor 7.7.7.7;
      }
    }
    ospf {
      traffic-engineering;
      area 0.0.0.0 {
        interface all;
        interface fxp0.0 {
          disable;
        }
      }
    }
}
```

Configuring Routing Instances and Layer 2 Circuits

Step-by-Step Procedure

This procedure describes how to configure the Layer 2 circuit and the Layer 3 VPN.

1. On Router PE2, configure the Layer 2 circuit. Include the **l2circuit** statement. Include the **neighbor** statement and specify the loopback IPv4 address of Router PE3 as the neighbor. Include the interface statement and specify **ge-1/0/2.0** as the logical interface that is participating in the Layer 2 circuit. Include the **virtual-circuit-id** statement and specify **100** as the identifier. Include the **no-control-word** statement for equipment that does not support the control word.

   ```
   [edit ]
   protocols {
     l2circuit {
       neighbor 3.3.3.3 {
         interface ge-1/0/2.0 {
           virtual-circuit-id 100;
           no-control-word;
         }
       }
     }
   }
   ```

2. On Router PE3, configure the Layer 2 circuit to Router PE2. Include the **l2circuit** statement. Include the **neighbor** statement and specify the loopback IPv4 address of Router PE2 as the neighbor. Include the interface statement and specify **lt-1/1/10.0** as the logical tunnel interface that is participating in the Layer 2 circuit. Include the **virtual-circuit-id** statement and specify **100** as the identifier. Include the **no-control-word** statement.

   ```
   [edit ]
   protocols {
     l2circuit {
       neighbor 2.2.2.2 {
         interface lt-1/1/10.0 {
           virtual-circuit-id 100;
           no-control-word;
         }
   ```

```
    }
  }
}
```

3. On Router PE3, configure the Layer 3 VPN (**L3VPN**) routing instance to Router PE5 at the **[edit routing-instances]** hierarchy level. Also configure the BGP peer group at the **[edit routing-instances L3VPN protocols]** hierarchy level.

```
[edit ]
routing-instances {
  L3VPN {
    instance-type vrf;
    interface ge-1/0/1.0;
    interface lt-1/1/10.1;
    route-distinguisher 65000:33;
    vrf-target target:65000:2;
    vrf-table-label;
    protocols {
      bgp {
        export direct;
        group ce3 {
          neighbor 90.90.90.2 {
            peer-as 100;
          }
        }
      }
    }
  }
}
```

4. On Router PE5, configure the Layer 3 VPN routing instance (**L3VPN**) at the **[edit routing-instances]** hierarchy level. Also configure the BGP peer group at the **[edit routing-instances L3VPN protocols]** hierarchy level.

```
[edit ]
routing-instances {
  L3VPN {
    instance-type vrf;
    interface ge-2/0/0.0;
    route-distinguisher 65000:5;
    vrf-target target:65000:2;
    vrf-table-label;
    protocols {
      bgp {
        group ce5 {
          neighbor 80.80.80.2 {
            peer-as 200;
          }
        }
      }
    }
  }
}
```

Configuring the Route Reflector

Step-by-Step Procedure

Although a route reflector is not required to interconnect a Layer 2 circuit with a Layer 3 VPN, this examples uses a route reflector. This procedure shows the relevant portion of the route reflector configuration.

1. Configure the route reflector with RSVP, MPLS, BGP and OSPF. The route reflector is a BGP peer with the PE routers. Notice that the BGP peer group configuration includes the **family** statement and specifies the **inet-vpn** option The **inet-vpn** option enables BGP to advertise network layer reachability information (NLRI) for the Layer 3 VPN routes. The configuration also includes the **family** statement and specifies the **l2vpn** option. The **l2vpn** option enables BGP to advertise NLRI for the Layer 2 circuit. Layer 2 circuits use the same internal BGP infrastructure as Layer 2 VPNs.

```
[edit ]
protocols {
  rsvp {
    interface all;
    interface fxp0.0 {
      disable;
    }
  }
  mpls {
    label-switched-path to-pe3 {
      to 3.3.3.3;
    }
    label-switched-path to-pe5 {
      to 5.5.5.5;
    }
    interface all;
    interface fxp0.0 {
      disable;
    }
  }
  bgp {
    group RR {
      type internal;
      local-address 7.7.7.7;
      family inet {
        unicast;
      }
      family inet-vpn {
        unicast;
      }
      family l2vpn {
        signaling;
      }
      cluster 7.7.7.7;
      neighbor 1.1.1.1;
      neighbor 2.2.2.2;
      neighbor 4.4.4.4;
      neighbor 5.5.5.5;
      neighbor 3.3.3.3;
    }
```

```
    }
    ospf {
      traffic-engineering;
      area 0.0.0.0 {
        interface all;
        interface fxp0.0 {
          disable;
        }
      }
    }
}
```

Interconnecting the Layer 2 Circuit with the Layer 3 VPN

Step-by-Step Procedure

Before you can configure the logical tunnel interface in an MX Series router, you must create the tunnel services interface to be used for tunnel services.

1. Create the tunnel service interface on Router PE3. Include the **bandwidth** statement at the **[edit chassis fpc *slot-number* pic *slot-number* tunnel-services]** hierarchy level and specify the amount of bandwidth to reserve for tunnel services in gigabits per second.

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

2. On Router PE3, configure the **lt-1/1/10** logical tunnel interface unit 0.

   Router PE3 is the router that is *stitching* the Layer 2 circuit to the Layer 3 VPN using the logical tunnel interface. The configuration of the peer unit interfaces is what makes the interconnection.

   Include the **encapsulation** statement and specify the **ethernet-ccc** option. Include the **peer-unit** statement and specify the logical interface unit **1** as the peer tunnel interface. Include the **family** statement and specify the **ccc** option.

   Configure the **lt-1/1/10** logical interface unit **1** with **ethernet** encapsulation. Include the **peer-unit** statement and specify the logical interface unit **0** as the peer tunnel interface. Include the **family** statement and specify the **inet** option. Also include the **address** statement and specify **70.70.70.1/24** as the IPv4 address of the interface.

NOTE: The peering logical interfaces must belong to the same logical tunnel interface derived from the Tunnel Services PIC.

```
[edit interfaces]
lt-1/1/10 {
  unit 0 {
```

```
    encapsulation ethernet-ccc;
    peer-unit 1;
    family ccc;
  }
  unit 1 {
    encapsulation ethernet;
    peer-unit 0;
    family inet {
      address 70.70.70.1/24;
    }
  }
}
```

3. On each router, commit the configuration.

```
user@host> commit check
configuration check succeeds
user@host> commit
```

Verifying the Layer 2 Circuit to Layer 3 VPN Interconnection

To verify that the interconnection is working properly, perform these tasks:

Verifying That the Layer 2 Circuit Connection to Router PE3 is Up

Purpose To verify that the Layer 2 circuit connection from Router PE2 to Router PE3 is **Up**. To also document the incoming and outgoing LDP labels and the circuit ID used by this Layer 2 circuit connection.

Action Verify that the Layer 2 circuit connection is up, using the **show l2circuit connections** command.

```
user@PE2> show l2circuit connections

Legend for connection status (St)
EI -- encapsulation invalid      NP -- interface h/w not present
MM -- mtu mismatch               Dn -- down
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
```

```
VM -- vlan id mismatch           CF -- Call admission control failure
OL -- no outgoing label          IB -- TDM incompatible bitrate
NC -- intf encaps not CCC/TCC    TM -- TDM misconfiguration
BK -- Backup Connection          ST -- Standby Connection
CB -- rcvd cell-bundle size bad  SP -- Static Pseudowire
LD -- local site signaled down   RS -- remote site standby
RD -- remote site signaled down  XX -- unknown

Legend for interface status
Up -- operational
Dn -- down
Neighbor: 3.3.3.3
    Interface                Type  St     Time last up          # Up trans
    ge-1/0/2.0(vc 100)       rmt   Up     Jan  7 02:14:13 2010           1
      Remote PE: 3.3.3.3, Negotiated control-word: No
      Incoming label: 301488, Outgoing label: 315264
      Negotiated PW status TLV: No
      Local interface: ge-1/0/2.0, Status: Up, Encapsulation: ETHERNET
```

Meaning The output shows that the Layer 2 circuit connection from Router PE2 to Router PE3 is **Up** and the connection is using the **ge-1/0/2.0** interface. Note that the outgoing label is **315264** and the incoming label is **301488**, the virtual circuit (VC) identifier is **100** and the encapsulation is **ETHERNET**.

Verifying LDP Neighbors and Targeted LDP LSPs on Router PE2

Purpose To verify that Router PE2 has a targeted LDP LSP to Router PE3 and that Router PE2 and Router PE3 are LDP neighbors.

Action Verify that Router PE2 has a targeted LDP LSP to Router PE3 and that Router PE2 and Router PE3 are LDP neighbors, using the **show ldp neighbor** command.

```
user@PE2> show ldp neighbor
Address            Interface          Label space ID         Hold time
3.3.3.3            lo0.0              3.3.3.3:0                38
```

Meaning The output shows that Router PE2 has an LDP neighbor with the IPv4 address of **3.3.3.3**. Address 3.3.3.3 is the lo0.0 interface address of Router PE3. Notice that Router PE2 uses the local **lo0.0** interface for the LSP.

Verifying that the routers are LDP neighbors also verifies that the targeted LSP is established.

Verifying the Layer 2 Circuit Routes on Router PE2

Purpose To verify that Router PE2 has a route for the Layer 2 circuit and that the route uses the LDP MPLS label to Router PE3.

Action Verify that Router PE2 has a route for the Layer 2 circuit and that the route uses the LDP MPLS label to Router PE3, using the **show route table mpls.0** command.

```
user@PE2> show route table mpls.0
mpls.0: 13 destinations, 13 routes (13 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
0                  *[MPLS/0] 1w3d 05:24:11, metric 1
                      Receive
1                  *[MPLS/0] 1w3d 05:24:11, metric 1
                      Receive
2                  *[MPLS/0] 1w3d 05:24:11, metric 1
                      Receive
300560             *[LDP/9] 16:12:23, metric 1
                    > to 10.10.2.1 via xe-0/1/0.0, Pop
300560(S=0)        *[LDP/9] 16:12:23, metric 1
                    > to 10.10.2.1 via xe-0/1/0.0, Pop
301008             *[LDP/9] 16:12:23, metric 1
                    > to 10.10.4.2 via xe-0/3/0.0, Swap 299856
301488             *[L2CKT/7] 11:07:28
                    > via ge-1/0/2.0, Pop
301536             *[LDP/9] 16:12:23, metric 1
                    > to 10.10.4.2 via xe-0/3/0.0, Pop
301536(S=0)        *[LDP/9] 16:12:23, metric 1
                    > to 10.10.4.2 via xe-0/3/0.0, Pop
301712             *[LDP/9] 12:41:22, metric 1
                    > to 10.10.5.2 via xe-0/2/0.0, Swap 315184
301728             *[LDP/9] 12:41:22, metric 1
                    > to 10.10.5.2 via xe-0/2/0.0, Pop
301728(S=0)        *[LDP/9] 12:41:22, metric 1
                    > to 10.10.5.2 via xe-0/2/0.0, Pop
ge-1/0/2.0        *[L2CKT/7] 11:07:28, metric2 1
                    > to 10.10.5.2 via xe-0/2/0.0, Push 315264
```

Meaning The output shows that Router PE2 pushes the **315264** outgoing label on the **L2CKT** route going out interface **ge-1/0/2.0**. The output also shows that Router PE2 pops the **301488** incoming label on the **L2CKT** coming from interface **ge-1/0/2.0**

Verifying That the Layer 2 Circuit Connection to Router PE2 is Up

Purpose To verify that the Layer 2 circuit connection from Router PE3 to Router PE2 is **Up**, To also document the incoming and outgoing LDP labels and the circuit ID used by this Layer 2 circuit connection.

Action Verify that the Layer 2 circuit connection is up, using the **show l2circuit connections** command.

```
user@PE3> show l2circuit connections

Layer-2 Circuit Connections:
Legend for connection status (St)
EI -- encapsulation invalid      NP -- interface h/w not present
MM -- mtu mismatch               Dn -- down
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
VM -- vlan id mismatch           CF -- Call admission control failure
OL -- no outgoing label          IB -- TDM incompatible bitrate
NC -- intf encaps not CCC/TCC    TM -- TDM misconfiguration
BK -- Backup Connection          ST -- Standby Connection
CB -- rcvd cell-bundle size bad  XX -- unknown

Legend for interface status
Up -- operational
Dn -- down
Neighbor: 2.2.2.2
```

```
Interface               Type  St     Time last up          # Up trans
lt-1/1/10.0(vc 100)     rmt   Up     Jan  7 02:15:03 2010           1
  Remote PE: 2.2.2.2, Negotiated control-word: No
  Incoming label: 315264, Outgoing label: 301488
  Local interface: lt-1/1/10.0, Status: Up, Encapsulation: ETHERNET
```

Meaning The output shows that the Layer 2 circuit connection from Router PE3 to Router PE2 is **Up** and the connection is using the logical tunnel (**lt**) interface. Note that the incoming label is **315264** and the outgoing label is **301488**, the virtual circuit (VC) identifier is **100**, and that the encapsulation is **ETHERNET**.

Verifying LDP Neighbors and Targeted LDP LSPs on Router PE3

Purpose To verify that Router PE3 has a targeted LDP LSP to Router PE2 and that Router PE3 and Router PE2 are LDP neighbors.

Action Verify that Router PE2 has a targeted LDP LSP to Router PE3 and that Router PE2 and Router PE3 are LDP neighbors, using the **show ldp neighbor** command.

```
user@PE2> show ldp neighbor
Address            Interface          Label space ID         Hold time
2.2.2.2            lo0.0              2.2.2.2:0                43
4.4.4.4            lo0.0              4.4.4.4:0                33
```

Meaning The output shows that Router PE3 has an LDP neighbor with the IPv4 address of **2.2.2.2**. Address 2.2.2.2 is the lo0.0 interface address of Router PE2. The output also shows that the interface used on Router PE3 for the LSP is **lo0.0**. Verifying that the routers are LDP neighbors also verifies that the targeted LSP is established.

Verifying a BGP Peer Session with the Route Reflector on Router PE3

Purpose To verify that Router PE3 has a peer session established with the route reflector.

Action Verify that Router PE3 has a peer session established with the route reflector, using the **show bgp summary** command.

```
user@PE2> show bgp summary

Groups: 2 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l3vpn.0            1          1          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
7.7.7.7               65000       1597       1612       0       1    12:03:21 Establ
  bgp.l2vpn.0: 0/0/0/0
  bgp.l3vpn.0: 1/1/1/0
  L3VPN.inet.0: 1/1/1/0
```

Meaning The output shows that Router PE3 has a peer session with the router with the IPv4 address of **7.7.7.7**. Address 7.7.7.7 is the lo0.0 interface address of the route reflector. The output also shows that the peer session state is **Establ**, meaning that the session is established.

Verifying the Layer 3 VPN Routes on Router PE3

Purpose To verify that Router PE3 has Layer 3 VPN routes to Router CE2, Router CE3, and Router CE5.

Action Verify that Router PE3 has routes to Router CE2, Router CE3, and Router CE5 in the Layer 3 VPN route table, using the **show route table L3VPN.inet.0** command. In this example, **L3VPN** is the name configured for the routing instance.

```
user@PE3> show route table L3VPN.inet.0
L3VPN.inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

70.70.70.0/24      *[Direct/0] 11:13:59
                    > via lt-1/1/10.1
70.70.70.1/32      *[Local/0] 11:13:59
                      Local via lt-1/1/10.1
80.80.80.0/24      *[BGP/170] 11:00:41, localpref 100, from 7.7.7.7
                      AS path: I
                    > to 10.10.6.2 via xe-2/1/0.0, Push 16
90.90.90.0/24      *[Direct/0] 11:54:41
                    > via ge-1/0/1.0
90.90.90.1/32      *[Local/0] 11:54:41
                      Local via ge-1/0/1.0
```

Meaning The output shows that Router PE3 has a route to the IPv4 subnetwork address of **70.70.70.0**. Address 70.70.70.2 is the interface address of Router CE2. The output shows that Router PE3 has a route to the IPv4 subnetwork address of **80.80.80.0**. Address 80.80.80.2 is the interface address of Router CE5. The output shows that Router PE3 has a route to the IPv4 subnetwork address of **90.90.90.0**. Address 90.90.90.2 is the interface address of Router CE3.

Verifying the Layer 2 Circuit Routes on Router PE3

Purpose To verify that Router PE3 has a route to Router PE2 in the Layer 2 circuit route table.

Action Verify that Router PE3 has a route to Router PE2 in the Layer 2 circuit route table, using the **show route table l2circuit.0** command.

```
user@PE3> show route table l2circuit.0
2.2.2.2:NoCtrlWord:5:100:Local/96 (1 entry, 1 announced)
        *L2CKT  Preference: 7
                Next hop type: Indirect
                Next-hop reference count: 1
                Next hop type: Router
                Next hop: 10.10.5.1 via xe-2/2/0.0, selected
                Protocol next hop: 2.2.2.2
                Indirect next hop: 8cae0a0 -
                State: <Active Int>
                Local AS: 65000
                Age: 11:16:50   Metric2: 1
                Task: l2 circuit
                Announcement bits (1): 0-LDP
```

```
                    AS path: I
                    VC Label 315264, MTU 1500
```

Meaning The output shows that Router PE3 has a route to the IPv4 address of **2.2.2.2**. Address 2.2.2.2 is the lo0.0 interface address of Router PE2. Note that the VC label is **315264**. This label is the same as the incoming MPLS label displayed using the **show l2circuit connections** command.

Verifying the MPLS Routes on Router PE3

Purpose To verify that Router PE3 has a route to Router PE2 in the MPLS route table.

Action Verify Router PE3 has a route to Router PE2 in the MPLS route table, using the **show route table mpls.0** command.

```
user@PE3> show route table mpls.0
mpls.0: 21 destinations, 21 routes (21 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 1w3d 05:29:02, metric 1
                      Receive
1                  *[MPLS/0] 1w3d 05:29:02, metric 1
                      Receive
2                  *[MPLS/0] 1w3d 05:29:02, metric 1
                      Receive
16                 *[VPN/0] 12:22:45
                      to table L3VPN.inet.0, Pop
315184             *[LDP/9] 12:45:14, metric 1
                    > to 10.10.20.1 via xe-2/0/0.0, Pop
315184(S=0)        *[LDP/9] 12:45:14, metric 1
                    > to 10.10.20.1 via xe-2/0/0.0, Pop
315200             *[LDP/9] 00:03:53, metric 1
                    > to 10.10.20.1 via xe-2/0/0.0, Swap 625297
                      to 10.10.6.2 via xe-2/1/0.0, Swap 299856
315216             *[LDP/9] 12:45:14, metric 1
                    > to 10.10.6.2 via xe-2/1/0.0, Pop
315216(S=0)        *[LDP/9] 12:45:14, metric 1
                    > to 10.10.6.2 via xe-2/1/0.0, Pop
315232             *[LDP/9] 12:45:06, metric 1
                    > to 10.10.1.1 via xe-2/3/0.0, Pop
315232(S=0)        *[LDP/9] 12:45:06, metric 1
                    > to 10.10.1.1 via xe-2/3/0.0, Pop
315248             *[LDP/9] 12:45:14, metric 1
                    > to 10.10.5.1 via xe-2/2/0.0, Pop
315248(S=0)        *[LDP/9] 12:45:14, metric 1
                    > to 10.10.5.1 via xe-2/2/0.0, Pop
315264             *[L2CKT/7] 11:11:20
                    > via lt-1/1/10.0, Pop
315312             *[RSVP/7] 11:26:01, metric 1
                    > to 10.10.6.2 via xe-2/1/0.0, label-switched-path to-pe5
315312(S=0)        *[RSVP/7] 11:26:01, metric 1
                    > to 10.10.6.2 via xe-2/1/0.0, label-switched-path to-pe5
315328             *[RSVP/7] 11:26:01, metric 1
                    > to 10.10.20.1 via xe-2/0/0.0, label-switched-path to-RR
315360             *[RSVP/7] 11:26:01, metric 1
                    > to 10.10.20.1 via xe-2/0/0.0, label-switched-path to-RR
316208             *[RSVP/7] 00:03:32, metric 1
                    > to 10.10.6.2 via xe-2/1/0.0, label-switched-path
```

```
Bypass->10.10.9.1
316208(S=0)        *[RSVP/7] 00:03:32, metric 1
                    > to 10.10.6.2 via xe-2/1/0.0, label-switched-path
Bypass->10.10.9.1
lt-1/1/10.0        *[L2CKT/7] 11:11:20, metric2 1
                    > to 10.10.5.1 via xe-2/2/0.0, Push 301488
```

Meaning The output shows that Router PE3 has a route for the Layer 2 circuit and that the route uses the LDP MPLS label to Router PE2. Notice that the **301488** label is the same as the outgoing label displayed on Router PE2 using the **show l2circuit connections** command.

Verifying Traffic Flow Between Router CE2 and Router CE3

Purpose To verify that the CE routers can send and receive traffic across the interconnection.

Action Verify that Router CE2 can send traffic to and receive traffic from Router CE3 across the interconnection, using the **ping** command.

```
user@CE2>ping 90.90.90.2
PING 90.90.90.2 (90.90.90.2): 56 data bytes
64 bytes from 90.90.90.2: icmp_seq=0 ttl=63 time=0.708 ms
64 bytes from 90.90.90.2: icmp_seq=1 ttl=63 time=0.610 ms
```

Meaning The output shows that Router CE2 can send an ICMP request to and receive a response from Router CE3 across the interconnection.

Verifying Traffic Flow Between Router CE2 and Router CE5

Purpose To verify that the CE routers can send and receive traffic across the interconnection.

Action Verify that Router CE2 can send traffic to and receive traffic from Router CE5 across the interconnection, using the **ping** command.

```
user@CE2>ping 80.80.80.2
PING 80.80.80.2 (80.80.80.2): 56 data bytes
64 bytes from 80.80.80.2: icmp_seq=0 ttl=62 time=0.995 ms
64 bytes from 80.80.80.2: icmp_seq=1 ttl=62 time=1.005 ms
```

Meaning The output shows that Router CE2 can send an ICMP request to and receive a response from Router CE5 across the interconnection.

Related Documentation

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

PART 3

Administration

CHAPTER 6

Layer 2 VPNs Reference

Supported Layer 2 VPN Standard

The Junos OS substantially supports Internet draft draft-kompella-ppvpn-l2vpn-03.txt, *Layer 2 VPNs Over Tunnels*.

Related Documentation

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

CHAPTER 7

Summary of Layer 2 VPNs Configuration Statements

control-channel

Syntax

```
control-channel {
    pwe3-control-word;
    pw-label-ttl-1;
    router-alert-label;
}
```

Hierarchy Level

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

Release Information

Statement introduced in Junos OS Release 10.0.

Description

Configure the Virtual Circuit Connection Verification (VCCV) BFD control channel. VCCV provides a control channel associated with a pseudowire. You can configure a number of different CV types for this control channel, based on the configuration of the pseudowire.

Options

pwe3-control-word—For BGP-based pseudowires that send OAM packets with a control word that has 0001b as the first nibble.

pw-label-ttl-1—For BGP-based pseudowires that send OAM packets with a router alert label.

router-alert-label—For BGP-based pseudowires that send OAM packets with the MPLS pseudowire label, time-to-live (TTL), set to 1.

Required Privilege Level

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

Related Documentation

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

control-word

Syntax (control-word | no-control-word);

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

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify the control word. The control word is 4 bytes long and is inserted between the Layer 2 protocol data unit (PDU) being transported and the virtual connection (VC) label that is used for demultiplexing.

- **control-word**—Enables the use of the control word.
- **no-control-word**—Disables the use of the control word.

Default The control word is enabled by default. You can also configure the control word explicitly using the **control-word** statement.

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

Related Documentation
- Disabling the Control Word for Layer 2 VPNs on page 19

description

Syntax description *text*;

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

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 11.1 for EX Series switches.

Description Describe the VPN or virtual private LAN service (VPLS) routing instance.

Options *text*—Provide a text description. If the text includes one or more spaces, enclose it in quotation marks (" "). Any descriptive text you include is displayed in the output of the **show route instance detail** command and has no effect on operation.

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

Related Documentation
- Configuring the Site on page 12
- Configuring an MPLS-Based Layer 2 VPN (CLI Procedure)

encapsulation (Logical Interface)

Syntax encapsulation (atm-ccc-cell-relay | atm-ccc-vc-mux | atm-cisco-nlpid | atm-mlppp-llc | atm-nlpid | atm-ppp-llc | atm-ppp-vc-mux | atm-snap | atm-tcc-snap | atm-tcc-vc-mux | atm-vc-mux | ether-over-atm-llc | ethernet-vpls-fr | ether-vpls-over-atm-llc | ethernet-vpls-ppp | ethernet | frame-relay-ccc | frame-relay-ppp | frame-relay-tcc | multilink-frame-relay-end-to-end | multilink-ppp | ppp-over-ether | ppp-over-ether-over-atm-llc | vlan-ccc | vlan-tcc | vlan-vpls);

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

Release Information Statement introduced before Junos OS Release 7.4.

Description Logical link-layer encapsulation type.

Options **atm-ccc-cell-relay**—Use Asynchronous Transfer Mode (ATM) cell relay encapsulation.

atm-ccc-vc-mux—Use ATM VC multiplex encapsulation on circuit cross-connect (CCC) circuits. When you use this encapsulation type, you can configure the family **ccc** only.

atm-cisco-nlpid—Use Cisco ATM Network Layer Protocol identifier (NLPID) encapsulation. When you use this encapsulation type, you can configure the family **inet** only.

atm-mlppp-llc—For ATM2 intelligent queuing (IQ) interfaces only, use Multilink Point-to-Point (MLPPP) over ATM adaptation layer 5 (AAL5) logical link control (LLC). For this encapsulation type, your routing platform must be equipped with a Link Services or Voice Services Physical Interface Card (PIC).

atm-nlpid—Use ATM NLPID encapsulation. When you use this encapsulation type, you can configure the family **inet** only.

atm-ppp-llc—For ATM2 IQ interfaces only, use Point-to-Point Protocol (PPP) over AAL5 logical link control (LLC) encapsulation.

atm-ppp-vc-mux—For ATM2 IQ interfaces only, use PPP over AAL5 multiplex encapsulation.

atm-snap—Use ATM Subnetwork Access Protocol (SNAP) encapsulation.

atm-tcc-snap—Use ATM SNAP encapsulation on translational cross-connect (TCC) circuits.

atm-tcc-vc-mux—Use ATM VC multiplex encapsulation on TCC circuits. When you use this encapsulation type, you can configure the family **tcc** only.

atm-vc-mux—Use ATM VC multiplex encapsulation. When you use this encapsulation type, you can configure the family **inet** only.

ether-over-atm-llc—For interfaces that carry IP version 4 (IPv4) traffic, use Ethernet over ATM LLC encapsulation. When you use this encapsulation type, you cannot configure multipoint interfaces.

ethernet-vpls-fr—Use in a VPLS setup when a CE device is connected to a PE device over a time division multiplexing (TDM) link. This encapsulation type enables the PE device to terminate the outer layer 2 Frame Relay connection, use the 802.1p bits inside the inner Ethernet header to classify the packets, look at the MAC address from the Ethernet header, and use it to forward the packet into a given VPLS instance.

ether-vpls-over-atm-llc—For ATM2 IQ interfaces only, use the Ethernet VPLS over ATM LLC encapsulation to bridge Ethernet interfaces and ATM interfaces over a VPLS routing instance (as described in RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*). Packets from the ATM interfaces are converted to standard ENET2/802.3 encapsulated Ethernet frames with the frame check sequence (FCS) field removed.

ethernet-vpls-ppp—Use in a VPLS setup when a CE device is connected to a PE device over a time division multiplexing (TDM) link. This encapsulation type enables the PE device to terminate the outer layer 2 PPP connection, use the 802.1p bits inside the inner Ethernet header to classify the packets, look at the MAC address from the Ethernet header, and use it to forward the packet into a given VPLS instance.

ethernet—Use Ethernet II encapsulation (as described in RFC 894, *A Standard For The Transmission Of IP Datagrams Over Ethernet Networks*).

frame-relay-ccc—Use Frame Relay encapsulation on CCC circuits. When you use this encapsulation type, you can configure the family **ccc** only.

frame-relay-ppp—Use Frame Relay encapsulation on PPP circuits.

frame-relay-tcc—Use Frame Relay encapsulation on TCC circuits for connecting unlike media. When you use this encapsulation type, you can configure the family **tcc** only.

multilink-frame-relay-end-to-end—Use Multilink Frame Relay (MLFR) FRF.15 encapsulation. This encapsulation is used only on multilink, link services, and voice services interfaces and their constituent T1 or E1 interfaces.

multilink-ppp—Use MLPPP encapsulation. This encapsulation is used only on multilink, link services, and voice services interfaces and their constituent T1 or E1 interfaces.

ppp-over-ether—For underlying Ethernet interfaces on Juniper Networks J Series Services Routers only, use PPP over Ethernet encapsulation. When you use this encapsulation type, you cannot configure the interface address. Instead you configure the interface address on the PPP interface. For more information, see the *Junos OS Interfaces and Routing Configuration Guide*.

ppp-over-ether-over-atm-llc—For underlying ATM interfaces on J Series Services Routers only, use PPP over Ethernet over ATM LLC encapsulation. When you use this encapsulation type, you cannot configure the interface address. Instead you configure the interface address on the PPP interface. For more information, see the *Junos OS Interfaces and Routing Configuration Guide*.

vlan-ccc—Use Ethernet virtual LAN (VLAN) encapsulation on CCC circuits. When you use this encapsulation type, you can configure the family **ccc** only.

vlan-tcc—Use Ethernet VLAN encapsulation on TCC circuits. When you use this encapsulation type, you can configure the family **tcc** only.

vlan-vpls—Use Ethernet VLAN encapsulation on virtual private LAN service (VPLS) circuits.

Required Privilege Level

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

Related Documentation

- Configuring CCC Encapsulation for Layer 2 VPNs on page 16
- Configuring TCC Encapsulation for Layer 2 VPNs and Layer 2 Circuits on page 17

encapsulation (Physical Interface)

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

Hierarchy Level [edit interfaces *interface-name*],
[edit logical-systems *logical-system-name* interfaces *interface-name*]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 11.1 for EX Series switches.

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

Default PPP encapsulation.

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

atm-pvc—Use ATM permanent virtual connection (PVC) encapsulation.

cisco-hdlc—Use Cisco-compatible HDLC framing.

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

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

ethernet-ccc—Use Ethernet CCC encapsulation on Ethernet interfaces that must accept packets carrying standard Tag Protocol ID (TPID) values. For example, Ethernet CCC encapsulation can be used to transparently transport any VLANs or other Ethernet frames entering a port across a Layer 2 circuit.

ethernet-over-atm—For interfaces that carry IPv4 traffic, use Ethernet over ATM encapsulation. When you use this encapsulation type, you cannot configure multipoint interfaces. As defined in RFC 1483 *Multiprotocol Encapsulation over ATM Adaptation Layer 5*, this encapsulation type allows ATM interfaces to connect to devices that support only bridged-mode protocol data units (BPDUs). The Junos OS does not completely support bridging, but accepts BPDU packets as a default gateway. If you use the router as an edge device, then the router acts as a default gateway. It accepts Ethernet LLC/SNAP frames with IP or Address Resolution Protocol (ARP) in the payload and drops the rest. For packets destined for the Ethernet LAN, a route lookup is done using the destination IP address. If the route lookup yields a full address match, the packet is encapsulated with an LLC/SNAP and media access control (MAC) header and forwarded to the ATM interface.

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

ethernet-vpls—Use Ethernet VPLS encapsulation on Ethernet interfaces that have VPLS enabled and that must accept packets carrying standard TPID values.

ethernet-vpls-fr—Use in a VPLS setup when a CE device is connected to a PE device over a time division multiplexing (TDM) link. This encapsulation type enables the PE device to terminate the outer layer 2 Frame Relay connection, use the 802.1p bits inside the inner Ethernet header to classify the packets, look at the MAC address from the Ethernet header, and use it to forward the packet into a given VPLS instance.

ethernet-vpls-ppp—Use in a VPLS setup when a CE device is connected to a PE device over a time division multiplexing (TDM) link. This encapsulation type enables the PE device to terminate the outer layer 2 PPP connection, use the 802.1p bits inside the inner Ethernet header to classify the packets, look at the MAC address from the Ethernet header, and use it to forward the packet into a given VPLS instance.

extended-frame-relay-ccc—Use Frame Relay encapsulation on CCC circuits. This encapsulation type allows you to dedicate data link connection identifiers (DLCIs) 1 through 1022 to CCC.

extended-frame-relay-tcc—Use Frame Relay encapsulation on TCC circuits to connect unlike media. This encapsulation type allows you to dedicate DLCIs 1 through 1022 to TCC.

extended-vlan-ccc—Use extended VLAN encapsulation on CCC circuits with Gigabit Ethernet and 4-port Fast Ethernet interfaces that must accept packets carrying 802.1Q values.

extended-vlan-tcc—For interfaces that carry IPv4 traffic, use extended VLAN encapsulation on TCC circuits with Gigabit Ethernet interfaces on which you want to use 802.1Q tagging. Extended Ethernet TCC is not currently supported on Fast Ethernet 48-port PICs.

extended-vlan-vpls—Use extended VLAN VPLS encapsulation on Ethernet interfaces that have VLAN 802.1Q tagging and VPLS enabled and that must accept packets carrying TPIDs 0x8100, 0x9100, and 0x9901.

flexible-ethernet-services—For Gigabit Ethernet IQ interfaces and Gigabit Ethernet PICs with small form-factor pluggable transceivers (SFPs) only, use flexible Ethernet services encapsulation when you want to configure multiple per-unit Ethernet encapsulations. This encapsulation type allows you to configure any combination of route, TCC, CCC, and VPLS encapsulations on a single physical port. Aggregated Ethernet bundles cannot use this encapsulation type. If you configure flexible Ethernet services encapsulation on the physical interface, VLAN IDs from 1 through 511 are no longer reserved for normal VLANs.

flexible-frame-relay—For IQ interfaces only, use flexible Frame Relay encapsulation when you want to configure multiple per-unit Frame Relay encapsulations. This encapsulation type allows you to configure any combination of TCC, CCC, and standard Frame Relay encapsulations on a single physical port. Also, each logical interface can have any DLCI value from 1 through 1022.

frame-relay—Use Frame Relay encapsulation.

frame-relay-ccc—Use Frame Relay encapsulation or Frame Relay encapsulation on CCC circuits.

frame-relay-port-ccc—Use Frame Relay port CCC encapsulation to transparently carry all the DLCIs between two CE routers without explicitly configuring each DLCI on the two provider edge (PE) routers with Frame Relay transport. When you use this encapsulation type, you can configure the family **ccc** only.

frame-relay-tcc—Use Frame Relay encapsulation on TCC circuits to connect unlike media.

multilink-frame-relay-uni-nni—Use MLFR user-to-network interface (UNI) network-to-network interface (NNI) encapsulation. This encapsulation is used only on link services and voice services interfaces functioning as FRF.16 bundles and their constituent T1 or E1 interfaces.

ppp—Use serial PPP encapsulation.

ppp-ccc—Use serial PPP encapsulation on CCC circuits. When you use this encapsulation type, you can configure the family **ccc** only.

ppp-tcc—Use serial PPP encapsulation on TCC circuits for connecting unlike media. When you use this encapsulation type, you can configure the family **tcc** only.

vlan-ccc—Use Ethernet VLAN encapsulation on CCC circuits.

vlan-vpls—Use VLAN VPLS encapsulation on Ethernet interfaces with VLAN tagging and VPLS enabled. Interfaces with VLAN VPLS encapsulation accept packets carrying standard TPID values only.

Required Privilege Level

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

Related Documentation

- Configuring CCC Encapsulation for Layer 2 VPNs on page 16
- Configuring TCC Encapsulation for Layer 2 VPNs and Layer 2 Circuits on page 17
- Configuring an MPLS-Based Layer 2 VPN (CLI Procedure)

encapsulation-type

Syntax encapsulation-type (atm-aal5 | atm-cell | atm-cell-port-mode | atm-cell-vc-mode | atm-cell-vp-mode | cesop | cisco-hdlc | ethernet | ethernet-vlan | frame-relay | frame-relay-port-mode | interworking | ppp | satop-e1 | satop-e3 | satop-t1 | satop-t3);

Hierarchy Level [edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls],
[edit protocols l2circuit neighbor *address* interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols l2vpn],
[edit routing-instances *routing-instance-name* protocols l2vpn neighbor *address*],
[edit routing-instances *routing-instance-name* protocols vpls]

Release Information Statement introduced in Junos OS Release 9.2.
Statement introduced in Junos OS Release 11.1 for EX Series switches.

Description Specify the type of Layer 2 traffic originating from the CE device. Only the **ethernet** and **ethernet-vlan** encapsulation types are supported for VPLS. Not all encapsulation types are supported on the switches. See the switch CLI.

Options **atm-aal5**—ATM Adaptation Layer (AAL/5)

atm-cell—ATM cell relay

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

atm-cell-vc-mode—ATM VC cell relay nonpromiscuous mode

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

cesop—CESOP-based Layer 2 VPN

cisco-hdlc—Cisco Systems–compatible HDLC

ethernet—Ethernet

ethernet-vlan—Ethernet VLAN

frame-relay—Frame Relay

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

interworking—Layer 2.5 interworking VPN

ppp—PPP

satsop-e1—SATSOP-E1–based Layer 2 VPN

satsop-e3—SATSOP-E3–based Layer 2 VPN

satsop-t1—SATSOP-T1–based Layer 2 VPN

satsop-t3—SATSOP-T3–based Layer 2 VPN

Default: For VPLS networks, the default encapsulation type is **ethernet**.

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

Related Documentation
- Configuring the Encapsulation Type on page 14
- Configuring VPLS Routing Instances
- Configuring Interfaces for Layer 2 Circuits
- Configuring an MPLS-Based Layer 2 VPN (CLI Procedure)

interface

Syntax

```
interface interface-name {
  description text;
  remote-site-id remote-site-id;
}
```

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

Release Information
Statement introduced before Junos OS Release 7.4.

Description
Configure an interface to handle traffic for a circuit configured for the Layer 2 VPN.

Options
interface-name—Name of the interface used for the Layer 2 VPN.

The remaining statements are explained separately.

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

Related Documentation
- Configuring the Site on page 12
- Configuring the Remote Site ID on page 12

l2vpn

Syntax

```
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;
    }
  }
}
```

Hierarchy Level

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

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 11.1 for EX Series switches.

Description

Enable a Layer 2 VPN routing instance on a PE router or switch.

The remaining statements are explained separately.

Required Privilege Level

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

Related Documentation

- Configuring a Layer 2 VPN Routing Instance on page 11
- Configuring an MPLS-Based Layer 2 VPN (CLI Procedure)

oam

Syntax

```
oam {
  ping-interval;
  bfd-liveness-detection {
    detection-time {
      threshold milliseconds;
    }
    minimum-interval milliseconds;
    minimum-receive-interval milliseconds;
    multiplier number;
    no-adaptation;
    transmit-interval {
      minimum-interval milliseconds;
      threshold milliseconds;
    }
    version bfd-protocol-version;
  }
  control-channel {
    pwe3–control-word;
    pseudowire-label-ttl-1;
    router-alert-label;
  }
}
```

Hierarchy Level

[edit routing-instances *routing-instance-name* protocols l2vpn],
[edit routing-instances *routing-instance-name* protocols vpls],
[edit routing-instances *routing-instance-name* protocols vpls neighbor *address*],
[edit protocols l2circuit neighbor *address* interface *interface-name*]

Release Information

Statement introduced in Junos OS Release 10.0.

Description

Allows you to configure bidirectional forwarding detection (BFD) and a control channel for a pseudowire, in addition to the corresponding operations and management functions to be used over that control channel. BFD provides a low resource fault detection mechanism for the continuous monitoring of the pseudowire data path and for detecting data plane failures. The **control-channel** statement is not applicable to Layer 2 circuit pseudowires.

Options

The **bfd-liveness-detection** statement and substatements are described in the *Junos OS Routing Protocols Configuration Guide*.

The other statements are explained separately.

Required Privilege Level

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

Related Documentation

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

policer

Syntax

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

Hierarchy Level

[edit interfaces *interface-name* unit *logical-unit-number* family (ccc | inet | tcc)],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family (ccc | inet | tcc)]

Release Information

Statement introduced before Junos OS Release 7.4.

Description

Use policing to control the amount of traffic flowing over the interfaces servicing a Layer 2 VPN.

Options

input *policer-template-name*—Name of one policer to evaluate when packets are received on the interface.

output *policer-template-name*—Name of one policer to evaluate when packets are transmitted on the interface.

Required Privilege Level

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

Related Documentation

- Configuring Traffic Policing in Layer 2 VPNs on page 18
- *Junos OS Policy Framework Configuration Guide*
- *Junos OS Network Interfaces Configuration Guide*

proxy

Syntax proxy inet-address *address*;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* family tcc],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family tcc]

Release Information Statement introduced before Junos OS Release 7.4.

Description For Layer 2.5 VPNs using an Ethernet interface as the TCC router, configure the IP address for which the TCC router is proxying. Ethernet TCC is supported on interfaces that carry IPv4 traffic only. Ethernet TCC encapsulation is supported on 1-port Gigabit Ethernet, 2-port Gigabit Ethernet, 4-port Gigabit Ethernet, and 4-port Fast Ethernet PICs only. Ethernet TCC is not supported on the T640 routing node.

Options **inet-address *address***—IP address for which the TCC router is acting as a proxy.

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

Related Documentation
- Configuring TCC Encapsulation for Layer 2 VPNs and Layer 2 Circuits on page 17

remote

Syntax remote (inet-address | mac-address) *address*;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* family tcc],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family tcc]

Release Information Statement introduced before Junos OS Release 7.4.

Description For Layer 2.5 VPNs employing an Ethernet interface as the TCC router, configure the location of the remote router. Ethernet TCC is supported on interfaces that carry IPv4 traffic only. Ethernet TCC encapsulation is supported on 1-port Gigabit Ethernet, 2-port Gigabit Ethernet, 4-port Gigabit Ethernet, and 4-port Fast Ethernet PICs only.

Options **inet-address*address***—The IP address of the remote site.

mac-address *address*—The MAC address of the remote site.

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

Related Documentation
- Configuring TCC Encapsulation for Layer 2 VPNs and Layer 2 Circuits on page 17

remote-site-id

Syntax remote-site-id *remote-site-ID*;

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

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 11.1 for EX Series switches.

Description Control the remote interface to which the interface should connect. If you do not explicitly configure the remote site ID, the order of the interfaces configured for the site determines the default value. This statement is optional.

Options ***remote-site-ID***—Identifier specifying the interface on the remote PE router the Layer 2 VPN routing instance connects to.

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

Related Documentation

- Configuring the Remote Site ID on page 12
- Configuring an MPLS-Based Layer 2 VPN (CLI Procedure)

site

Syntax

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

Hierarchy Level

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

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 11.1 for EX Series switches.

Description

Specify the site name, site identifier, and interfaces connecting to the site. Allows you to configure a remote site ID for remote sites.

Options

site-identifier *identifier*—Numerical identifier for the site used as a default reference for the remote site ID.

site-name—Name of the site.

The remaining statements are explained separately.

Required Privilege Level

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

Related Documentation

- Configuring the Site on page 12
- Configuring an MPLS-Based Layer 2 VPN (CLI Procedure)

site-identifier

Syntax site-identifier *identifier*;

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

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 11.1 for EX Series switches.

Description Specify the numerical identifier for the local Layer 2 VPN site.

Options ***identifier***—The numerical identifier for the Layer 2 VPN site, which can be any number from 1 through 65,534.

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

Related Documentation

- Configuring the Site on page 12
- Configuring an MPLS-Based Layer 2 VPN (CLI Procedure)

site-preference

Syntax

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

Hierarchy Level

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

Release Information

Statement introduced in Junos OS Release 9.1.

Description

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 VE identifier, the advertisement with the highest local preference value is preferred. You can use this statement to enable multihoming for Layer 2 VPNs.

Options

preference-value—Specify the preference value advertised for a Layer 2 VPN.

Range: 1 through 65,535

backup—Set the preference value to 1.

primary—Set the preference value to 65,535.

Required Privilege Level

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

Related Documentation

- Configuring a Site Preference and Layer 2 VPN Multihoming on page 14

traceoptions

Syntax

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

Hierarchy Level

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

Release Information

Statement introduced before Junos OS Release 7.4.

Description

Trace traffic flowing through a Layer 2 VPN.

Options

disable—(Optional) Disable the tracing operation. You can use this option to disable a single operation when you have defined a broad group of tracing operations, such as **all**.

file *filename*—Name of the file to receive the output of the tracing operation. Enclose the name in quotation marks (" ").

files *number*—(Optional) Maximum number of trace files. When a trace file named ***trace-file*** reaches its maximum size, it is renamed ***trace-file*.0**, then ***trace-file*.1**, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum number of files, you also must specify a maximum file size with the **size** option.

Range: 2 through 1000 files

Default: 2 files

flag *flag*—Tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements.

- **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

flag-modifier—(Optional) Modifier for the tracing flag. You can specify the following modifier:

- **detail**—Provide detailed trace information
- **receive**—Trace received packets
- **send**—Trace transmitted packets

no-world-readable—(Optional) Prevents any user from reading the trace file.

size *size*—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named ***trace-file*** reaches this size, it is renamed ***trace-file*.0**. When ***trace-file*** again reaches its maximum size, ***trace-file*.0** is renamed ***trace-file*.1** and ***trace-file*** is renamed ***trace-file*.0**. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum file size, you also must specify a maximum number of trace files with the **files** option.

Syntax: ***x*k** to specify kilobytes, ***x*m** to specify megabytes, or ***x*g** to specify gigabytes

Range: 10 KB through the maximum file size supported on your system

Default: 1 MB

world-readable—(Optional) Allow any user to read the trace file.

Default: The default is **no-world-readable**.

Required Privilege Level

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

Related Documentation

- Tracing Layer 2 VPN Traffic and Operations on page 15

PART 4

Index

Index

Symbols

B

C

D

E

F

I

L

M

N

O

P

R

S

T

V