JUNIPER NETWORKS

Junos® OS

Layer 2 Circuits 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 Circuits 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

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

Part 1 Overview

Chapter 1 **Introduction to Layer 2 Circuits 3**

Layer 2 Circuit Overview 3
Layer 2 Circuit Bandwidth Accounting and Call Admission Control 4
Bandwidth Accounting and Call Admission Control Overview 4
Selecting an LSP Based on the Bandwidth Constraint 4
LSP Path Protection and CAC 5
Secondary Paths and CAC 5
Fast Reroute and CAC 6
Link and Node Protection and CAC 6
Layer 2 Circuits Trunk Mode 6
Egress Protection LSPs for Layer 2 Circuits 7

Chapter 2 **Introduction to Configuring Layer 2 Circuits 9**

Configuring LDP for Layer 2 Circuits 9

Part 2 Configuration

Chapter 3 **Configuring Layer 2 Circuits 13**

Introduction to Configuring Layer 2 Circuits 13
Configuring Local Interface Switching in Layer 2 Circuits 14
Configuring the Interfaces for the Local Interface Switch 15
Enabling Local Interface Switching When the MTU Does Not Match 15
Configuring Interfaces for Layer 2 Circuits 16
Configuring the Address for the Neighbor of the Layer 2 Circuit 16
Configuring the Neighbor Interface for the Layer 2 Circuit 16
Configuring a Community for the Layer 2 Circuit 17
Configuring the Control Word for Layer 2 Circuits 17
Configuring the Encapsulation Type for the Layer 2 Circuit Neighbor Interface 19

Enabling the Layer 2 Circuit When the Encapsulation Does Not Match . . . 19
Configuring the MTU for the Layer 2 Circuit Neighbor Interface . . . 20
Configuring the Protect Interface . . . 21
Configuring the Protect Interface From Switching Over to the Primary Interface . . . 21
Configuring the Pseudowire Status TLV . . . 21
Configuring Layer 2 Circuits over Both RSVP and LDP LSPs . . . 22
Configuring the Virtual Circuit ID . . . 23
Configuring the Interface Encapsulation Type for Layer 2 Circuits . . . 23
Configuring ATM2 IQ Interfaces for Layer 2 Circuits . . . 24
Configuring Static Layer 2 Circuits . . . 24
Configuring Policies for Layer 2 Circuits . . . 25
Configuring the Layer 2 Circuit Community . . . 25
Configuring the Policy Statement for the Layer 2 Circuit Community . . . 26
Example: Configuring a Policy for a Layer 2 Circuit Community . . . 27
Verifying the Layer 2 Circuit Policy Configuration . . . 28
Configuring ATM Trunking on Layer 2 Circuits . . . 28
Configuring Bandwidth Allocation and Call Admission Control in Layer 2 Circuits . . . 30
Reducing APS Switchover Time in Layer 2 Circuits . . . 31
Configuring Per-Packet Load Balancing . . . 31
Configuring Fast APS Switchover . . . 32
Tracing Layer 2 Circuit Operations . . . 33

Chapter 4 **Layer 2 Circuits Example . . . 35**

Introduction to Layer 2 Circuit Protect Interfaces Example . . . 35
Configuring Router PE1 . . . 36
Configuring Router PE2 . . . 37
Configuring Router CE1 . . . 39
Configuring Router CE2 . . . 39
Example: Configuring an Egress Protection LSP for a Layer 2 Circuit . . . 40

Chapter 5 **Additional Examples . . . 51**

Using the Layer 2 Interworking Interface to Interconnect a Layer 2 VPN to a Layer 2 VPN . . . 51
Example: Interconnecting a Layer 2 Circuit with a Layer 2 VPN . . . 53
Using the Layer 2 Interworking Interface to Interconnect a Layer 2 Circuit to a Layer 2 VPN . . . 61
Layer 2 Circuit Overview . . . 63
Applications for Interconnecting a Layer 2 Circuit with a Layer 3 VPN . . . 64
Applications for Interconnecting a Layer 2 Circuit with a Layer 2 Circuit . . . 64
Example: Interconnecting a Layer 2 Circuit with a Layer 2 Circuit . . . 65

Part 3 Administration

Chapter 6 **Layer 2 Circuit Reference . . . 83**

Supported Layer 2 Circuit Standards . . . 83

Chapter 7 Summary of Layer 2 Circuit Configuration Statements 85

bandwidth 85
community 86
control-word 87
description 87
egress-protection (Layer 2 circuit) 88
egress-protection (MPLS) 88
encapsulation (Physical Interface) 89
encapsulation-type 93
end-interface 94
fast-aps-switch 95
ignore-encapsulation-mismatch 96
ignore-mtu-mismatch 96
install-nexthop 97
interface 98
l2circuit 100
local-switching 101
mtu 102
neighbor 103
protect-interface 104
protected-l2circuit 105
protector-interface 106
protector-pe 106
pseudowire-status-tlv 107
psn-tunnel-endpoint 108
static 109
traceoptions 110
virtual-circuit-id 111

Part 4 Index

Index 115

List of Figures

Part 1 Overview

Chapter 1 Introduction to Layer 2 Circuits 3

Figure 1: Components of a Layer 2 Circuit 3
Figure 2: Egress protection LSP 7

Part 2 Configuration

Chapter 3 Configuring Layer 2 Circuits 13

Figure 3: ATM Trunking on Layer 2 Circuits 28

Chapter 4 Layer 2 Circuits Example 35

Figure 4: Layer 2 Circuits Using Protect Interfaces 35
Figure 5: Egress Protection LSP Configured from Router PE2 to Router PE3 41

Chapter 5 Additional Examples 51

Figure 6: Physical Topology of a Layer 2 Circuit to a Layer 2 VPN Connection . . . 54
Figure 7: Logical Topology of a Layer 2 Circuit to a Layer 2 VPN Connection 54
Figure 8: Physical Topology of a Layer 2 Circuit Terminating into a Layer 2 Circuit 66
Figure 9: Logical Topology of a Layer 2 Circuit Terminating into a Layer 2 Circuit 66

List of Tables

About the Documentation xi

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

About the Documentation

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

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 xiii 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 xiii 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

- Introduction to Layer 2 Circuits on page 3
- Introduction to Configuring Layer 2 Circuits on page 9

CHAPTER 1

Introduction to Layer 2 Circuits

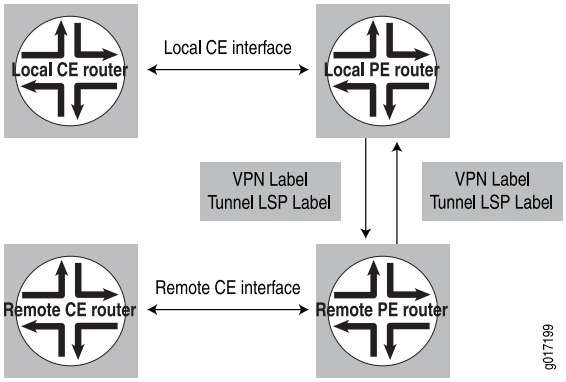
- Layer 2 Circuit Overview on page 3
- Layer 2 Circuit Bandwidth Accounting and Call Admission Control on page 4
- Egress Protection LSPs for Layer 2 Circuits on page 7

Layer 2 Circuit Overview

A Layer 2 circuit is a point-to-point Layer 2 connection transported by means of MPLS or another tunneling technology on the service provider's network. A Layer 2 circuit is similar to a circuit cross-connect (CCC), except that multiple Layer 2 circuits can be transported over a single label-switched path (LSP) tunnel between two provider edge (PE) routers. In contrast, each CCC requires a dedicated LSP.

The Junos OS implementation of Layer 2 circuits supports only the remote form of a Layer 2 circuit; that is, a connection from a local customer edge (CE) router to a remote CE router. Figure 1 on page 3 illustrates the components of a Layer 2 circuit.

Figure 1: Components of a Layer 2 Circuit



The interfaces shown in Figure 1 on page 3 are logical interfaces. Packets are sent to the remote CE router by means of an egress virtual private network (VPN) label advertised by the remote PE router. The VPN label transits over either an RSVP or an LDP LSP (or other type) tunnel to the remote PE router connected to the remote CE router. If you configure RSVP for Layer 2 circuits, you must also configure LDP.

Return traffic sent from the remote CE router to the local CE router uses an ingress VPN label advertised by the local PE router, which again transits over an RSVP and LDP LSP

to the local PE router from the remote PE router. LDP is the signaling protocol used for advertising VPN labels.

Layer 2 Circuit Bandwidth Accounting and Call Admission Control

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

- Bandwidth Accounting and Call Admission Control Overview on page 4
- Selecting an LSP Based on the Bandwidth Constraint on page 4
- LSP Path Protection and CAC on page 5
- Layer 2 Circuits Trunk Mode on page 6

Bandwidth Accounting and Call Admission Control Overview

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

Providing this type of service guarantee requires the following:

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

Selecting an LSP Based on the Bandwidth Constraint

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

- If multiple LSPs meet the bandwidth requirements, the first LSP found that can satisfy the bandwidth requirements for the Layer 2 circuit is selected.
- If there is more than one next hop mapped to the same LSP, then all the next hops that map to that LSP and pass CAC constraints are installed. This allows the Layer 2 circuit routes to restore themselves quickly in case of failure.
- The available bandwidth on the selected LSP is decremented by the bandwidth required for each Layer 2 circuit. Similarly, when the Layer 2 circuit route is changed or deleted (for example, when the route is disassociated from that particular LSP), the bandwidth on the corresponding LSP is incremented.
- There are no priorities among different Layer 2 circuits competing for the same LSP next hop in the core network.
- When an LSP's bandwidth changes, the Layer 2 circuits using that LSP repeat the CAC process again.

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

LSP Path Protection and CAC

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

For more information about MPLS path protection features, see the *Junos OS MPLS Applications Configuration Guide*.

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

- Secondary Paths and CAC on page 5
- Fast Reroute and CAC on page 6
- Link and Node Protection and CAC on page 6

Secondary Paths and CAC

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

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

CAC allows the Layer 2 circuit to be switched to the secondary path if the primary path fails.

- If the LSP has primary and secondary paths configured with different bandwidths, each path must run through CAC independently. If the active path for that LSP passes CAC constraints successfully, then that next hop is installed and the corresponding LSP is selected to transport the Layer 2 circuit traffic. The LSP's secondary paths are then checked for CAC, and installed if there is sufficient bandwidth.

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

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

Fast Reroute and CAC

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

Link and Node Protection and CAC

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

Layer 2 Circuits Trunk Mode

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

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

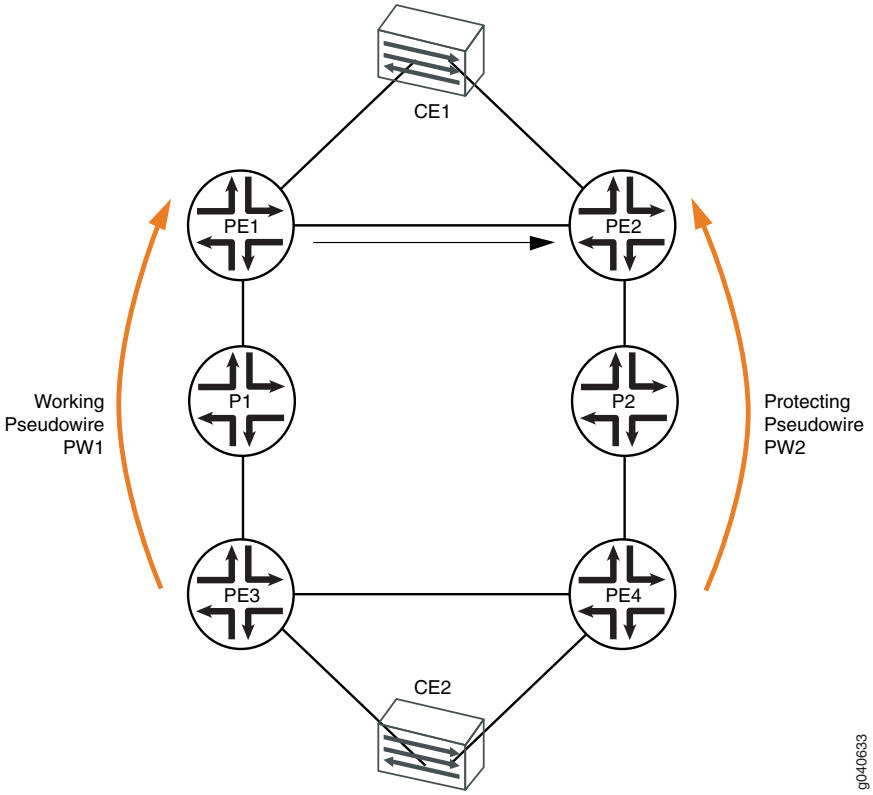
- CoS queues in Layer 2 circuit trunk mode—For ATM2 IQ interfaces, you can configure ATM CoS queues for Layer 2 circuit trunk mode.
- Layer 2 circuit trunk mode scheduling—For ATM2 IQ interfaces configured to use Layer 2 circuit trunk mode, you can share a scheduler among 32 trunks on an ATM port.
-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

For a detailed overview and configuration documentation, see the *Junos OS Network Interfaces Configuration Guide* and *Junos OS Class of Service Configuration Guide*.

Egress Protection LSPs for Layer 2 Circuits

An egress protection LSP provides link protection for link between PE routers and CE devices as illustrated in Figure 2 on page 7.

Figure 2: Egress protection LSP



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

In the topology shown in Figure 2 on page 7, if there was a link or node failure in the core network (for example, a link failure from router P1 to PE1, from router PE3 to P1, or a node failure of router P1), MPLS fast reroute can be triggered on the transport LSPs between router PE3 and router PE1 to repair the connection within tens of milliseconds. Egress protection LSPs address the problem of when a link failure occurs at the edge of the network (for example, a link failure on router PE1 to device CE1).

An egress protection LSP has been configured from router PE1 to router PE2. In the event of a link failure between router PE1 and device CE1, traffic can be switched to the egress

protection LSP. Traffic from device CE2 can now be routed through path PE3-P1-PE1-PE2 to reach device CE1.

CHAPTER 2

Introduction to Configuring Layer 2 Circuits

- Configuring LDP for Layer 2 Circuits on page 9

Configuring LDP for Layer 2 Circuits

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

For detailed information about how to configure LDP, see the *Junos OS MPLS Applications Configuration Guide*.

PART 2

Configuration

- Configuring Layer 2 Circuits on page 13
- Layer 2 Circuits Example on page 35
- Additional Examples on page 51

CHAPTER 3

Configuring Layer 2 Circuits

- Introduction to Configuring Layer 2 Circuits on page 13
- Configuring Local Interface Switching in Layer 2 Circuits on page 14
- Configuring Interfaces for Layer 2 Circuits on page 16
- Configuring Static Layer 2 Circuits on page 24
- Configuring Policies for Layer 2 Circuits on page 25
- Configuring ATM Trunking on Layer 2 Circuits on page 28
- Configuring Bandwidth Allocation and Call Admission Control in Layer 2 Circuits on page 30
- Reducing APS Switchover Time in Layer 2 Circuits on page 31
- Tracing Layer 2 Circuit Operations on page 33

Introduction to Configuring Layer 2 Circuits

To configure a Layer 2 circuit, include the **l2circuit** statement:

```
l2circuit {
  local-switching {
    interface interface-name {
      description text;
      end-interface {
        interface interface-name;
        protect-interface interface-name;
      }
      ignore-mtu-mismatch;
      protect-interface interface-name;
    }
  }
  neighbor address {
    interface interface-name {
      community community-name;
      (control-word | no-control-word);
      description text;
      encapsulation-type type;
      mtu mtu-number;
      protect-interface interface-name;
      psn-tunnel-endpoint address;
      static {
```

```
            incoming-label label;
            outgoing-label label;
          }
          virtual-circuit-id identifier;
        }
      }
      traceoptions {
        file filename <files number> <size size> <world-readable | no-world-readable>;
        flag flag <flag-modifier> <disable>;
      }
    }
```

You can include this statement at the following hierarchy levels:

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

Configuring Local Interface Switching in Layer 2 Circuits

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

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

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

You can include this statement at the following hierarchy levels:

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

The following sections describe how to configure local interface switching:

- Configuring the Interfaces for the Local Interface Switch on page 15
- Enabling Local Interface Switching When the MTU Does Not Match on page 15

Configuring the Interfaces for the Local Interface Switch

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

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

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

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

For more information about how to configure protect interfaces, see "Configuring the Protect Interface" on page 21.

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

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

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

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

Enabling Local Interface Switching When the MTU Does Not Match

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

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

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

You can include this statement at the following hierarchy levels:

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

Configuring Interfaces for Layer 2 Circuits

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

- Configuring the Address for the Neighbor of the Layer 2 Circuit on page 16
- Configuring the Neighbor Interface for the Layer 2 Circuit on page 16
- Configuring the Interface Encapsulation Type for Layer 2 Circuits on page 23
- Configuring ATM2 IQ Interfaces for Layer 2 Circuits on page 24

Configuring the Address for the Neighbor of the Layer 2 Circuit

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

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

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

You can include this statement at the following hierarchy levels:

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

Configuring the Neighbor Interface for the Layer 2 Circuit

Each Layer 2 circuit is represented by the logical interface connecting the local provider edge (PE) router to the local customer edge (CE) router. This interface is tied to the Layer 2 circuit neighbor configured in "Configuring the Address for the Neighbor of the Layer 2 Circuit" on page 16.

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

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

```
    protect-interface interface-name;
    pseudowire-status-tlv;
    psn-tunnel-endpoint address;
    virtual-circuit-id identifier;
}
```

You can include this statement at the following hierarchy levels:

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

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

- Configuring a Community for the Layer 2 Circuit on page 17
- Configuring the Control Word for Layer 2 Circuits on page 17
- Configuring the Encapsulation Type for the Layer 2 Circuit Neighbor Interface on page 19
- Enabling the Layer 2 Circuit When the Encapsulation Does Not Match on page 19
- Configuring the MTU for the Layer 2 Circuit Neighbor Interface on page 20
- Configuring the Protect Interface on page 21
- Configuring the Protect Interface From Switching Over to the Primary Interface on page 21
- Configuring the Pseudowire Status TLV on page 21
- Configuring Layer 2 Circuits over Both RSVP and LDP LSPs on page 22
- Configuring the Virtual Circuit ID on page 23

Configuring a Community for the Layer 2 Circuit

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

```
community community-name;
```

You can include this statement at the following hierarchy levels:

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

For information about how to configure a routing policy for a Layer 2 circuit, see "Configuring Policies for Layer 2 Circuits" on page 25.

Configuring the Control Word for Layer 2 Circuits

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

However, individual bits are available in a control word that can carry Layer 2 protocol control information. The control information is mapped into the control word, which

allows the header of a Layer 2 protocol to be stripped from the frame. The remaining data and control word can be sent over the Layer 2 circuit, and the frame can be reassembled with the proper control information at the egress point of the circuit.

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

- Frame Relay—The control word supports the transport of discard eligible (DE), forward explicit congestion notification (FECN), and backward explicit congestion notification (BECN) information. For configuration information, see "Configuring the Control Word for Frame Relay Interfaces" on page 18.
- ATM AAL5 mode—The control word supports the transport of sequence number processing, ATM cell loss priority (CLP), and explicit forward congestion indication (EFCI) information. When you configure an AAL5 mode Layer 2 circuit, the control information is carried by default and no additional configuration is needed.
- ATM cell-relay mode—The control word supports sequence number processing only. When you configure a cell-relay mode Layer 2 circuit, the sequence number information is carried by default and no additional configuration is needed.

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

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

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

- Configuring the Control Word for Frame Relay Interfaces on page 18
- Disabling the Control Word for Layer 2 Circuits on page 19

Configuring the Control Word for Frame Relay Interfaces

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

For information about how to configure the control bits, see the *Junos OS Network Interfaces Configuration Guide* and the *Junos OS Feature Guides*.

Disabling the Control Word for Layer 2 Circuits

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

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

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

Configuring the Encapsulation Type for the Layer 2 Circuit Neighbor Interface

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

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

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

You can include this statement at the following hierarchy levels:

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

Enabling the Layer 2 Circuit When the Encapsulation Does Not Match

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

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

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

Configuring the MTU for the Layer 2 Circuit Neighbor Interface

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

- Enabling the Layer 2 Circuit When the MTU Does Not Match on page 20
- Configuring the MTU Advertised for a Layer 2 Circuit on page 20

Enabling the Layer 2 Circuit When the MTU Does Not Match

You can configure the Junos OS to allow a Layer 2 circuit to be established even though the MTU configured on the PE router does not match the MTU configured on the remote PE router by including the **ignore-mtu-mismatch** statement:

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

You can include this statement at the following hierarchy levels:

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

Configuring the MTU Advertised for a Layer 2 Circuit

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

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

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

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

- An explicitly configured MTU is signaled to the remote PE device. The configured MTU is also compared to the MTU received from the remote PE device. If there is a conflict, the Layer 2 circuit is taken down.
- If you configure an MTU for an ATM cell relay interface on an ATM II PIC, the configured MTU is used to compute the cell bundle size advertised for that Layer 2 circuit, instead of the default interface MTU.
- A configured MTU is used only in the control plane. It is not enforced in the data plane. You need to ensure that the CE device for a given Layer 2 circuit uses the correct MTU for data transmission.

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

```
mtu mtu-number;
```

You can include this statement at the following hierarchy levels:

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

Configuring the Protect Interface

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

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

```
protect-interface interface-name;
```

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

For an example of how to configure a protect interface for a Layer 2 circuit, see "Introduction to Layer 2 Circuit Protect Interfaces Example" on page 35.

Configuring the Protect Interface From Switching Over to the Primary Interface

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

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

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

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

Configuring the Pseudowire Status TLV

The pseudowire status type length variable (TLV) is used to communicate the status of a pseudowire back and forth between two PE routers. For Layer 2 circuit configurations, you can configure the PE router to negotiate the pseudowire with its neighbor using the pseudowire status TLV. This same functionality is also available for LDP VPLS neighbor

configurations. The pseudowire status TLV is configurable for each pseudowire connection and is disabled by default. The pseudowire status negotiation process assures that a PE router reverts back to the label withdraw method for pseudowire status if its remote PE router neighbor does not support the pseudowire status TLV.

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

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

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

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

Configuring Layer 2 Circuits over Both RSVP and LDP LSPs

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

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

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

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

You can include this statement at the following hierarchy levels:

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

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

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

Example: PSN Tunnel Endpoint

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

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

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

Configuring the Virtual Circuit ID

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

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

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

```
virtual-circuit-id identifier;
```

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

Configuring the Interface Encapsulation Type for Layer 2 Circuits

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

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

```
encapsulation encapsulation;
```

You can include this statement at the following hierarchy levels:

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

Configuring ATM2 IQ Interfaces for Layer 2 Circuits

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

The configuration statements are as follows:

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

For more information about these statements, see the *Junos OS System Basics Configuration Guide*. For more information about how to configure ATM2 IQ interfaces, see the *Junos OS Network Interfaces Configuration Guide*.

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

The Junos OS implementation has the following differences:

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

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

Configuring Static Layer 2 Circuits

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

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

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

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

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

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

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

Configuring Policies for Layer 2 Circuits

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

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

- Configuring the Layer 2 Circuit Community on page 25
- Configuring the Policy Statement for the Layer 2 Circuit Community on page 26
- Verifying the Layer 2 Circuit Policy Configuration on page 28

Configuring the Layer 2 Circuit Community

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

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

You can include this statement at the following hierarchy levels:

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

name identifies the community or communities.

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

- A normal community uses the following community ID format:

  as-number*:*community-value

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

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

- An extended community uses the following community ID format:

  type*:*administrator*:*assigned-number

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

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

  assigned-number identifies the local provider.

  You also need to configure the community for the Layer 2 circuit interface; see "Configuring a Community for the Layer 2 Circuit" on page 17.

Configuring the Policy Statement for the Layer 2 Circuit Community

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

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

You can include this statement at the following hierarchy levels:

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

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

```
install-nexthop except;
```

You can include this statement at the following hierarchy levels:

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

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

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

You can include these statements at the following hierarchy levels:

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

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

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

You can include these statements at the following hierarchy levels:

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

Example: Configuring a Policy for a Layer 2 Circuit Community

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

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

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

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

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

Verifying the Layer 2 Circuit Policy Configuration

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

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

For more information about how to configure routing policies, see the *Junos OS Policy Framework Configuration Guide*.

Configuring ATM Trunking on Layer 2 Circuits

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

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

Figure 3: ATM Trunking on Layer 2 Circuits

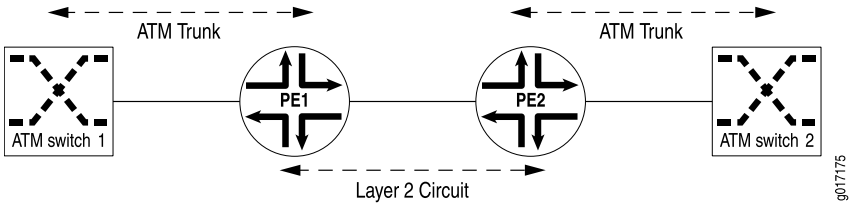


Figure 3 on page 28 illustrates how ATM switches could be linked together by a Layer 2 circuit. The PE1 Router is configured to receive ATM trunk traffic from ATM Switch 1. As each ATM cell is received on the PE1 Router, it is classified by means of the class-of-service (CoS) information in the cell header and then encapsulated as a labeled packet. The

CoS information and cell loss priority (CLP) of the ATM cell are copied into the experimental (EXP) bits of the MPLS label. The labeled packet is then transported across the service provider network to the PE2 Router by means of a Layer 2 circuit.

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

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

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

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

```
[edit chassis fpc number pic number]
atm-l2circuit-mode trunk (uni | nni);
```

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

You also need to configure each ATM trunk for a specific logical interface. Each ATM trunk has a trunk identifier in the range from 0 to 31. This configuration step is in addition to the typical configuration steps you follow related to configuring interfaces for Layer 2 circuits, as described in "Configuring Interfaces for Layer 2 Circuits" on page 16.

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

```
trunk-id number;
```

You can include this statement at the following hierarchy levels:

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

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

Configuring Bandwidth Allocation and Call Admission Control in Layer 2 Circuits

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

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

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

```
bandwidth bandwidth;
```

Specify the bandwidth in bits per second.

You can include this statement at the following hierarchy levels:

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

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

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

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

You can include this statement at the following hierarchy levels:

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

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

Reducing APS Switchover Time in Layer 2 Circuits

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

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

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

NOTE:

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

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

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

- Configuring Per-Packet Load Balancing on page 31
- Configuring Fast APS Switchover on page 32

Configuring Per-Packet Load Balancing

To configure per-packet load balancing for all routes:

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

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

```
user@host# set load-balance per-packet
```

For example:

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

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

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

For example:

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

Configuring Fast APS Switchover

To configure fast APS switchover:

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

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

For example:

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

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

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

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

For example:

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

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

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

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

For example:

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

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

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

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

   For example:

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

Tracing Layer 2 Circuit Operations

To trace the creation of and changes to Layer 2 circuits, include the **traceoptions** statement:

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

You can include this statement at the following hierarchy levels:

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

Specify the following flags to trace the indicated operations on Layer 2 circuits:

- **connections**—Layer 2 circuit connections (events and state changes)
- **error**—Error conditions
- **FEC**—Layer 2 circuit advertisements received or sent using LDP
- **topology**—Layer 2 circuit topology changes caused by reconfiguration or advertisements received from other PE routers

CHAPTER 4

Layer 2 Circuits Example

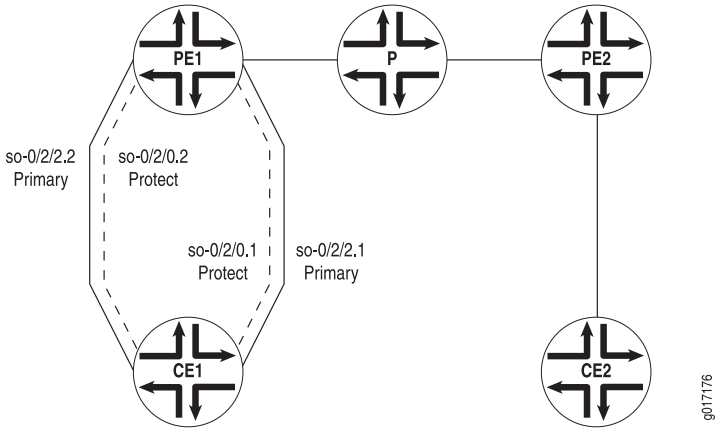
- Introduction to Layer 2 Circuit Protect Interfaces Example on page 35
- Example: Configuring an Egress Protection LSP for a Layer 2 Circuit on page 40

Introduction to Layer 2 Circuit Protect Interfaces Example

This example illustrates how you might configure a Layer 2 circuit with protect interfaces. Protect interfaces act as backups for their associated interfaces. The primary interface has priority over the protect interface and carries network traffic as long as it is functional. If the primary interface fails, the protect interface is activated. These interfaces can also share the same virtual path identifier (VPI) or virtual circuit identifier (VCI).

Figure 4 on page 35 shows the network topology used in this example.

Figure 4: Layer 2 Circuits Using Protect Interfaces



The following sections describe how to configure a Layer 2 circuit to use a protect interface:

- Configuring Router PE1 on page 36
- Configuring Router PE2 on page 37
- Configuring Router CE1 on page 39
- Configuring Router CE2 on page 39

Configuring Router PE1

Configure an interface for traffic to Router CE1 from Router PE1 at the **[edit interfaces]** hierarchy level:

```
[edit interfaces]
so-0/2/2 {
   description "Router CE1 so-0/2/2";
   no-keepalives;
   encapsulation frame-relay-ccc;
   unit 1 {
      encapsulation frame-relay-ccc;
      point-to-point;
      dlci 600;
   }
   unit 2 {
      encapsulation frame-relay-ccc;
      point-to-point;
      dlci 602;
   }
}
```

Configure an interface for traffic to Router CE1 from Router PE1 at the **[edit interfaces]** hierarchy level. Logical interface **so-0/2/0.2** acts as the protect interface for **so-0/2/2.2**, and logical interface **so-0/2/0.1** acts as the protect interface for **so-0/2/2.1**:

```
[edit interfaces]
so-0/2/0 {
   description "to Router CE1 so-0/3/0";
   no-keepalives;
   encapsulation frame-relay-ccc;
   unit 1 {
      encapsulation frame-relay-ccc;
      dlci 600;
   }
   unit 2 {
      encapsulation frame-relay-ccc;
      dlci 602;
   }
}
```

Configure an interface for traffic to Router PE2 from Router PE1 at the **[edit interfaces]** hierarchy level:

```
[edit interfaces]
so-0/2/1 {
   description "to Router PE2 so-1/0/1";
   unit 0 {
      family inet {
         address 100.100.40.22/32 {
            destination 100.100.40.23;
         }
      }
      family iso;
      family mpls;
```

```
    }
}
```

Configure an interface for traffic to Router PE2 from Router PE1 at the **[edit interfaces]** hierarchy level:

```
[edit interfaces]
so-0/2/3 {
    description "Router PE2 so-1/0/3";
    unit 0 {
        family inet;
        family iso;
        family mpls;
    }
    lo0 {
        unit 0 {
            family inet {
                address 127.0.0.1/32;
                address 10.100.40.200/32;
            }
            family iso {
                address 47.0005.80ff.f800.0000.0108.0001.1921.6800.4213.00;
            }
        }
    }
}
```

Configure the Layer 2 circuit by including the **l2circuit** statement at the **[edit protocols]** hierarchy level. The logical interfaces for the Layer 2 circuits and their corresponding protect interfaces are included here:

```
[edit protocols]
l2circuit {
    neighbor 10.100.40.210 {
        interface so-0/2/2.2 {
            protect-interface so-0/2/0.2;
            virtual-circuit-id 2;
            no-control-word;
        }
        interface so-0/2/2.1 {
            protect-interface so-0/2/0.1;
            virtual-circuit-id 1;
            no-control-word;
        }
    }
}
```

Configuring Router PE2

Configure an interface for traffic to Router CE2 from Router PE2:

```
[edit interfaces]
so-1/0/0 {
    description "to Router CE2 so-0/2/0";
    no-keepalives;
    encapsulation frame-relay-ccc;
```

```
    unit 1 {
      encapsulation frame-relay-ccc;
      point-to-point;
      dlci 700;
    }
    unit 2 {
      encapsulation frame-relay-ccc;
      point-to-point;
      dlci 702;
    }
  }
```

Configure an interface for traffic to Router PE1 from Router PE2:

```
[edit interfaces]
so-1/0/1 {
  description "to Router PE1 so-0/2/1";
  unit 0 {
    family inet {
      address 100.100.40.23/32 {
        destination 100.100.40.22;
      }
    }
    family iso;
    family mpls;
  }
}
```

Configure an interface for traffic to Router PE1 from Router PE2:

```
[edit interfaces]
so-1/0/3 {
  description "to Router PE1 so-0/2/3";
  unit 0 {
    family inet;
    family iso;
    family mpls;
  }
}
lo0 {
  unit 0 {
    family inet {
      address 127.0.0.1/32;
      address 10.100.40.210/32;
    }
    family iso {
      address 47.0005.80ff.f800.0000.0108.0001.1921.6800.4216.00;
    }
  }
}
```

Configure the Layer 2 circuit at the **[edit protocols]** hierarchy level:

```
[edit protocols]
l2circuit {
  neighbor 10.100.40.200 {
    interface so-1/0/0.1 {
```

```
        virtual-circuit-id 1;
        no-control-word;
      }
      interface so-1/0/0.2 {
        virtual-circuit-id 2;
        no-control-word;
      }
    }
}
```

Configuring Router CE1

Configure an interface for traffic to Router PE1 from Router CE1:

```
[edit interfaces]
so-0/3/0 {
  description "to Router PE1 so-0/2/0";
  no-keepalives;
  encapsulation frame-relay;
  unit 1 {
    dlci 601;
    family inet {
      address 12.12.12.1/24;
    }
  }
}
```

Configure an interface for traffic to Router PE1 from Router CE1:

```
[edit interfaces]
so-0/3/1 {
  description "Router PE1 so-0/2/2";
  no-keepalives;
  encapsulation frame-relay;
  unit 0 {
    dlci 600;
    family inet {
      address 10.10.10.1/24;
      address 11.1.1.1/24;
    }
    family iso;
    family mpls;
  }
  unit 2 {
    dlci 602;
    family inet {
      address 13.13.13.1/24;
    }
  }
}
```

Configuring Router CE2

Configure an interface for traffic to Router PE2 from Router CE2:

```
[edit interfaces]
so-0/2/0 {
```

```
    description "to Router PE2 so-1/0/0";
    no-keepalives;
    encapsulation frame-relay;
    unit 1 {
      dlci 700;
      family inet {
        address 10.10.10.2/24;
        address 11.1.1.2/24;
        address 12.12.12.2/24;
      }
    }
    unit 2 {
      dlci 702;
      family inet {
        address 13.13.13.2/24;
      }
    }
  }
```

Example: Configuring an Egress Protection LSP for a Layer 2 Circuit

This example shows how to configure an egress protection LSP.

- Requirements on page 40
- Egress Protection LSP Overview on page 40
- Egress Protection LSP Configuration on page 42

Requirements

Egress protection LSPs are supported on Juniper Networks MX Series routers only. This requirement applies to the PE routers facilitating the egress protection LSP.

Egress Protection LSP Overview

If there is a link or node failure in the core network, a protection mechanism such as MPLS fast reroute can be triggered on the transport LSPs between the PE routers to repair the connection within tens of milliseconds. An egress protection LSP addresses the problem of when a link failure occurs at the edge of the network (for example, a link failure between a PE router and a CE device). Egress protection LSPs do not address the problem of a node failure at the edge of the network (for example, a failure of a PE router). An egress protection LSP is an RSVP-signaled ultimate hop popping LSP.

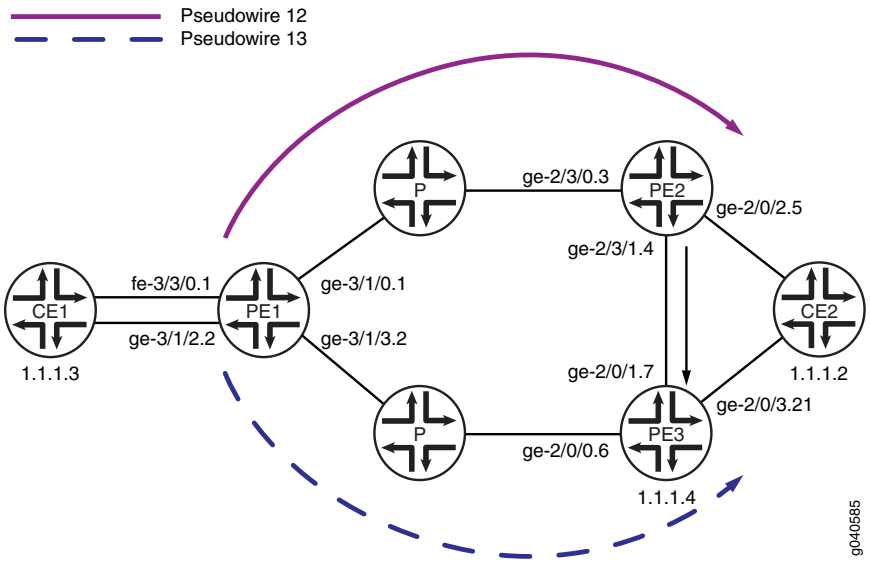
This example includes the following configuration concepts and statements that are unique to the configuration of an egress protection LSP:

- **context-identifier**—Specifies an IPV4 address used to define the pair of PE routers participating in the egress protection LSP. The context identifier is used to assign an identifier to the protector PE router. The identifier is propagated to the other PE routers participating in the network, making it possible for the protected egress PE router to signal the egress protection LSP to the protector PE router.
- **egress-protection**—Configures the protector information for the protected Layer 2 circuit and configures the protector Layer 2 circuit at the **[edit protocols l2circuit]**

hierarchy level. Configures an LSP as an egress protection LSP at the **[edit protocols mpls label-switched-path *lsp-name*]** hierarchy level. It also configures the context identifier at the **[edit protocols mpls]** hierarchy level.

- **protected-l2circuit**—Specifies which Layer 2 circuit is to be protected by the egress protect LSP. This statement includes the following sub-statements: **ingress-pe**, **egress-pe**, and **virtual-circuit-id**. These sub-statements specify the address of the PE router at the ingress of the Layer 2 circuit, the address of the PE router at the egress of the Layer 2 circuit, and the Layer 2 circuit's identifier respectively.
- **protector-interface**—Specify the interface used by the egress protection LSP. In the event of a local link failure to a CE device, the egress protect LSP uses the interface specified to communicate with the protector PE router.
- **protector-pe**—Specify the IPv4 address of the protector PE router. The protector PE router must have a connection to the same CE device as the protected PE router for the egress protect LSP to function. This statement includes the following sub-statements: **context-identifier** and **lsp**. The **lsp** statement specifies the LSP to be used as the actual egress protection LSP.

Figure 5: Egress Protection LSP Configured from Router PE2 to Router PE3



Pseudowires are configured along two paths, one from router PE1 to router PE2 (pseudowire 12) and one from router PE1 to router PE3 (pseudowire 13). In the event of a failure on the link between router PE2 and device CE2, traffic is switched to the egress protection LSP configured between router PE2 and router PE3 (the protector PE router):

- Device CE1—Traffic origin
- Router PE1—Ingress PE router
- Router PE2—Egress PE router

- Router PE3—Protector PE router
- Device CE2—Traffic destintation

This example shows how to configure routers PE1, PE2, and PE3.

Egress Protection LSP Configuration

- Step-by-Step Procedure on page 43

CLI Quick Configuration

To quickly configure an egress protection LSP, copy the following commands into a text file, modify the interface configurations to match your equipment, remove any line breaks, and then paste the commands into the CLI. This group of set commands is for router PE1.

```
set protocols rsvp interface ge-3/1/0.1
set protocols rsvp interface ge-3/1/3.2
set protocols mpls interface ge-3/1/0.1
set protocols mpls interface ge-3/1/3.2
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-3/1/0.1
set protocols ospf area 0.0.0.0 interface ge-3/1/3.2
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ldp interface ge-3/1/0.1
set protocols ldp interface ge-3/1/3.2
set protocols ldp interface lo0.0
set protocols l2circuit neighbor 1.1.1.3 interface fe-3/3/0.1 virtual-circuit-id 32
set protocols l2circuit neighbor 1.1.1.3 interface fe-3/3/0.1 egress-protection
  protector-interface ge-3/1/2.2
set protocols l2circuit neighbor 1.1.1.4 interface ge-3/1/2.2 virtual-circuit-id 33
set policy-options policy-statement load-balance-example then load-balance per-packet
set routing-options router-id 1.1.1.2
set routing-options forwarding-table export load-balance-example
```

To quickly configure an egress protection LSP, copy the following commands into a text file, modify the interface configurations to match your equipment, remove any line breaks, and then paste the commands into the CLI. This group of set commands is for router PE2.

```
[edit]
set protocols rsvp tunnel-services
set protocols rsvp interface ge-2/3/0.3
set protocols rsvp interface ge-2/3/1.4 link-protection
set protocols ldp interface ge-2/3/0.3
set protocols ldp interface ge-2/3/1.4
set protocols ldp interface lo0.0
set protocols ldp upstream-label-assignment
set protocols mpls label-switched-path protected-lsp to 2.2.3.4
set protocols mpls label-switched-path protected-lsp egress-protection
set protocols mpls interface ge-2/3/0.3
set protocols mpls interface ge-2/3/1.4
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-2/3/0.3
set protocols ospf area 0.0.0.0 interface ge-2/3/1.4
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols l2circuit neighbor 1.1.1.2 interface ge-2/0/2.5 virtual-circuit-id 23
```

```
set protocols l2circuit neighbor 1.1.1.2 interface ge-2/0/2.5 egress-protection protector-pe
  1.1.1.4
set protocols l2circuit neighbor 1.1.1.2 interface ge-2/0/2.5 egress-protection protector-pe
  context-identifier 2.2.3.4
set policy-options policy-statement load-balance-example then load-balance per-packet
set routing-options router-id 1.1.1.3
set routing-options forwarding-table export load-balance-example
```

To quickly configure an egress protection LSP, copy the following commands into a text file, modify the interface configurations to match your equipment, remove any line breaks, and then paste the commands into the CLI. This group of set commands is for router PE3.

```
set protocols rsvp tunnel-services
set protocols rsvp interface ge-2/0/0.6
set protocols rsvp interface ge-2/0/1.7
set protocols mpls interface ge-2/0/0.6
set protocols mpls interface ge-2/0/1.7
set protocols mpls egress-protection context-identifier 2.2.3.4 protector
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-2/0/0.6
set protocols ospf area 0.0.0.0 interface ge-2/0/1.7
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ldp interface ge-2/0/0.6
set protocols ldp interface ge-2/0/1.7
set protocols ldp interface lo0.0
set protocols ldp upstream-label-assignment
set protocols l2circuit neighbor 1.1.1.2 interface ge-2/0/3.21 virtual-circuit-id 42
set protocols l2circuit neighbor 1.1.1.2 interface ge-2/0/3.21 egress-protection
  protected-l2circuit PW1
set protocols l2circuit neighbor 1.1.1.2 interface ge-2/0/3.21 egress-protection
  protected-l2circuit ingress-pe 1.1.1.2
set protocols l2circuit neighbor 1.1.1.2 interface ge-2/0/3.21 egress-protection
  protected-l2circuit egress-pe 1.1.1.3
set protocols l2circuit neighbor 1.1.1.2 interface ge-2/0/3.21 egress-protection
  protected-l2circuit virtual-circuit-id 31
```

Step-by-Step Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode.

To configure an egress protection LSP, complete the following steps for router PE1:

1. Configure RSVP. Include the interface linked to router PE2 and the interface linked to router PE3.

```
[edit]
user@PE1# edit protocols rsvp
[edit protocols rsvp]
user@PE1# set interface ge-3/1/0.1
[edit protocols rsvp]
user@PE1# set interface ge-3/1/3.2
```

2. Configure LDP. Include the interface linked to router PE2, the interface linked to router PE3, and the loopback interface.

```
[edit]
user@PE1# edit protocols ldp
[edit protocols ldp]
user@PE1# set interface ge-3/1/0.1
[edit protocols ldp]
user@PE1# set interface ge-3/1/3.2
[edit protocols ldp]
user@PE1# set interface lo0.0
```

3. Configure MPLS. Include the interface linked to router PE2 and the interface linked to router PE3.

```
[edit]
user@PE1# edit protocols mpls
[edit protocols mpls]
user@PE1# set interface ge-3/1/0.1
[edit protocols mpls]
user@PE1# set interface ge-3/1/3.2
```

4. Configure OSPF. Include the interface linked to router PE2, the interface linked to router PE3, and the loopback interface in the configuration for the OSPF area.

```
[edit]
user@PE1# edit protocols ospf
[edit protocols ospf]
user@PE1# set interface traffic-engineering
[edit protocols ospf]
user@PE1# set area 0.0.0.0 interface ge-3/1/0.1
[edit protocols ospf]
user@PE1# set area 0.0.0.0 interface ge-3/1/3.2
[edit protocols ospf]
user@PE1# set area 0.0.0.0 interface lo0.0 passive
```

5. Configure Layer 2 circuits to use the egress protection LSP to protect against a link failure to device CE1.

```
[edit]
user@PE1# edit protocols l2circuit
[edit protocols l2circuit]
user@PE1# set neighbor 1.1.1.3 interface fe-3/3/0.1 virtual-circuit-id 32
[edit protocols l2circuit]
user@PE1# edit neighbor 1.1.1.3
[edit protocols l2circuit neighbor 1.1.1.3]
user@PE1# set interface fe-3/3/0.1 egress-protection protector-interface ge-3/1/2.2
[edit protocols l2circuit]
user@PE1# set neighbor 1.1.1.4 interface ge-3/1/2.2 virtual-circuit-id 33
```

6. Configure a load balancing policy.

```
[edit]
user@PE1# set policy-options policy-statement load-balance-example then
  load-balance per-packet
```

7. Configure the routing options to export routes based on the load balancing policy.

```
[edit]
```

```
user@PE1# set routing-options router-id 1.1.1.2
[edit]
user@PE1# set routing-options forwarding-table export load-balance-example
```

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

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode.

To configure an egress protection LSP, complete the following steps for router PE2:

1. Configure RSVP. Include the interface linked to the ingress PE router and the interface linked to the CE device.

```
[edit]
user@PE2# edit protocols rsvp
[edit protocols rsvp]
user@PE2# set tunnel-services
[edit protocols rsvp]
user@PE2# set interface ge-2/3/0.3
[edit protocols rsvp]
user@PE2# set interface ge-2/3/1.4 link-protection
```

2. Configure LDP. Include the interface linked to the ingress PE router and the interface linked to the CE device.

```
[edit]
user@PE2# edit protocols ldp
[edit protocols ldp]
user@PE2# set interface ge-2/3/0.3
[edit protocols ldp]
user@PE2# set interface ge-2/3/1.4
[edit protocols ldp]
user@PE2# set interface lo0.0
[edit protocols ldp]
user@PE2# set upstream-label-assignment
```

3. Configure MPLS and the LSP which acts as the egress protection LSP.

```
[edit]
user@PE2# edit protocols mpls
[edit protocols mpls]
user@PE2# set interface ge-2/3/0.3
[edit protocols mpls]
user@PE2# set interface ge-2/3/1.4
[edit protocols mpls]
user@PE2# set label-switched-path protected-lsp to 2.2.3.4
[edit protocols mpls]
user@PE2# set label-switched-path protected-lsp egress-protection
```

4. Configure OSPF.

```
[edit]
user@PE2# edit protocols ospf
[edit protocols ospf]
user@PE2# set interface traffic-engineering
[edit protocols ospf]
```

```
user@PE2# set interface area 0.0.0.0 interface ge-2/3/0.3
[edit protocols ospf]
user@PE2# set interface area 0.0.0.0 interface ge-2/3/1.4
[edit protocols ospf]
user@PE2# set interface area 0.0.0.0 interface lo0.0 passive
```

5. Configure the Layer 2 circuit to use the egress protection LSP.

```
[edit]
user@PE2# edit protocols l2circuit
[edit protocols l2circuit]
user@PE2# set neighbor 1.1.1.2 interface ge-2/0/2.5 virtual-circuit-id 23
[edit protocols l2circuit]
user@PE2# edit neighbor 1.1.1.2
[edit protocols l2circuit neighbor 1.1.1.2]
user@PE2# set interface ge-2/0/2.5 egress-protectionprotector-pe 1.1.1.4
[edit protocols l2circuit neighbor 1.1.1.2]
user@PE2# set interface ge-2/0/2.5 egress-protection protector-pe
  context-identifier 2.2.3.4
```

6. Configure a load balancing policy.

```
[edit]
user@PE1# set policy-options policy-statement load-balance-example then
  load-balance per-packet
```

7. Configure the routing options to export routes based on the load balancing policy.

```
[edit]
user@PE2# set routing-options router-id 1.1.1.3
[edit]
user@PE2# set routing-options forwarding-table export load-balance-example
```

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

Step-by-Step Procedure

To configure an egress protection LSP, complete the following steps for router PE3:

1. Configure RSVP. Include the interface linked to the ingress PE router and the interface linked to the CE device.

```
[edit]
user@PE3# edit protocols rsvp
[edit protocols rsvp]
user@PE3# set tunnel-services
[edit protocols rsvp]
user@PE3# set interface ge-2/0/0.6
[edit protocols rsvp]
user@PE3# set interface ge-2/0/1.7
```

2. Configure LDP. Include the interface linked to the ingress PE router and the interface linked to the CE device.

```
[edit]
user@PE3# edit protocols ldp
[edit protocols ldp]
user@PE3# set interface ge-2/0/0.6
[edit protocols ldp]
user@PE3# set interface ge-2/0/1.7
[edit protocols ldp]
```

```
user@PE3# set interface lo0.0
[edit protocols ldp]
user@PE3# set upstream-label-assignment
```

3. Configure MPLS and the LSP which acts as the egress protection LSP.

```
[edit]
user@PE3# edit protocols mpls
[edit protocols mpls]
user@PE3# set interface ge-2/0/0.6
[edit protocols mpls]
user@PE3# set interface ge-2/0/1.7
[edit protocols mpls]
user@PE3# set egress-protection context-identifier 2.2.3.4 protector
```

4. Configure OSPF.

```
[edit]
user@PE3# edit protocols ospf
[edit protocols ospf]
user@PE3# set interface traffic-engineering
[edit protocols ospf]
user@PE3# set area 0.0.0.0 interface ge-2/0/0.6
[edit protocols ospf]
user@PE3# set area 0.0.0.0 interface ge-2/0/1.7
[edit protocols ospf]
user@PE3# set area 0.0.0.0 interface lo0.0 passive
```

5. Configure the Layer 2 circuit to use the egress protection LSP.

```
[edit]
user@PE3# edit protocols l2circuit
[edit protocols l2circuit]
user@PE3# set neighbor 1.1.1.2 interface ge-2/0/3.21 virtual-circuit-id 42
[edit protocols l2circuit]
user@PE3# edit neighbor 1.1.1.2
[edit protocols l2circuit neighbor 1.1.1.2]
user@PE3# set interface ge-2/0/3.21 egress-protection protected-l2circuit ingress-pe
  1.1.1.2
[edit protocols l2circuit neighbor 1.1.1.2]
user@PE3# set interface ge-2/0/3.21 egress-protection protected-l2circuit egress-pe
  1.1.1.3
[edit protocols l2circuit neighbor 1.1.1.2]
user@PE3# set interface ge-2/0/3.21 egress-protection
  protected-l2circuitvirtual-circuit-id 31
```

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

Results From configuration mode, confirm your configuration on router PE1 by entering the **show protocols**, **show policy-options**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@PE1# show protocols
rsvp {
  interface ge-3/1/0.1;
```

```
    interface ge-3/1/3.2;
}
mpls {
    interface ge-3/1/0.1;
    interface ge-3/1/3.2;
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface ge-3/1/0.1;
        interface ge-3/1/3.2;
        interface lo0.0 {
            passive;
        }
    }
}
ldp {
    interface ge-3/1/0.1;
    interface ge-3/1/3.2;
    interface lo0.0;
}
l2circuit {
    neighbor 1.1.1.3 {
        interface fe-3/3/0.1 {
            virtual-circuit-id 32;
            egress-protection {
                protector-interface ge-3/1/2.2;
            }
        }
    }
    neighbor 1.1.1.4 {
        interface ge-3/1/2.2 {
            virtual-circuit-id 33;
        }
    }
}
[edit]
user@PE1# show policy-options
policy-statement load-balance-example {
    then {
        load-balance per-packet;
    }
}
[edit]
user@PE1# show routing-options
router-id 1.1.1.2;
forwarding-table {
    export load-balance-example;
}
```

From configuration mode, confirm your configuration on router PE2 by entering the **show protocols**, **show policy-options**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
```

```
user@PE2# show protocols
rsvp {
    tunnel-services;
    interface ge-2/3/0.3;
    interface ge-2/3/1.4 {
        link-protection;
    }
}
mpls {
    label-switched-path protected-lsp {
        to 2.2.3.4;
        egress-protection;
    }
    interface ge-2/3/0.3;
    interface ge-2/3/1.4;
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface ge-2/3/0.3;
        interface ge-2/3/1.4;
        interface lo0.0 {
            passive;
        }
    }
}
ldp {
    interface ge-2/3/0.3;
    interface ge-2/3/1.4;
    interface lo0.0;
    upstream-label-assignment;
}
l2circuit {
    neighbor 1.1.1.2 {
        interface ge-2/0/2.5 {
            virtual-circuit-id 23;
            egress-protection {
                protector-pe 1.1.1.4 context-identifier 2.2.3.4;
            }
        }
    }
}

[edit]
user@PE2# show policy-options
policy-options {
    policy-statement load-balance-example {
        then {
            load-balance per-packet;
        }
    }
}

[edit]
user@PE2# show routing-options
routing-options {
    router-id 1.1.1.3;
```

```
    forwarding-table {
      export load-balance-example;
    }
}
```

From configuration mode, confirm your configuration on router PE3 by entering the **show protocols** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@PE3# show protocols
rsvp {
  tunnel-services;
  interface ge-2/0/0.6;
  interface ge-2/0/1.7;
}
mpls {
  interface ge-2/0/0.6;
  interface ge-2/0/1.7;
  egress-protection {
    context-identifier 2.2.3.4 {
      protector;
    }
  }
}
ospf {
  traffic-engineering;
  area 0.0.0.0 {
    interface ge-2/0/0.6;
    interface ge-2/0/1.7;
    interface lo0.0 {
      passive;
    }
  }
}
ldp {
  interface ge-2/0/0.6;
  interface ge-2/0/1.7;
  interface lo0.0;
  upstream-label-assignment;
}
l2circuit {
  neighbor 1.1.1.2 {
    interface ge-2/0/3.21 {
      virtual-circuit-id 42;
      egress-protection {
        protected-l2circuit PW1 ingress-pe 1.1.1.2 egress-pe 1.1.1.3 virtual-circuit-id 31;
      }
    }
  }
}
```


CHAPTER 5

Additional Examples

- Using the Layer 2 Interworking Interface to Interconnect a Layer 2 VPN to a Layer 2 VPN on page 51
- Example: Interconnecting a Layer 2 Circuit with a Layer 2 VPN on page 53
- Using the Layer 2 Interworking Interface to Interconnect a Layer 2 Circuit to a Layer 2 VPN on page 61
- Layer 2 Circuit Overview on page 63
- Applications for Interconnecting a Layer 2 Circuit with a Layer 3 VPN on page 64
- Applications for Interconnecting a Layer 2 Circuit with a Layer 2 Circuit on page 64
- Example: Interconnecting a Layer 2 Circuit with a Layer 2 Circuit on page 65

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

Instead of using a physical Tunnel PIC for looping the packet received from the Layer 2 VPN to another Layer 2 VPN, the Layer 2 Interworking interface uses Junos OS to stitch together both Layer 2 VPN routes.

To configure the interworking interface, include the **iw0** statement. The **iw0** statement is configured at the **[edit interfaces]** hierarchy level.

```
[edit interfaces]
iw0 {
   unit 0 {
      peer 1;
   }
   unit 1 {
      peer 0;
   }
}
```

The configuration of an interworking (iw) interface is similar to the configuration of a logical tunnel (lt) interface. In this example, the logical interfaces must be associated with the endpoints of both Layer 2 VPN connections terminating on this router. To make the association, include the **interfaces** statement and specify **iw0** as the interface name. Include the statement at the **[edit routing-instances *routing-instances-name* protocols l2vpn site *site-name*]** hierarchy level for each routing instance. The **routing-instances** statement is configured at the **[edit routing-instances]** hierarchy level.

```
[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;
            }
         }
      }
   }
}
```

In addition to the **iw0** interface configuration, Layer 2 interworking **l2iw** protocols need to be configured. Without the **l2iw** configuration, the **l2iw** routes will not be formed, regardless of whether any **iw** interfaces are present. Only standard trace options can be configured within the l2iw protocol. The minimum configuration necessary for the feature to work is shown below:

```
[edit]
protocols {
   l2iw;
}
```

Related Documentation

- Layer 2 VPN Overview
- Layer 2 VPN Applications
- Example: Interconnecting a Layer 2 VPN with a Layer 2 VPN

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

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

- Requirements on page 53
- Overview and Topology on page 53
- Configuration on page 55

Requirements

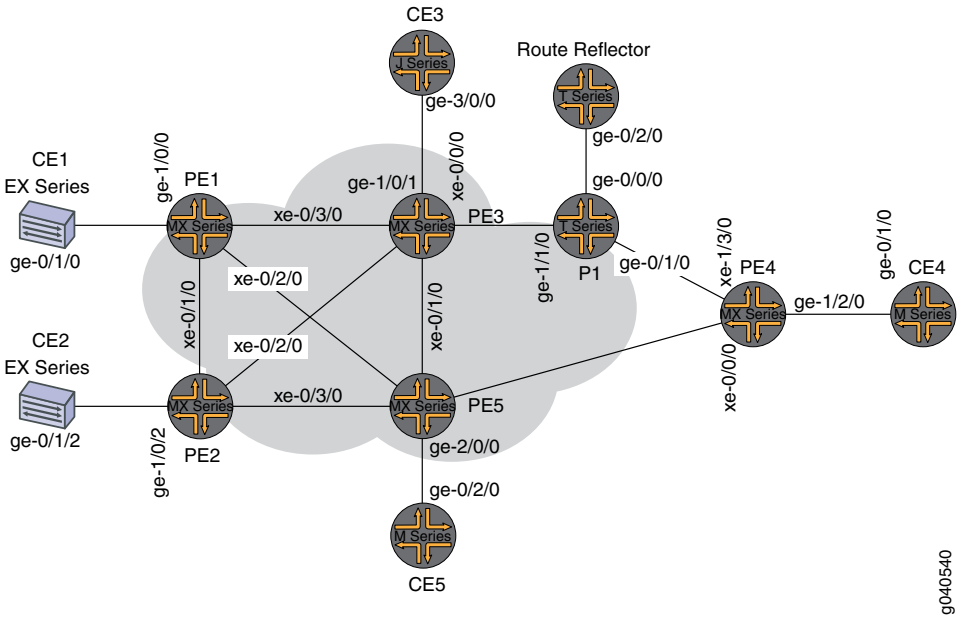
This example uses the following hardware and software components:

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

Overview and Topology

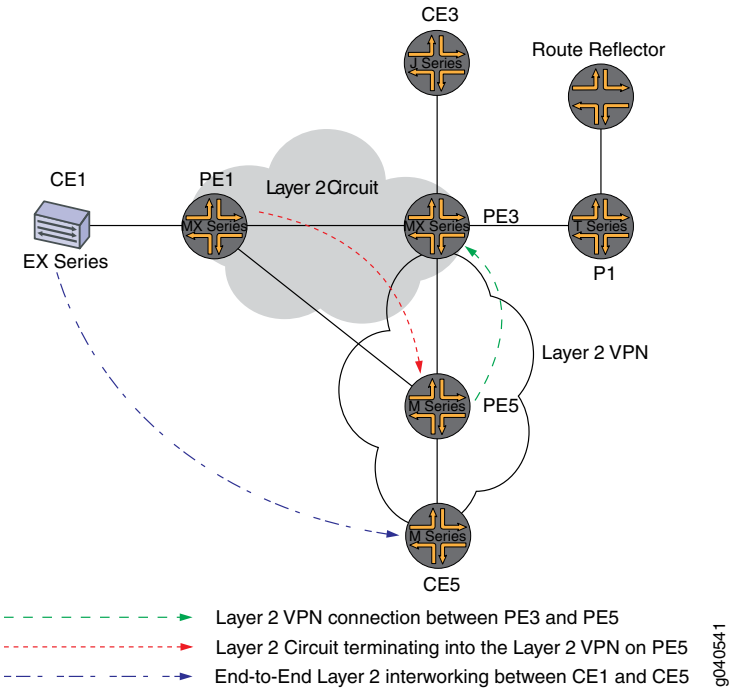
The physical topology of a Layer 2 circuit to a Layer 2 VPN connection is shown in Figure 6 on page 54.

Figure 6: Physical Topology of a Layer 2 Circuit to a Layer 2 VPN Connection



The logical topology of a Layer 2 circuit to a Layer 2 VPN connection is shown in Figure 7 on page 54.

Figure 7: Logical Topology of a Layer 2 Circuit to a Layer 2 VPN Connection



Configuration

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

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

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

This example is organized in the following sections:

- Configuring Protocols on the PE and P Routers on page 55
- Verification on page 59

Configuring Protocols on the PE and P Routers

Step-by-Step Procedure

Base Configuration

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

1. Configure all the PE and P routers with OSPF as the IGP. Enable the MPLS, LDP, and BGP protocols on all interfaces except **fxp.0**. LDP is used as the signaling protocol on Router PE1 for the Layer 2 circuit. The following configuration snippet shows the protocol configuration for Router PE1:

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

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

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

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

```
    }
}
```

Step-by-Step Procedure

Configuring Interfaces

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

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

2. Router PE5 is the router that is *stitching* the Layer 2 circuit to the Layer 2 VPN using the interworking interface. The configuration of the peer unit interfaces is what makes the interconnection.

   On Router PE5, configure the **iw0** interface with two logical interfaces. To configure the **iw0** interface, include the **interfaces** statement and specify**iw0** as the interface name. For the unit 0 logical interface, include the **peer-unit** statement and specify the logical interface **unit 1** as the peer l interface. For the unit 1 logical interface, include the **peer-unit** statement and specify the logical interface **unit 0** as the peer interface.

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

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

   ```
   [edit interfaces]
   ```

```
lo0 {
  unit 0 {
    family inet {
      address 5.5.5.5/32;
    }
  }
}
```

Step-by-Step Procedure

Configuring the Layer 2 circuit protocol

1. On Router PE1, configure the IP address of the remote PE router with the **neighbor** statement. The loopback address and router ID of the PE neighbor is commonly the neighbor's IP address. To allow a Layer 2 circuit to be established even though the maximum transmission unit (MTU) configured on the PE router does not match the MTU configured on the remote PE router, include the **ignore-mtu-mismatch** statement.

   ```
   [edit]
   protocols {
     l2circuit {
       neighbor 5.5.5.5 {
         interface ge-1/0/0.0 {
           virtual-circuit-id 100;
           no-control-word;
           ignore-mtu-mismatch;
         }
       }
     }
   }
   ```

2. On Router PE5, configure the IP address of the remote PE router. To configure the IP address of the remote PE router, include the **neighbor** statement and specify the IP address of the loopback interface on Router PE1. Configure the virtual circuit ID to be the same as the virtual circuit ID on the neighbor router. To allow a Layer 2 circuit to be established even though the MTU configured on the local PE router does not match the MTU configured on the remote PE router, include the **ignore-mtu-mismatch** statement. Also disable the use of the control word for demultiplexing by including the **no-control-word** statement.

   ```
   [edit protocols]
   l2circuit {
     neighbor 1.1.1.1 {
       interface iw0.0 {
         virtual-circuit-id 100;
         no-control-word;
         ignore-mtu-mismatch;
       }
     }
   }
   ```

3. On Router PE5, configure the Layer 2 VPN protocols by including the **l2vpn** statement at the **[edit routing-instances *routing-instances-name* protocols]** hierarchy level. To configure the **iw0** interface, include the **interfaces** statement and specify **iw0** as the interface name. The **iw0** interface is configured under the Layer 2 VPN protocols to

receive the looped packet from the **iw0.1**. Protocols l2vpn is configured on Router PE5 with a site CE5, which is configured in the BGP L2VPN. The CE1 should have communication to CE5, through the Layer 2 interworking configuration on Router PE5.

```
[edit]
routing-instances {
   L2VPN {
      instance-type l2vpn;
      interface ge-2/0/0.0;
      interface iw0.1;
      route-distinguisher 65000:5;
      vrf-target target:65000:2;
      protocols {
         l2vpn {
            encapsulation-type ethernet;
            site CE5 {
               site-identifier 5;
               interface ge-2/0/0.0 {
                  remote-site-id 3;
               }
            }
            site l2-circuit {
               site-identifier 6;
               interface iw0.1 {
                  remote-site-id 3;
               }
            }
         }
      }
   }
}
```

4. In addition to the **iw0** interface configuration, Layer 2 interworking **l2iw** protocol must be configured. Without the **l2iw** configuration, the **l2iw** routes are not formed, regardless of whether any **iw** interfaces are present.

   On Router PE5, configure the **l2iw** protocol. To configure the protocol, include the **l2iw** statement at the **[edit protocols]** hierarchy level.

```
[edit]
protocols {
   l2iw;
}
```

Verification

Step-by-Step Procedure

Verifying the Layer 2 Circuit Connection on Router PE1

1. On Router PE1, use the **show l2circuit connections** command to verify that the Layer 2 Circuit from Router PE1 to Router PE5 is **Up**.

```
user@PE1> show l2circuit connections

Layer-2 Circuit Connections:
Legend for connection status (St)
```

```
EI -- encapsulation invalid      NP -- interface h/w not present
MM -- mtu mismatch               Dn -- down
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
VM -- vlan id mismatch           CF -- Call admission control failure
OL -- no outgoing label          IB -- TDM incompatible bitrate
NC -- intf encaps not CCC/TCC    TM -- TDM misconfiguration
BK -- Backup Connection          ST -- Standby Connection
CB -- rcvd cell-bundle size bad  XX -- unknown
SP -- Static Pseudowire

Legend for interface status
Up -- operational
Dn -- down
Neighbor: 5.5.5.5
    Interface                Type  St     Time last up   # Up trans
    ge-1/0/0.0(vc 100)       rmt   Up     Jan  3 22:00:49 2010    1
      Remote PE: 5.5.5.5, Negotiated control-word: No
      Incoming label: 301328, Outgoing label: 300192
      Local interface: ge-1/0/0.0, Status: Up, Encapsulation: ETHERNET
```

2. On Router PE5, use the **show l2vpn connections** command to verify that the Layer 2 VPN connection is **Up** using the **iw0** peer interface of the Layer 2 circuit.

```
user@PE5> show l2vpn connections

Instance: L2VPN
  Local site: CE5 (5)
    connection-site           Type  St     Time last up     # Up trans
    l2-circuit (6)            loc   OR
    3                         rmt   Up
     Jan  3 22:51:12 2010           1
      Remote PE: 3.3.3.3, Negotiated control-word: Yes (Null)
      Incoming label: 800258, Outgoing label: 800000
      Local interface: ge-2/0/0.0, Status: Up, Encapsulation: ETHERNET
  Local site: l2-circuit (6)
    connection-site           Type  St     Time last up    # Up trans
    CE5 (5)                   loc   OR
    3                         rmt   Up     Jan  3 22:56:38 2010    1
      Remote PE: 3.3.3.3, Negotiated control-word: Yes (Null)
      Incoming label: 800262, Outgoing label: 800001
      Local interface: iw0.1, Status: Up, Encapsulation: ETHERNET
```

Step-by-Step Procedure Verifying the Layer 2 Circuit terminating into Layer 2 VPN connection

1. On Router PE 5, use the **show l2circuit connections** command to verify that the Layer 2 circuit is **Up** using the **iw0** interface. This will be looped through the **iwo.1** interface to the Layer 2 VPN.

```
user@PE5> show l2circuit connections


Layer-2 Circuit Connections:
Neighbor: 1.1.1.1
    Interface            Type  St     Time last up  # Up trans
    iw0.0(vc 100)  rmt   Up Jan  3 21:59:07 2010   1
      Remote PE: 1.1.1.1, Negotiated control-word: No
      Incoming label: 300192, Outgoing label: 301328
```


2. On Router PE 5, use the **show route table mpls.0** command to verify the Layer 2 circuit and Layer 2 VPN routes. In the example below, the Layer 2 circuit is associated with LDP label **301328** and the Layer 2 VPN is associated with LDP label **800001**. Notice the two **iw0** interfaces that are used for the Layer 2 interworking route.

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

mpls.0: 18 destinations, 20 routes (18 active, 2 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 5d 20:07:31, metric 1
                      Receive
1                  *[MPLS/0] 5d 20:07:31, metric 1
                      Receive
2                  *[MPLS/0] 5d 20:07:31, metric 1
                      Receive
299776             *[LDP/9] 2d 03:00:51, metric 1
300048             *[LDP/9] 2d 03:00:49, metric 1
                    > to 10.10.6.1 via xe-0/1/0.0, Pop
300048(S=0)        *[LDP/9] 2d 03:00:49, metric 1
                    > to 10.10.6.1 via xe-0/1/0.0, Pop
300192             *[L2IW/6] 19:11:05, metric2 1
                    > to 10.10.6.1 via xe-0/1/0.0, Swap 800001
                    [L2CKT/7] 20:08:36
                    > via iw0.0, Pop
800258             *[L2VPN/7] 19:16:31
                    > via ge-2/0/0.0, Pop       Offset: 4
800262         *[L2IW/6] 19:11:05, metric2 1
          > to 10.10.3.1 via xe-1/1/0.0, Swap 301328
          [L2VPN/7] 19:11:05
          > via iw0.1, Pop     Offset: 4
ge-2/0/0.0         *[L2VPN/7] 19:16:31, metric2 1
                    > to 10.10.6.1 via xe-0/1/0.0, Push 800000 Offset: -4
iw0.0              *[L2CKT/7] 20:08:36, metric2 1
                    > to 10.10.3.1 via xe-1/1/0.0, Push 301328
iw0.1              *[L2VPN/7] 19:11:05, metric2 1
                    > to 10.10.6.1 via xe-0/1/0.0, Push 800001 Offset: -4
```

Related Documentation

- Layer 2 Circuit Overview on page 63
- Layer 2 VPN Overview
- Layer 2 VPN Applications
- Using the Layer 2 Interworking Interface to Interconnect a Layer 2 Circuit to a Layer 2 VPN on page 61

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

Instead of using a physical Tunnel PIC for looping the packet received from the Layer 2 circuit, the Layer 2 interworking interface uses Junos OS to stitch together both Layer 2 VPN routes.

To configure the interworking interface, include the **iw0** statement. The **iw0** statement is configured at the **[edit interfaces]** hierarchy level. This specifies the peering between two logical interfaces. This configuration is similar to the configuration for a logical tunnel interface. The logical Interfaces must be associated with the endpoints of a Layer 2 circuit and Layer 2 VPN connections.

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

Configure the Layer 2 circuit protocol by including the **l2circuit** statement at the **[edit protocols]** hierarchy level and specifying the **neighbor** and **iw0** interface.

```
[edit protocols]
l2circuit {
   neighbor 1.2.3.4 {
      interface iw0.0;
   }
}
```

Configure the Layer 2 VPN connection, by including the ***routing-instance-name*** statement at the **[edit routing-instances]** hierarchy level and specifying the **instance-type l2vpn** option.

```
[edit routing-instances]
routing-instance-name {
   instance-type l2vpn;
   interface iw0.1;
   ...
   protocols {
      l2vpn {
         <l2vpn configuration>;
      }
   }
}
```

In addition to the **iw0** interface configuration, Layer 2 interworking **l2iw** protocols must be enabled. Without the **l2iw** configuration, the **l2iw** routes will not be formed, regardless of whether any **iw** interfaces are present. Within the l2iw protocols, only trace options can be configured in the standard fashion. The minimum configuration necessary for the feature to work is shown below:

```
[edit]
protocols {
   l2iw;
}
```

Related Documentation

- Layer 2 Circuit Overview on page 63
- Layer 2 VPN Overview
- Example: Interconnecting a Layer 2 Circuit with a Layer 2 VPN on page 53

Layer 2 Circuit Overview

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

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

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

Related Documentation

- Layer 3 VPN Overview
- Layer 2 VPN Overview
- Layer 2 VPN Applications
- Applications for Interconnecting a Layer 2 Circuit with a Layer 2 Circuit on page 64
- Applications for Interconnecting a Layer 2 Circuit with a Layer 3 VPN on page 64
- Example: Interconnecting a Layer 2 Circuit with a Layer 2 Circuit on page 65
- Example: Interconnecting a Layer 2 Circuit with a Layer 3 VPN
- Example: Interconnecting a Layer 2 Circuit with a Layer 2 VPN on page 53

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

MPLS-based Layer 2 services are growing in demand among enterprise and service providers. This creates new challenges related to interoperability between Layer 2 and Layer 3 services for service providers who want to provide end-to-end value-added services. There are various reasons to stitch different Layer 2 services to one another and to Layer 3 services. For example, to expand the service offerings and to expand geographically. The Junos OS has various features to address the needs of the service provider.

Interconnecting a Layer 2 Circuit with a Layer 3 VPN includes the following benefits:

- Interconnecting a Layer 2 Circuit with a Layer 3 VPN enables the sharing of a service provider's core network infrastructure between IP and Layer 2 circuit services, reducing the cost of providing those services. A Layer 2 MPLS circuit allows service providers to create a Layer 2 circuit service over an existing IP and MPLS backbone.
- Service providers do not have to invest in separate Layer 2 equipment to provide Layer 2 circuit service. A service provider can configure a provider edge router to run any Layer 3 protocol in addition to the Layer 2 protocols. Customers who prefer to maintain control over most of the administration of their own networks want Layer 2 circuit connections with their service provider instead of a Layer 3 VPN connection.

Related Documentation

- Layer 2 Circuit Overview on page 63
- Layer 3 VPN Overview
- Example: Interconnecting a Layer 2 Circuit with a Layer 3 VPN

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

MPLS-based Layer 2 services are growing in demand among enterprise and service providers. This creates new challenges for service providers who want to provide end-to-end value-added services. There are various reasons to stitch different Layer 2 services to one another and to Layer 3 services, for example, to expand the service offerings and to expand geographically. The Junos OS has various features to address the needs of the service provider.

Interconnecting a Layer 2 circuit with a Layer 2 circuit includes the following benefits:

- Interconnecting a Layer 2 circuit with a Layer 2 circuit enables the sharing of a service provider's core network infrastructure between Layer 2 circuit services, reducing the cost of providing those services. A Layer 2 MPLS circuit allows service providers to create a Layer 2 circuit service over an existing IP and MPLS backbone.
- Service providers do not have to invest in separate Layer 2 equipment to provide Layer 2 circuit service. A service provider can configure a provider edge router to run any Layer 2 protocol. Customers who prefer to maintain control over most of the administration of their own networks want Layer 2 circuit connections with their service provider instead of a Layer 3 VPN connection.

Related Documentation

- Layer 2 Circuit Overview on page 63
- Example: Interconnecting a Layer 2 Circuit with a Layer 2 Circuit on page 65

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

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

- Requirements on page 65
- Overview and Topology on page 65
- Configuration on page 67

Requirements

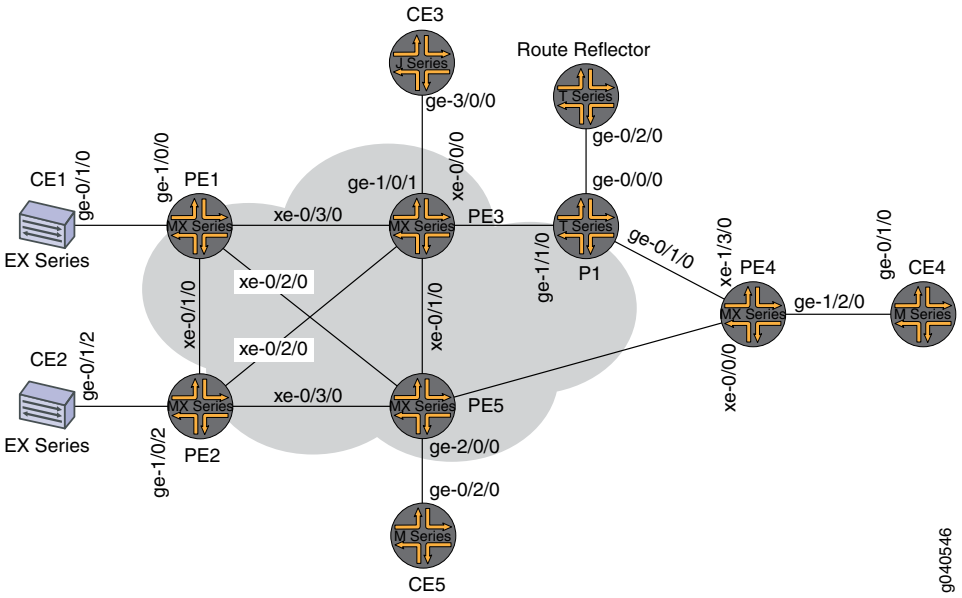
This example uses the following hardware and software components:

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

Overview and Topology

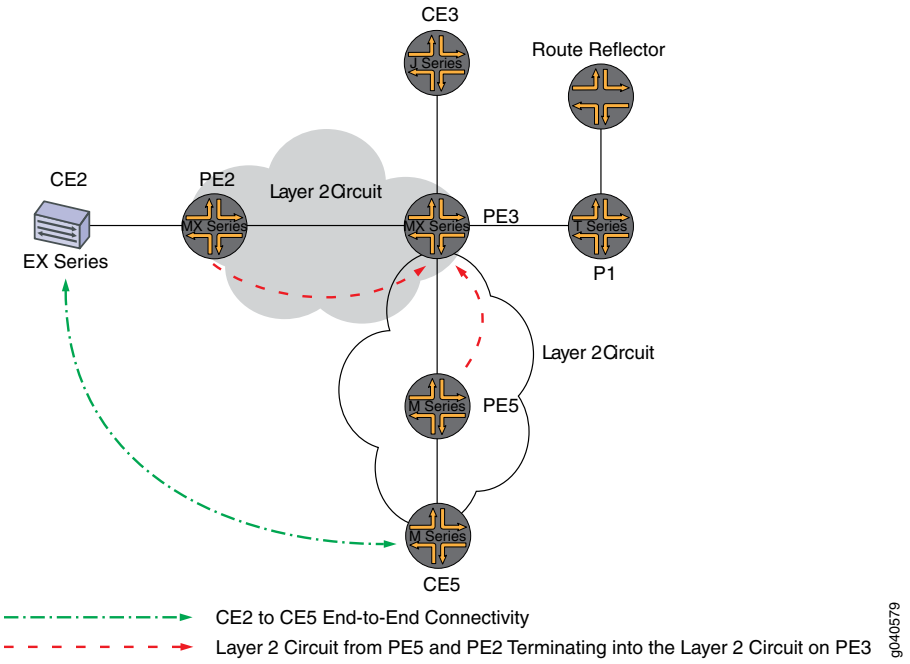
The physical topology of a Layer 2 circuit to Layer 2 circuit interconnection is shown in Figure 8 on page 66

Figure 8: Physical Topology of a Layer 2 Circuit Terminating into a Layer 2 Circuit



The logical topology of a Layer 2 circuit to Layer 2 circuit interconnection is shown in Figure 9 on page 66

Figure 9: Logical Topology of a Layer 2 Circuit Terminating into a Layer 2 Circuit



Configuration

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

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

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

This example contains the following procedures:

- Configuring PE Router Customer-facing and Loopback Interfaces on page 67
- Configuring Core-facing Interfaces on page 68
- Configuring Protocols on page 70
- Configuring the Layer 2 Circuits on page 71
- Interconnecting the Layer 2 Circuits on page 73
- Verifying the Layer 2 Circuit to Layer 2 Circuit Interconnection on page 74

Configuring PE Router Customer-facing and Loopback Interfaces

Step-by-Step Procedure

To begin building the interconnection, configure the interfaces on the PE routers. If your network contains provider (P) routers, configure the interfaces on the P routers also. This example shows the configuration for Router PE1 and Router PE5.

1. On Router PE1, configure the **ge-1/0/0** interface encapsulation. To configure the interface encapsulation, include the **encapsulation** statement and specify the **ethernet-ccc** option (vlan-ccc encapsulation is also supported). Configure the **ge-1/0/0.0** logical interface family for circuit cross-connect functionality. To configure the logical interface family, include the **family** statement and specify the **ccc** option.

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

```
        family inet {
          address 1.1.1.1/32;
        }
      }
    }
```

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

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

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

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

Configuring Core-facing Interfaces

Step-by-Step Procedure

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

1. On Router PE1, configure the **xe-0/3/0** interface. Include the **family** statement and specify the **inet** address family. Include the **address** statement and specify **10.10.1.1/30** as the interface address. Include the **family** statement and specify the **mpls** address family.

```
[edit interfaces]
xe-0/3/0 {
  unit 0 {
    family inet {
```

```
        address 10.10.1.1/30;
      }
      family mpls;
    }
  }
```

2. On Router PE3, configure the core-facing interfaces. Include the **family** statement and specify the **inet** address family. Include the **address** statement and specify the IPv4 addresses shown in the example as the interface addresses. Include the **family** statement and specify the **mpls** address family. In the example, the **xe-0/0/0** interface is connected to the route reflector, the **xe-0/1/0** interface is connected to Router PE5, the **xe-0/2/0** interface is connected to Router PE2, and the **xe-0/3/0** interface is connected to Router PE1.

```
[edit interfaces]
xe-0/0/0 {
  unit 0 {
    family inet {
      address 10.10.20.2/30;
    }
    family mpls;
  }
}
xe-0/1/0 {
  unit 0 {
    family inet {
      address 10.10.6.1/30;
    }
    family mpls;
  }
}
xe-0/2/0 {
  unit 0 {
    family inet {
      address 10.10.5.2/30;
    }
    family mpls;
  }
}
xe-0/3/0 {
  unit 0 {
    family inet {
      address 10.10.1.2/30;
    }
    family mpls;
  }
}
```

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

```
[edit interfaces]
xe-0/1/0 {
  unit 0 {
```

```
    family inet {
      address 10.10.6.2/30;
    }
    family mpls;
  }
}
```

Configuring Protocols

Step-by-Step Procedure

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

Configure all of the PE routers and P routers with OSPF as the IGP protocol. Enable MPLS and LDP protocols on all of the interfaces except **fxp.0**.

1. On Router PE1, enable OSPF as the IGP. Enable the MPLS and LDP protocols on all interfaces except **fxp.0**. LDP is used as the signaling protocol on Router PE1 for the Layer 2 circuit. The following configuration snippet shows the protocol configuration for Router PE1:

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

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

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

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

Configuring the Layer 2 Circuits

Step-by-Step Procedure

This procedure describes how to configure the Layer 2 circuits.

NOTE: In this example the **ignore-mtu-mismatch** statement is required for the circuit to come up.

1. On Router PE1, configure the Layer 2 circuit. Include the **l2circuit** statement. Include the **neighbor** statement and specify the loopback IPv4 address of Router PE3 as the neighbor. Include the interface statement and specify **ge-1/0/0.0** as the logical interface that is participating in the Layer 2 circuit. Include the **virtual-circuit-id** statement and specify **100** as the identifier. Include the **ignore-mtu-mismatch** statement to allow a Layer 2 circuit to be established even though the maximum transmission unit (MTU) configured on the local PE router does not match the MTU configured on the remote PE router.

   ```
   [edit]
   protocols {
     l2circuit {
       neighbor 3.3.3.3 {
         interface ge-1/0/0.0 {
           virtual-circuit-id 100;
           ignore-mtu-mismatch;
         }
       }
     }
   }
   ```

2. On Router PE5, configure the Layer 2 circuit. Include the **l2circuit** statement. Include the **neighbor** statement and specify the loopback IPv4 address of Router PE3 as the neighbor. Include the interface statement and specify **ge-2/0/0.0** as the logical interface that is participating in the Layer 2 circuit. Include the **virtual-circuit-id** statement and specify **200** as the identifier. Include the **ignore-mtu-mismatch**

statement to allow a Layer 2 circuit to be established even though the MTU configured on the local PE router does not match the MTU configured on the remote PE router.

```
[edit]
protocols {
    l2circuit {
        neighbor 3.3.3.3 {
            interface ge-2/0/0.0 {
                virtual-circuit-id 200;
                ignore-mtu-mismatch;
            }
        }
    }
}
```

3. On Router PE3, configure the Layer 2 circuit to Router PE1. Include the **l2circuit** statement. Include the **neighbor** statement and specify the loopback IPv4 address of Router PE1 as the neighbor. Include the interface statement and specify **iw0.0** as the logical interworking interface that is participating in the Layer 2 circuit. Include the **virtual-circuit-id** statement and specify **100** as the identifier. Include the **ignore-mtu-mismatch** statement to allow a Layer 2 circuit to be established even though the MTU configured on the local PE router does not match the MTU configured on the remote PE router.

   On Router PE3, configure the Layer 2 circuit to Router PE5. Include the **l2circuit** statement. Include the **neighbor** statement and specify the loopback IPv4 address of Router PE5 as the neighbor. Include the interface statement and specify **iw0.1** as the logical interworking interface that is participating in the Layer 2 circuit. Include the **virtual-circuit-id** statement and specify **200** as the identifier. Include the **ignore-mtu-mismatch** statement.

```
[edit protocols]
l2circuit {
    neighbor 1.1.1.1 {
        interface iw0.0 {
            virtual-circuit-id 100;
            ignore-mtu-mismatch;
        }
    }
    neighbor 5.5.5.5 {
        interface iw0.1 {
            virtual-circuit-id 200;
            ignore-mtu-mismatch;
        }
    }
}
```

Interconnecting the Layer 2 Circuits

Step-by-Step Procedure

Router PE3 is the router that is *stitching* the Layer 2 circuits together using the interworking interface. The configuration of the peer unit interfaces is what makes the interconnection.

1. On Router PE3, configure the **iw0.0** interface. Include the **encapsulation** statement and specify the **ethernet-ccc** option. Include the **peer-unit** statement and specify the logical interface unit **1** as the peer tunnel interface.

   On Router PE3, configure the **iw0.1** interface. Include the **encapsulation** statement and specify the **ethernet-ccc** option. Include the **peer-unit** statement and specify the logical interface unit **0** as the peer tunnel interface.

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

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

   ```
   [edit]
   protocols {
     l2iw;
   }
   ```

3. On each router, commit the configuration.

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

Verifying the Layer 2 Circuit to Layer 2 Circuit Interconnection

Step-by-Step Procedure

Verify that the Layer 2 circuit connection on Router PE1 is up, the LDP neighbors are correct, and the MPLS label operations are correct.

1. On Router PE1, use the **show l2circuit connections** command to verify that the Layer 2 circuit from Router PE1 to Router PE3 is **Up**.

```
user@PE1> show l2circuit connections

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

Legend for interface status
Up -- operational
Dn -- down
Neighbor: 3.3.3.3
    Interface                 Type  St     Time last up   # Up trans
    ge-1/0/0.0(vc 100)        rmt   Up     Jan  5 22:00:49 2010    1
      Remote PE: 3.3.3.3, Negotiated control-word: Yes (Null)
      Incoming label: 301328, Outgoing label: 314736
      Local interface: ge-1/0/0.0, Status: Up, Encapsulation: ETHERNET
```

2. On Router PE1, use the **show ldp neighbor** command to verify that the IPv4 address of Router PE3 is shown as the LDP neighbor.

```
user@PE1> show ldp neighbor

Address            Interface          Label space ID         Hold time
3.3.3.3            lo0.0              3.3.3.3:0                41
```

3. On Router PE 1, use the **show route table mpls.0** command to verify the Layer 2 circuit is using the LDP label to Router PE3 in both directions (Push and Pop). In the example below, the Layer 2 circuit is associated with LDP label **301328**.

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

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

0                  *[MPLS/0] 1w1d 08:25:39, metric 1
                      Receive
1                  *[MPLS/0] 1w1d 08:25:39, metric 1
                      Receive
2                  *[MPLS/0] 1w1d 08:25:39, metric 1
                      Receive
300432             *[LDP/9] 3d 01:13:57, metric 1
                    > to 10.10.2.2 via xe-0/1/0.0, Pop
```

```
300432(S=0)        *[LDP/9] 3d 01:13:57, metric 1
                    > to 10.10.2.2 via xe-0/1/0.0, Pop
300768             *[LDP/9] 3d 01:13:57, metric 1
                    > to 10.10.3.2 via xe-0/2/0.0, Pop
300768(S=0)        *[LDP/9] 3d 01:13:57, metric 1
                    > to 10.10.3.2 via xe-0/2/0.0, Pop
300912             *[LDP/9] 3d 01:13:57, metric 1
                    > to 10.10.3.2 via xe-0/2/0.0, Swap 299856
301264             *[LDP/9] 3d 01:13:53, metric 1
                    > to 10.10.1.2 via xe-0/3/0.0, Swap 308224
301312             *[LDP/9] 3d 01:13:56, metric 1
                    > to 10.10.1.2 via xe-0/3/0.0, Pop
301312(S=0)        *[LDP/9] 3d 01:13:56, metric 1
                    > to 10.10.1.2 via xe-0/3/0.0, Pop
301328        *[L2CKT/7] 02:33:26
          > via ge-1/0/0.0, Pop     Offset: 4
ge-1/0/0.0      *[L2CKT/7] 02:33:26, metric2 1
          > to 10.10.1.2 via xe-0/3/0.0, Push 314736 Offset: -4
```

4. On Router PE3, use the **show l2circuit connections** command to verify that the Layer 2 circuit from Router PE3 to Router PE5 is **Up**, that the Layer 2 circuit from Router PE3 to Router PE1 is **Up**, that the connections to Router PE1 and Router PE5 use the iw0 interface, and that the status for both local iw0 interfaces is **Up**.

```
user@PE3> show l2circuit connections

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

Legend for interface status
Up -- operational
Dn -- down
Neighbor: 1.1.1.1
    Interface                 Type  St     Time last up          # Up trans
    iw0.0(vc 100)             rmt   Up     Jan  5 13:50:14 2010           1
      Remote PE: 1.1.1.1, Negotiated control-word: Yes (Null)
      Incoming label: 314736, Outgoing label: 301328
      Local interface: iw0.0, Status: Up, Encapsulation: ETHERNET
Neighbor: 5.5.5.5
    Interface                 Type  St     Time last up          # Up trans
    iw0.1(vc 200)             rmt   Up     Jan  5 13:49:58 2010           1
      Remote PE: 5.5.5.5, Negotiated control-word: Yes (Null)
      Incoming label: 314752, Outgoing label: 300208
      Local interface: iw0.1, Status: Up, Encapsulation: ETHERNET
```

5. On Router PE3, use the **show ldp neighbor** command to verify that the correct IPv4 addresses are shown as the LDP neighbor.

```
user@PE3> show ldp neighbor
```

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

6. On Router PE3, use the **show route table mpls.0** command to verify that the **mpls.0** routing table is populated with the Layer 2 interworking routes. Notice that in this example, the router is swapping label **314736** received from Router PE1 on the **iw0.0** to label **301328**.

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

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

0                  *[MPLS/0] 1w1d 08:28:24, metric 1
                      Receive
1                  *[MPLS/0] 1w1d 08:28:24, metric 1
                      Receive
2                  *[MPLS/0] 1w1d 08:28:24, metric 1
                      Receive
308160             *[LDP/9] 3d 01:16:55, metric 1
                    > to 10.10.1.1 via xe-0/3/0.0, Pop
308160(S=0)        *[LDP/9] 3d 01:16:55, metric 1
                    > to 10.10.1.1 via xe-0/3/0.0, Pop
308176             *[LDP/9] 3d 01:16:54, metric 1
                    > to 10.10.6.2 via xe-0/1/0.0, Pop
308176(S=0)        *[LDP/9] 3d 01:16:54, metric 1
                    > to 10.10.6.2 via xe-0/1/0.0, Pop
308192             *[LDP/9] 00:21:40, metric 1
                    > to 10.10.20.1 via xe-0/0/0.0, Swap 601649
                      to 10.10.6.2 via xe-0/1/0.0, Swap 299856
308208             *[LDP/9] 3d 01:16:54, metric 1
                    > to 10.10.5.1 via xe-0/2/0.0, Pop
308208(S=0)        *[LDP/9] 3d 01:16:54, metric 1
                    > to 10.10.5.1 via xe-0/2/0.0, Pop
308224             *[LDP/9] 3d 01:16:52, metric 1
                    > to 10.10.20.1 via xe-0/0/0.0, Pop
308224(S=0)        *[LDP/9] 3d 01:16:52, metric 1
                    > to 10.10.20.1 via xe-0/0/0.0, Pop
314736             *[L2IW/6] 02:35:31, metric2 1
                    > to 10.10.6.2 via xe-0/1/0.0, Swap 300208
                    [L2CKT/7] 02:35:31
                    > via iw0.0, Pop       Offset: 4
314752             *[L2IW/6] 02:35:31, metric2 1
                    > to 10.10.1.1 via xe-0/3/0.0, Swap 301328
                    [L2CKT/7] 02:35:47
                    > via iw0.1, Pop       Offset: 4
iw0.0              *[L2CKT/7] 02:35:31, metric2 1
                   > to 10.10.1.1 via xe-0/3/0.0, Push 301328 Offset: -4
iw0.1              *[L2CKT/7] 02:35:47, metric2 1
                    > to 10.10.6.2 via xe-0/1/0.0, Push 300208 Offset:
-4
```

7. Verify that Router CE1 can send traffic to and receive traffic from Router CE5 across the interconnection, using the **ping** command.

```
user@CE1>ping 40.40.40.11
```

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

8. Verify that Router CE5 can send traffic to and receive traffic from Router CE1 across the interconnection, using the **ping** command.

```
user@CE5>ping 40.40.40.1

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

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

The relevant sample configuration for Router PE1 follows.

Router PE1

```
[edit]
interfaces {
  xe-0/1/0 {
    unit 0 {
      family inet {
        address 10.10.2.1/30;
      }
      family mpls;
    }
  }
  xe-0/2/0 {
    unit 0 {
      family inet {
        address 10.10.3..1/30;
      }
      family mpls;
    }
  }
  xe-0/3/0 {
    unit 0 {
      family inet {
        address 10.10.1.1/30;
      }
      family mpls;
    }
  }
  ge-1/0/0 {
    encapsulation ethernet-ccc;
    unit 0 {
      family ccc;
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 1.1.1.1/32;
```

```
        }
      }
    }
  }
  forwarding-options {
    hash-key {
      family inet {
        layer-3;
        layer-4;
      }
      family mpls {
        label-1;
        label-2;
      }
    }
  }
  routing-options {
    static {
      route 172.0.0.0/8 next-hop 172.19.59.1;
    }
    autonomous-system 65000;
  }
  protocols {
    mpls {
      interface all;
      interface fxp0.0 {
        disable;
      }
    }
    ospf {
      traffic-engineering;
      area 0.0.0.0 {
        interface all;
        interface fxp0.0 {
          disable;
        }
      }
    }
    ldp {
      interface all;
      interface fxp0.0 {
        disable;
      }
    }
    l2circuit {
      neighbor 3.3.3.3 {
        interface ge-1/0/0.0 {
          virtual-circuit-id 100;
          ignore-mtu-mismatch;
        }
      }
    }
  }
```

The relevant sample configuration for Router PE3 follows.

Router PE3

```
[edit]
interfaces {
    xe-0/0/0 {
        unit 0 {
            family inet {
                address 10.10.20.2/30;
            }
            family mpls;
        }
    }
    xe-0/1/0 {
        unit 0 {
            family inet {
                address 10.10.6.1/30;
            }
            family mpls;
        }
    }
    xe-0/2/0 {
        unit 0 {
            family inet {
                address 10.10.5.2/30;
            }
            family mpls;
        }
    }
    xe-0/3/0 {
        unit 0 {
            family inet {
                address 10.10.1.2/30;
            }
            family mpls;
        }
    }
    ge-1/0/1 {
        encapsulation ethernet-ccc;
        unit 0 {
            family ccc;
        }
    }
    iw0 {
        unit 0 {
            encapsulation ethernet-ccc;
            peer-unit 1;
        }
        unit 1 {
            encapsulation ethernet-ccc;
            peer-unit 0;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 3.3.3.3/32;
            }
        }
```

```
    }
  }
  routing-options {
    static {
      route 172.0.0.0/8 next-hop 172.19.59.1;
    }
    autonomous-system 65000;
  }
  protocols {
    l2iw;
    mpls {
      interface all;
      interface fxp0.0 {
        disable;
      }
    }
    ospf {
      area 0.0.0.0 {
        interface all;
        interface fxp0.0 {
          disable;
        }
      }
    }
    ldp {
      interface all;
      interface fxp0.0 {
        disable;
      }
    }
    l2circuit {
      neighbor 1.1.1.1 {
        interface iw0.0 {
          virtual-circuit-id 100;
          ignore-mtu-mismatch;
        }
      }
      neighbor 5.5.5.5 {
        interface iw0.1 {
          virtual-circuit-id 200;
          ignore-mtu-mismatch;
        }
      }
    }
  }
```

Related Documentation

- Layer 2 Circuit Overview on page 63
- Applications for Interconnecting a Layer 2 Circuit with a Layer 2 Circuit on page 64

PART 3

Administration

- Layer 2 Circuit Reference on page 83
- Summary of Layer 2 Circuit Configuration Statements on page 85

CHAPTER 6

Layer 2 Circuit Reference

- Supported Layer 2 Circuit Standards on page 83

Supported Layer 2 Circuit Standards

The Junos OS substantially supports the following RFCs, which define standards for Layer 2 circuits.

- RFC 4447, *Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)*

  The Junos OS does not support Section 5.3, "The Generalized PWid FEC Element."

- RFC 4448, *Encapsulation Methods for Transport of Ethernet over MPLS Networks*

The following Internet drafts do not define standards, but provide information about Layer 2 technologies. The IETF classifies them as "Historic."

- Internet draft draft-martini-l2circuit-encap-mpls-11.txt, *Encapsulation Methods for Transport of Layer 2 Frames Over IP and MPLS Networks*

  The Junos OS differs from the Internet draft in the following ways:

  - A packet with a sequence number of 0 (zero) is treated as out of sequence.
  - Any packet that does not have the next incremental sequence number is considered out of sequence.
  - When out-of-sequence packets arrive, the expected sequence number for the neighbor is set to the sequence number in the Layer 2 circuit control word.

- Internet draft draft-martini-l2circuit-trans-mpls-19.txt, *Transport of Layer 2 Frames Over MPLS*

Related Documentation

- Supported Carrier-of-Carriers and Interprovider VPN Standards
- Supported Layer 2 VPN Standard
- Supported Layer 3 VPN Standards
- Supported Multicast VPN Standards
- Supported VPLS Standards

- Accessing Standards Documents on the Internet

CHAPTER 7

Summary of Layer 2 Circuit Configuration Statements

bandwidth

Syntax bandwidth (*bandwidth* | ct*number bandwidth*);

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

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify bandwidth allocation for a Layer 2 circuit or for the class types of a Layer 2 circuit.

Options ***bandwidth***—Configure the bandwidth in bits per second for the Layer 2 circuit. You cannot configure the bandwidth for the Layer 2 circuit and for the class types at the same time.

ct*number bandwidth*—Configure the bandwidth in bits per second for a class type on the Layer 2 circuit. You can configure bandwidth for up to four class types (**ct0**, **ct1**, **ct2**, **ct3**) per Layer 2 circuit. If you configure the class types, you must configure them in order, starting with class type **ct0**.

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

Related Documentation
- Configuring Bandwidth Allocation and Call Admission Control in Layer 2 Circuits on page 30

community

Syntax

```
community community-name {
    invert-match;
    members community-members;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* policy-options],
[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*],
[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name* backup-neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vpls neighbor *address* backup-neighbor *address*],
[edit policy-options],
[edit protocols l2circuit neighbor *address* interface *interface-name*],
[edit protocols l2circuit neighbor *address* interface *interface-name* backup-neighbor *address*],
[edit routing-instances *routing-instance-name* protocols vpls neighbor *address* backup-neighbor *address*]

Release Information

Statement introduced before Junos OS Release 7.4.
Hierarchy levels associated with the **backup-neighbor** statement (pseudowire redundancy) added in Junos OS Release 9.2.

Description

Specify the community for the Layer 2 circuit.

Options

invert-match—Invert the results of the community expression match.

members *community-members*—Specify the members of the community.

Required Privilege Level

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

Related Documentation

- Configuring the Layer 2 Circuit Community on page 25
- Configuring Redundant Pseudowires for Layer 2 Circuits and VPLS

control-word

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

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

Release Information Statement introduced before Junos OS Release 7.4.

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

Options **control-word**—Enable the use of the control word.

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

no-control-word—Disable the use of the control word.

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

Related Documentation
- Configuring the Control Word for Frame Relay Interfaces on page 18

description

Syntax description *text*;

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

Release Information Statement introduced before Junos OS Release 7.4.

Description Provide a text description for the Layer 2 circuit. If the text includes one or more spaces, enclose the entire text string in quotation marks (" ").

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

Related Documentation
- Configuring Routing Instances on PE Routers in VPNs

egress-protection (Layer 2 circuit)

Syntax

```
egress-protection {
  protected-l2circuit {
    egress-pe address;
    ingress-pe address;
    virtual-circuit-id identifier;
  }
  protector-interface interface-name;
  protector-pe address {
    context-identifier identifier;
    lsp lsp-name;
  }
}
```

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

Release Information Statement introduced in Junos OS release 10.4.

Description Configures an egress protection virtual circuit (EPVC).

Options The other statements are explained separately.

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

Related Documentation

- Example: Configuring an Egress Protection LSP for a Layer 2 Circuit on page 40

egress-protection (MPLS)

Syntax

```
egress-protection {
  context-identifier context-id;
}
```

Hierarchy Level [edit logical-systems *logical-system-name* protocols mpls],
[edit logical-systems *logical-system-name* protocols mpls label-switched-path *lsp-name*],
[edit protocols mpls],
[edit protocols mpls label-switched-path *lsp-name*]

Release Information Statement introduced in Junos OS Release 10.4.

Description Enables an Edge Protection Virtual Circuit (EPVC) for the MPLS protocol.

Options **context-identifier *context-id*—**(Optional) Specify the context identifier using an IPv4 address.

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

encapsulation (Physical Interface)

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

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

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

Description Specify the physical link-layer encapsulation type.

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

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

atm-pvc—Use ATM PVC encapsulation.

cisco-hdlc—Use Cisco-compatible High-Level Data Link Control (HDLC) framing.

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

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

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

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

ethernet-over-atm—For interfaces that carry IPv4 traffic, use Ethernet over ATM encapsulation. When you use this encapsulation type, you cannot configure multipoint interfaces. As defined in RFC 1483, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*, this encapsulation type allows ATM interfaces to connect to devices that support only bridge protocol data units (BPDUs). 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 ARP in the payload, and drops the rest. For packets destined to the Ethernet LAN, a route lookup is done using the destination IP address. If the route lookup yields a full address match, the packet is encapsulated with an LLC/SNAP and MAC header, and the packet is forwarded to the ATM interface.

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

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

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

extended-frame-relay-ether-type-tcc—Use extended Frame Relay ether type TCC for Cisco-compatible Frame Relay for DLCIs 1 through 1022. This encapsulation is used for circuits with different media on either side of the connection.

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

extended-vlan-bridge—Use extended VLAN bridge encapsulation on Ethernet interfaces that have IEEE 802.1Q VLAN tagging and bridging enabled and that must accept packets carrying TPID 0x8100 or a user-defined TPID.

extended-vlan-ccc—Use extended VLAN encapsulation on CCC circuits with Gigabit Ethernet and 4-port Fast Ethernet interfaces that must accept packets carrying 802.1Q values. For 8-port, 12-port, and 48-port Fast Ethernet PICs, extended VLAN CCC is not supported. For 4-port Gigabit Ethernet PICs, extended VLAN CCC is not supported.

extended-vlan-tcc—For interfaces that carry IPv4 traffic, use extended VLAN encapsulation on TCC circuits with Gigabit Ethernet interfaces on which you want to use 802.1Q tagging. For 4-port Gigabit Ethernet PICs, extended VLAN TCC is not supported.

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

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

flexible-ethernet-services—For Gigabit Ethernet IQ interfaces and Gigabit Ethernet PICs with small form-factor pluggable transceivers (SFPs) (except the 10-port Gigabit Ethernet PIC and the built-in Gigabit Ethernet port on the M7i router), use flexible Ethernet services encapsulation when you want to configure multiple per-unit Ethernet encapsulations. Aggregated Ethernet bundles can use this encapsulation type. This encapsulation type allows you to configure any combination of route, TCC, CCC, Layer 2 virtual private networks (VPNs), and VPLS encapsulations on a single physical port. If you configure flexible Ethernet services encapsulation on the physical interface, VLAN IDs from 1 through 511 are no longer reserved for normal VLANs.

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

frame-relay—Use Frame Relay encapsulation.

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

frame-relay-ether-type—Use Frame Relay ether type encapsulation for compatibility with Cisco Frame Relay.

frame-relay-ether-type-tcc—Use Frame Relay ether type TCC for Cisco-compatible Frame Relay on TCC circuits to connect different media.

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

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

generic-services—Use generic services encapsulation for services with a hierarchical scheduler.

multilink-frame-relay-uni-nni—Use MLFR UNI NNI encapsulation. This encapsulation is used on link services, voice services interfaces functioning as FRF.16 bundles, and their constituent T1 or E1 interfaces, and is supported on LSQ and redundant LSQ interfaces.

ppp—Use serial PPP encapsulation.

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

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

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

vlan-vci-ccc—Use ATM-to-Ethernet interworking encapsulation on CCC circuits. When you use this encapsulation type, you can configure the **ccc** family only. All logical interfaces configured on the Ethernet interface must also have the encapsulation type set to **vlan-vci-ccc**.

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

Required Privilege Level

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

Related Documentation

- Configuring Interface Encapsulation on Physical Interfaces
- Defining the Encapsulation for Switching Cross-Connects
- Configuring ATM Interface Encapsulation
- Configuring VLAN Encapsulation
- Configuring ATM-to-Ethernet Interworking
- Configuring Extended VLAN Encapsulation
- Configuring Encapsulation for Layer 2 Wholesale VLAN Interfaces
- Configuring Interfaces for Layer 2 Circuits on page 16
- Configuring Interfaces for Layer 2 Circuits on page 16
- Configuring Interface Encapsulation on PTX Series Packet Transport Switches

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 Local Site on PE Routers in Layer 2 VPNs
- Configuring VPLS Routing Instances
- Configuring the Encapsulation Type for the Layer 2 Circuit Neighbor Interface on page 19
- Configuring an MPLS-Based Layer 2 VPN (CLI Procedure)

end-interface

Syntax

```
end-interface {
  interface interface-name;
  no-revert;
  protect-interface interface-name;
}
```

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

Release Information
Statement introduced before Junos OS Release 7.4.

Description
Specify the end interface for a local interface switch.

The remaining statements are explained separately.

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

Related Documentation
- Configuring Local Interface Switching in Layer 2 Circuits on page 14

fast-aps-switch

Syntax fast-aps-switch;

Hierarchy Level [edit interfaces *interface-name* sonet-options aps]

Release Information Statement introduced in Junos OS Release 12.1.

Description (M320 routers with Channelized OC3/STM1 Circuit Emulation PIC with SFP only) Reduce the Automatic Protection Switching (APS) switchover time in Layer 2 circuits.

NOTE:

- Configuring this statement reduces the APS switchover time only when the Layer 2 circuit encapsulation type for the interface receiving traffic from a Layer 2 circuit neighbor is SAToP.
- When the **fast-aps-switch** statement is configured in revertive APS mode, you must configure an appropriate value for revert time to achieve reduction in APS switchover time.
- To prevent the logical interfaces in the data path from being shut down, configure appropriate hold-time values on all the interfaces in the data path that support TDM.
- The **fast-aps-switch** statement cannot be configured when the APS **annex-b** option is configured.
- The interfaces that have the **fast-aps-switch** statement configured cannot be used in virtual private LAN service (VPLS) environments.

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

Related Documentation
- Reducing APS Switchover Time in Layer 2 Circuits on page 31

ignore-encapsulation-mismatch

Syntax ignore-encapsulation-mismatch;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols l2circuit local-switching interface *interface-name*],
[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*],
[edit protocols l2circuit local-switching interface *interface-name*],
[edit protocols l2circuit neighbor *address* interface *interface-name*]

Release Information
Statement introduced in Junos OS Release 9.2.
Statement extended to support local switching in Junos OS Release 10.4.

Description Allow a Layer 2 circuit to be established even though the encapsulation configured on the CE device interface does not match the encapsulation configured on the Layer 2 circuit interface.

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

Related Documentation
- Enabling the Layer 2 Circuit When the Encapsulation Does Not Match on page 19

ignore-mtu-mismatch

Syntax ignore-mtu-mismatch;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols l2circuit local-switching interface *interface-name*],
[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*],
[edit protocols l2circuit local-switching interface *interface-name*],
[edit protocols l2circuit neighbor *address* interface *interface-name*]

Release Information
Statement introduced in Junos OS Release 8.5.
Support for remote PE routers added in Junos OS Release 9.2.

Description Ignore the MTU configuration set for the physical interface associated with the local switching interface or with the remote PE router. This allows a Layer 2 circuit to be brought up between two logical interfaces that are defined on physical interfaces with different MTU values.

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

Related Documentation
- Enabling the Layer 2 Circuit When the MTU Does Not Match on page 20
- Enabling Local Interface Switching When the MTU Does Not Match on page 15

install-nexthop

Syntax install-nexthop (except | lsp *lsp-name* | lsp-regex *lsp-regular-expression*);

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

Release Information Statement introduced before Junos OS Release 7.4.

Description Select a specific label-switched path (LSP), or select an LSP from a set of similarly named LSPs as the traffic destination for the configured community. Also can prevent the installation of any matching next hops.

Options

except—Prevent the installation of any matching next hops.

lsp *lsp-name*—Configure a specific LSP.

lsp-regex *lsp-regular-expression*—Configure a range of similarly named LSPs. You can use the following wildcard characters when configuring an LSP regular expression:

- Asterisk (*)—Match any characters.
- Period (.)—Match any single digit.

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

Related Documentation

- Configuring the Policy Statement for the Layer 2 Circuit Community on page 26

interface

Syntax

```
interface interface-name {
  backup-neighbor address {
    community name;
    psn-tunnel-endpoint address;
    standby;
    virtual-circuit-id number;
  }
  bandwidth (bandwidth | ctnumber bandwidth);
  community community-name;
  (control-word | no-control-word);
  description text;
  egress-protection {
    protected-l2circuit {
      egress-pe address;
      ingress-pe address;
      virtual-circuit-id identifier;
    }
    protector-interface interface-name;
    protector-pe address {
      context-identifier identifier;
      lsp lsp-name;
      }
    }
  }
  encapsulation-type type;
  ignore-encapsulation-mismatch;
  ignore-mtu-mismatch;
  mtu mtu-number;
  no-revert;
  protect-interface interface-name;
  pseudowire-status-tlv;
  psn-tunnel-endpoint address;
  revert-time seconds;
  virtual-circuit-id identifier;
}
```

Hierarchy Level

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

Release Information

Statement introduced before Junos OS Release 7.4.

Description

Interface over which Layer 2 circuit traffic travels.

Options

interface-name—Name of the interface to configure.

The remaining statements are explained separately.

Required Privilege Level

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

Related Documentation

- Configuring the Neighbor Interface for the Layer 2 Circuit on page 16

l2circuit

Syntax

```
l2circuit {
  local-switching {
    interface interface-name {
      description text;
      end-interface {
        interface interface-name;
        protect-interface interface-name;
      }
      ignore-mtu-mismatch;
      protect-interface interface-name;
    }
  }
  neighbor address {
    interface interface-name {
      bandwidth (bandwidth | ctnumber bandwidth);
      community community-name;
      (control-word | no-control-word);
      description text;
      encapsulation-type type;
      ignore-encapsulation-mismatch;
      ignore-mtu-mismatch;
      mtu mtu-number;
      protect-interface interface-name;
      pseudowire-status-tlv;
      psn-tunnel-endpoint address;
      virtual-circuit-id identifier;
    }
  }
  traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
  }
}
```

Hierarchy Level

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

Release Information

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

Description

Enables a Layer 2 circuit.

The remaining statements are explained separately.

Required Privilege Level

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

Related Documentation

- Configuring ATM Trunking on Layer 2 Circuits on page 28
- Configuring Bandwidth Allocation and Call Admission Control in Layer 2 Circuits on page 30

- Configuring Interfaces for Layer 2 Circuits on page 16
- Configuring LDP for Layer 2 Circuits on page 9
- Configuring Policies for Layer 2 Circuits on page 25
- Configuring Static Layer 2 Circuits on page 24
- Introduction to Configuring Layer 2 Circuits on page 13
- Tracing Layer 2 Circuit Operations on page 33

local-switching

Syntax

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

Hierarchy Level [edit logical-systems *logical-system-name* protocols l2circuit],
[edit protocols l2circuit]

Release Information Statement introduced before Junos OS Release 7.4.

Description Configure a local switching interface. A local switching interface allows you to terminate a virtual circuit on the local router.

The remaining statements are explained separately.

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

Related Documentation

- Configuring Local Interface Switching in Layer 2 Circuits on page 14

mtu

Syntax mtu *mtu-number*;

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

Release Information Statement introduced before Junos OS Release 7.4.

Description Configure the MTU to be advertised for the Layer 2 circuit.

Options ***mtu-number***—MTU number to be advertised for the Layer 2 circuit.

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

Related Documentation
- Configuring the MTU Advertised for a Layer 2 Circuit on page 20

neighbor

Syntax

```
neighbor address {
  interface interface-name {
    bandwidth (bandwidth | ctnumber bandwidth);
    community community-name;
    (control-word | no-control-word);
    description text;
    ignore-encapsulation-mismatch;
    ignore-mtu-mismatch;
    mtu mtu-number;
    protect-interface interface-name;
    pseudowire-status-tlv;
    psn-tunnel-endpoint address;
    static {
      incoming-label label;
      outgoing-label label;
    }
    virtual-circuit-id identifier;
  }
}
```

Hierarchy Level [edit logical-systems *logical-system-name* protocols l2circuit],
[edit protocols l2circuit]

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

Description Each Layer 2 circuit is represented by the logical interface connecting the local provider edge (PE) router or switch to the local customer edge (CE) router or switch. All the Layer 2 circuits using a particular remote PE router or switch designated for remote CE routers or switches are listed under the **neighbor** statement (neighbor designates the PE router or switch). Each neighbor is identified by its IP address and is usually the end-point destination for the LSP tunnel (transporting the Layer 2 circuit).

Options ***address***—IP address of a neighboring router or switch.

The remaining statements are explained separately.

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

Related Documentation

- Configuring the Neighbor Interface for the Layer 2 Circuit on page 16

protect-interface

Syntax protect-interface *interface-name*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*],
[edit logical-systems *logical-system-name* protocols l2circuit local-switching interface *interface-name*],
[edit logical-systems *logical-system-name* protocols l2circuit local-switching interface *interface-name* end-interface],
[edit protocols l2circuit local-switching interface *interface-name*],
[edit protocols l2circuit neighbor *address* interface *interface-name*],
[edit protocols l2circuit local-switching interface *interface-name* end-interface]

Release Information Statement introduced before Junos OS Release 7.4.

Description Provide a backup for the protected interface in case of failure. Network traffic uses the primary interface only, as long as the primary interface functions.

Options ***interface-name***—Name of the protect interface to configure.

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

Related Documentation
- Configuring the Protect Interface on page 21

protected-l2circuit

Syntax

```
protected-l2circuit {
  egress-pe address;
  ingress-pe address;
  virtual-circuit-id identifier;
}
```

Hierarchy Level

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

Release Information

Statement introduced in Junos OS release 10.4.

Description

Configures the protected Layer 2 circuit as part of an egress protection virtual circuit (EPVC).

Options

egress-pe *address*—Specify the address of the egress PE router for the protected Layer 2 circuit.

ingress-pe *address*—Specify the address of the ingress PE router for the protected Layer 2 circuit.

virtual-circuit-id *identifier*—Specify the virtual circuit identifier for the protected Layer 2 circuit.

Required Privilege Level

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

Related Documentation

- Example: Configuring an Egress Protection LSP for a Layer 2 Circuit on page 40

protector-interface

Syntax protector-interface *interface-name*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name* egress-protection],
[edit protocols l2circuit neighbor *address* interface *interface-name* egress-protection]

Release Information Statement introduced in Junos OS release 10.4.

Description Configures the protector interface for an egress protection LSP.

Options ***interface-name***——Name of the interface used to protect traffic for an egress protection LSP.

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

Related Documentation
- Example: Configuring an Egress Protection LSP for a Layer 2 Circuit on page 40

protector-pe

Syntax
```
protector-pe address {
  context-identifier identifier;
  lsp lsp-name;
  }
}
```

Hierarchy Level [edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name* egress-protection],
[edit protocols l2circuit neighbor *address* interface *interface-name* egress-protection]

Release Information Statement introduced in Junos OS release 10.4.

Description Configures the protector PE router for an egress protection LSP. Test.

Options ***address***——IPv4 address for the protector PE router.

context-identifier *identifier*——Identifies the context for the egress protection LSP.

lsp *lsp-name*——Specifies the LSP for the egress protection LSP.

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

Related Documentation
- Example: Configuring an Egress Protection LSP for a Layer 2 Circuit on page 40

pseudowire-status-tlv

Syntax pseudowire-status-tlv;

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

Release Information Statement introduced in Junos OS Release 10.0.

Description Enables the pseudowire type length variable (TLV). The pseudowire status TLV is used to communicate the status of a pseudowire back and forth between two PE routers. The pseudowire status TLV is configurable for each pseudowire connection and is disabled by default.

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

Related Documentation
- Configuring the Pseudowire Status TLV on page 21

psn-tunnel-endpoint

Syntax psn-tunnel-endpoint *address*;

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

Release Information Statement introduced before Junos OS Release 7.4.
Hierarchy levels associated with the **backup-neighbor** statement added in Junos OS Release 9.2.

Description Specify the endpoint of the packet switched network (PSN) tunnel on the remote PE router.

Options ***address***—Address for the tunnel endpoint.

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

Related Documentation
- Configuring Layer 2 Circuits over Both RSVP and LDP LSPs on page 22
- Configuring Redundant Pseudowires for Layer 2 Circuits and VPLS

static

Syntax

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

Hierarchy Level

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

Release Information

Statement introduced in Junos OS Release 9.5.

Description

Configures static Layer 2 circuit pseudowires. Static pseudowires are designed for networks that do not support LDP or do not have LDP enabled. You configure a static pseudowire by configuring static values for the in and out labels needed to enable a pseudowire connection.

Options

incoming-label *label*—Configure the incoming label for the static pseudowire.

outgoing-label *label*—Configure the outgoing label for the static pseudowire.

Required Privilege Level

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

Related Documentation

- Configuring Static Layer 2 Circuits on page 24

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* protocols l2circuit],
[edit protocols l2circuit]

Release Information Statement introduced before Junos OS Release 7.4.

Description Trace traffic flowing through a Layer 2 circuit.

Options

disable—(Optional) Disable the tracing operation. You can use this option to disable a single operation when you have defined a broad group of tracing operations, such as **all**.

file *filename*—Name of the file to receive the output of the tracing operation. Enclose the name in quotation marks (" ").

files *number*—(Optional) Maximum number of trace files. When a trace file named ***trace-file*** reaches its maximum size, it is renamed ***trace-file*.0**, then ***trace-file*.1**, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum number of files, you also must specify a maximum file size with the **size** option.

Range: 2 through 1000 files

Default: 2 files

flag *flag*—Tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements.

- **connections**—Layer 2 circuit connections (events and state changes)
- **error**—Error conditions
- **fec**—Layer 2 circuit advertisements received or sent by means of LDP
- **topology**—Layer 2 circuit topology changes caused by reconfiguration or advertisements received from other PE routers

flag-modifier—(Optional) Modifier for the tracing flag. You can specify the **detail** modifier if you want to provide detailed trace information.

no-world-readable—(Optional) Do not allow any user to read the log file.

size *size*—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named ***trace-file*** reaches this size, it is renamed ***trace-file*.0**. When the ***trace-file*** again reaches its maximum size, ***trace-file*.0** is renamed

trace-file*.1** and ***trace-file is renamed ***trace-file*.0**. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum file size, you also must specify a maximum number of trace files with the **files** option.

Syntax: ***x*k** to specify kilobytes, ***x*m** to specify megabytes, or ***x*g** to specify gigabytes

Range: 10 KB through the maximum file size supported on your system

Default: 1 MB

world-readable—(Optional) Allow any user to read the log file.

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

Related Documentation
- Tracing Layer 2 Circuit Operations on page 33

virtual-circuit-id

Syntax
virtual-circuit-id *identifier*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name*],
[edit logical-systems *logical-system-name* protocols l2circuit neighbor *address* interface *interface-name* backup-neighbor *address*],
[edit protocols l2circuit neighbor *address* interface *interface-name*],
[edit protocols l2circuit neighbor *address* interface *interface-name* backup-neighbor *address*]

Release Information
Statement introduced before Junos OS Release 7.4. Hierarchy levels for **backup-neighbor** (pseudowire redundancy) added in Junos OS Release 9.2.
Statement introduced in Junos OS Release 11.1 for EX Series switches.

Description
Uniquely identify a Layer 2 circuit for either a regular pseudowire or a redundant pseudowire.

Options
identifier—1 through 4,294,967,295

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

Related Documentation
- Configuring the Virtual Circuit ID on page 23
- Configuring Redundant Pseudowires for Layer 2 Circuits and VPLS

PART 4

Index

- Index on page 115

Index

Symbols

#, comments in configuration statements....xiv
( ), in syntax descriptions....xiv
< >, in syntax descriptions....xiv
[ ], in configuration statements....xiv
{ }, in configuration statements....xiv
| (pipe), in syntax descriptions....xiv

A

ATM trunking....6
    Layer 2 circuits....28

B

bandwidth accounting....4
bandwidth statement....85
    Layer 2 circuits
        usage guidelines....30
braces, in configuration statements....xiv
brackets
    angle, in syntax descriptions....xiv
    square, in configuration statements....xiv

C

CAC....4, 30
    fast reroute....6
    link and node protection....6
    LSP path protection....5
    secondary paths....5
call admission control *See* CAC
CCC
    Frame Relay, control word....18
comments, in configuration statements....xiv
community statement....86
    Layer 2 circuits
        usage guidelines....17, 25
control word, Frame Relay....18
control-word statement
    Layer 2 circuits....87
        usage guidelines....19
conventions
    text and syntax....xiii
curly braces, in configuration statements....xiv
customer support....xv
    contacting JTAC....xv

D

description statement....87
documentation
    comments on....xiv

E

egress protection LSP, configuring....40
egress-protection statement
    Layer 2 circuits....88
    MPLS....88
encapsulation mismatch
    Layer 2 circuits....19
encapsulation statement
    physical interface....89
encapsulation-type statement....93
    Layer 2 circuits
        usage guidelines....19
end-interface statement....94
    usage guidelines....14
example
    egress protection LSP....40

F

fast reroute, CAC....6
fast-aps-switch
    configuring....31
fast-aps-switch statement....95
font conventions....xiii

I

ignore-encapsulation-mismatch statement....96
    usage guidelines....19
ignore-mtu-mismatch statement....96
    usage guidelines....15, 20
install-nexthop statement....97
    usage guidelines....26
interface statement
    Layer 2 circuits....98
        usage guidelines....16

L

l2circuit statement....100

Layer 2 circuits
CAC....4
egress protection LSP....40
supported software standards....83
Layer 2 circuit
MTU....20
Layer 2 circuits
ATM trunking....6, 28
bandwidth accounting....4
call admission control....4
encapsulation mismatch....19
local interface switching....14
static pseudowires....24
trunk mode....6
link and node protection, CAC....6
local interface switching....14
local switching interface
MTU....15
local-switching statement
Layer 2 circuits....101
usage guidelines....14

M
manuals
comments on....xiv
MTU
Layer 2 circuit....20
local switching interface....15
mtu statement....102
usage guidelines....20

N
neighbor statement....103
no-control-word statement
Layer 2 circuits
usage guidelines....19

P
parentheses, in syntax descriptions....xiv
protect-interface statement....104
protected-l2circuit statement....105
protector-interface statement....106
protector-pe statement....106
pseudowire-status-tlv statement....107
usage guidelines....21
pseudowires
static....24
psn-tunnel-endpoint statement....108
usage guidelines....22

R
reducing aps switchover time....31
layer 2 circuits
configuring....31

S
static pseudowires
Layer 2 circuits....24
static statement
Layer 2 circuits....109
usage guidelines....24
support, technical *See* technical support
syntax conventions....xiii

T
technical support
contacting JTAC....xv
traceoptions statement....110
trunk mode....6

V
virtual-circuit-id statement....111